Metagenomic analysis of the biodiversity and seasonal variation in the meromictic Antarctic lake, Ace Lake

Pratibha Panwar

A thesis in fulfilment of the requirements for
the degree of Doctor of Philosophy

UNSW
SYDNEY

School of Biotechnology and Biomolecular Sciences
Faculty of Science
University of New South Wales

January 2021

THESIS TITLE & ABSTRACT

Thesis Title

Metagenomic analysis of the biodiversity and seasonal variation in the meromictic Antarctic lake, Ace Lake

Thesis Abstract

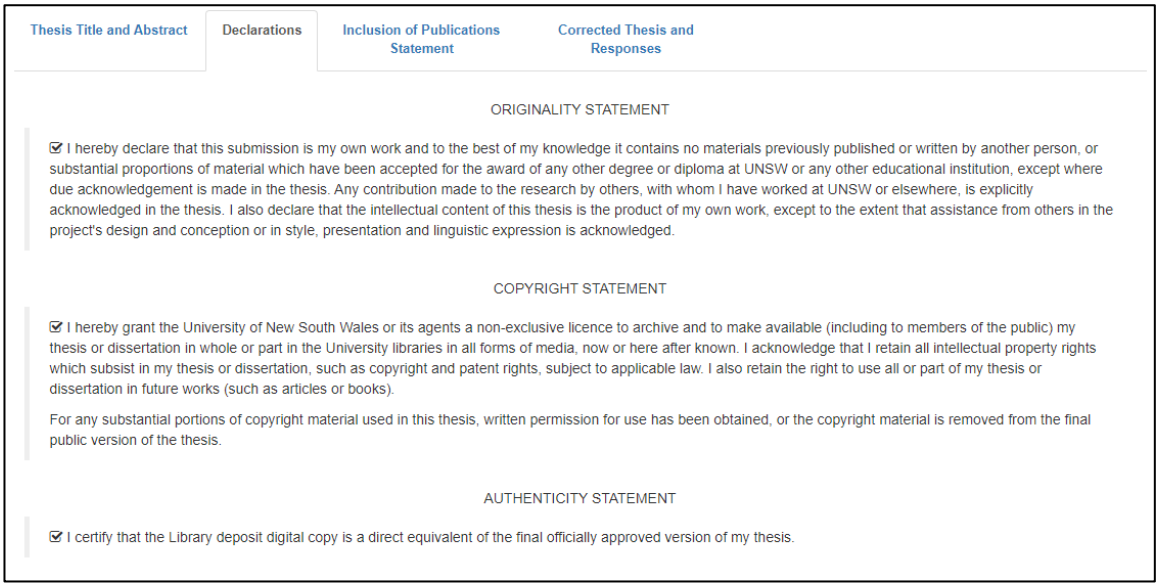
Ace lake is a stratified lake in the Vestfold Hills, Antarctica. The presence of a thick ice-cover for ~11 months of the year and a strong salinity gradient are responsible for its permanent stratification. Taxonomy analyses showed depth-based segregation of its microbial community, including viruses. Functional potential analyses of the lake taxa highlighted their roles in nutrient cycling.

In this thesis, the seasonal changes in Ace Lake microbial community were studied using a time-series of metagenomes utilizing the Cavlab metagenome analysis pipeline. Statistical analyses of taxa abundance and environmental factors revealed the effects of the polar light cycle, with 24 hours of daylight in summer and no sunlight in winter, on the phototrophs identified in the lake, indicating the importance of light-based primary production in summer to prevail through the dark winter. Analysis of viral data generated from the metagenomes showed the presence of viruses, including a 'huge phage', throughout the lake, with a diverse population existing in the oxic zone. Analysis of virus-host associations of phototrophic bacteria revealed that the availability of light, rather than viral predation, was probably responsible for seasonal variations in host abundances.

Genomic variation in *Synechococcus* and *Chlorobium* populations, analysed using metagenome-assembled genomes (MAGs) from Ace Lake, revealed phylotypes that highlighted their adaptation to the lake environment. *Synechococcus* phylotypes were linked to complex interaction with viruses, whereas some *Chlorobium* phylotypes were inferred to interact with *Synechococcus*. Some *Chlorobium* phylotypes were also inferred to have improved photosynthetic capacity, which might contribute to the very high abundance of this species in Ace Lake.

Comparative genomic analysis of *Chlorobium* was performed using MAGs from Ace Lake, Ellis Fjord, and Taynaya Bay and the genome of a non-Antarctic *Chlorobium phaeovibrioides*. A single *Chlorobium* species, distinct from the non-Antarctic species, was prevalent in the oxycline of all three stratified systems, highlighting its endemicity to the Vestfold Hills. Potential *Chlorobium* viruses, representing generalist viruses, were identified in aquatic systems from the Vestfold Hills and the Rauer Islands, indicating a widespread geographic distribution. Seasonal variation in the *Chlorobium* population appeared to be caused by reliance on sunlight rather than the impact of viral predation, and was inferred to benefit the host by restricting the ability of specialist viruses to establish effective lifecycles. The findings in this thesis highlight the seasonal influence on Ace Lake biodiversity, the adaptations and potential interactions of the two key species *Synechococcus* and *Chlorobium*, and the endemicity of Ace Lake *Chlorobium* to the Vestfold Hills.

ORIGINALITY, COPYRIGHT AND AUTHENTICITY STATEMENTS



Thesis Title and Abstract | Declarations | Inclusion of Publications Statement | Corrected Thesis and Responses

ORIGINALITY STATEMENT

☑ I hereby declare that this submission is my own work and to the best of my knowledge it contains no materials previously published or written by another person, or substantial proportions of material which have been accepted for the award of any other degree or diploma at UNSW or any other educational institution, except where due acknowledgement is made in the thesis. Any contribution made to the research by others, with whom I have worked at UNSW or elsewhere, is explicitly acknowledged in the thesis. I also declare that the intellectual content of this thesis is the product of my own work, except to the extent that assistance from others in the project's design and conception or in style, presentation and linguistic expression is acknowledged.

COPYRIGHT STATEMENT

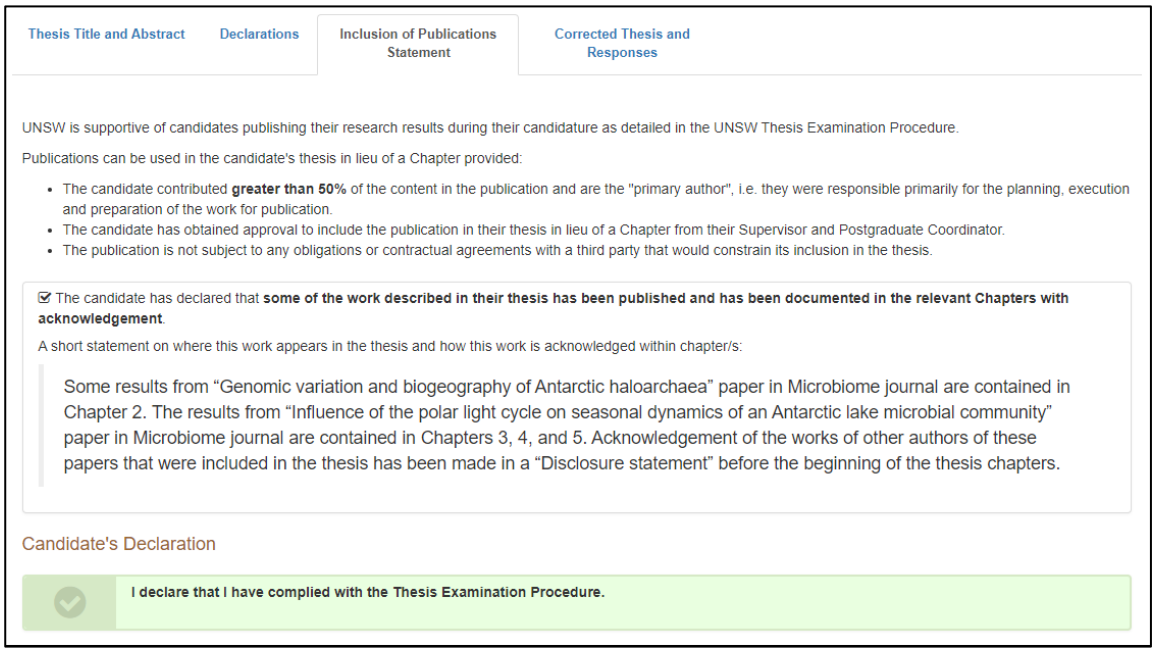
☑ I hereby grant the University of New South Wales or its agents a non-exclusive licence to archive and to make available (including to members of the public) my thesis or dissertation in whole or part in the University libraries in all forms of media, now or here after known. I acknowledge that I retain all intellectual property rights which subsist in my thesis or dissertation, such as copyright and patent rights, subject to applicable law. I also retain the right to use all or part of my thesis or dissertation in future works (such as articles or books).

For any substantial portions of copyright material used in this thesis, written permission for use has been obtained, or the copyright material is removed from the final public version of the thesis.

AUTHENTICITY STATEMENT

☑ I certify that the Library deposit digital copy is a direct equivalent of the final officially approved version of my thesis.

INCLUSION OF PUBLICATIONS STATEMENT



Thesis Title and Abstract | Declarations | Inclusion of Publications Statement | Corrected Thesis and Responses

UNSW is supportive of candidates publishing their research results during their candidature as detailed in the UNSW Thesis Examination Procedure.

Publications can be used in the candidate's thesis in lieu of a Chapter provided:

- The candidate contributed **greater than 50%** of the content in the publication and are the "primary author", i.e. they were responsible primarily for the planning, execution and preparation of the work for publication.
- The candidate has obtained approval to include the publication in their thesis in lieu of a Chapter from their Supervisor and Postgraduate Coordinator.
- The publication is not subject to any obligations or contractual agreements with a third party that would constrain its inclusion in the thesis.

☑ The candidate has declared that **some of the work described in their thesis has been published and has been documented in the relevant Chapters with acknowledgement**.

A short statement on where this work appears in the thesis and how this work is acknowledged within chapter/s:

> Some results from “Genomic variation and biogeography of Antarctic haloarchaea” paper in Microbiome journal are contained in Chapter 2. The results from “Influence of the polar light cycle on seasonal dynamics of an Antarctic lake microbial community” paper in Microbiome journal are contained in Chapters 3, 4, and 5. Acknowledgement of the works of other authors of these papers that were included in the thesis has been made in a “Disclosure statement” before the beginning of the thesis chapters.

Candidate's Declaration

I declare that I have complied with the Thesis Examination Procedure.

Abstract

Ace lake is a stratified lake in the Vestfold Hills, Antarctica. The presence of a thick ice-cover for ~11 months of the year and a strong salinity gradient are responsible for its permanent stratification. Taxonomy analyses showed depth-based segregation of its microbial community, including viruses. Functional potential analyses of the lake taxa highlighted their roles in nutrient cycling.

In this thesis, the seasonal changes in Ace Lake microbial community were studied using a time-series of metagenomes utilizing the Cavlab metagenome analysis pipeline. Statistical analyses of taxa abundance and environmental factors revealed the effects of the polar light cycle, with 24 hours of daylight in summer and no sunlight in winter, on the phototrophs identified in the lake, indicating the importance of light-based primary production in summer to prevail through the dark winter. Analysis of viral data generated from the metagenomes showed the presence of viruses, including a 'huge phage', throughout the lake, with a diverse population existing in the oxic zone. Analysis of virus-host associations of phototrophic bacteria revealed that the availability of light, rather than viral predation, was probably responsible for seasonal variations in host abundances.

Genomic variation in *Synechococcus* and *Chlorobium* populations, analysed using metagenome-assembled genomes (MAGs) from Ace Lake, revealed phylotypes that highlighted their adaptation to the lake environment. *Synechococcus* phylotypes were linked to complex interaction with viruses, whereas some *Chlorobium* phylotypes were inferred to interact with *Synechococcus*. Some *Chlorobium* phylotypes were also inferred to have improved photosynthetic capacity, which might contribute to the very high abundance of this species in Ace Lake.

Comparative genomic analysis of *Chlorobium* was performed using MAGs from Ace Lake, Ellis Fjord, and Taynaya Bay and the genome of a non-Antarctic *Chlorobium phaeovibrioides*. A single *Chlorobium* species, distinct from the non-Antarctic species, was prevalent in the oxycline of all three stratified systems, highlighting its endemicity to the Vestfold Hills. Potential *Chlorobium* viruses, representing generalist viruses, were identified in aquatic systems from the Vestfold Hills and the Rauer Islands,

indicating a widespread geographic distribution. Seasonal variation in the *Chlorobium* population appeared to be caused by reliance on sunlight rather than the impact of viral predation, and was inferred to benefit the host by restricting the ability of specialist viruses to establish effective lifecycles. The findings in this thesis highlight the seasonal influence on Ace Lake biodiversity, the adaptations and potential interactions of the two key species *Synechococcus* and *Chlorobium*, and the endemicity of Ace Lake *Chlorobium* to the Vestfold Hills.

Acknowledgements

First and foremost, I would like to thank my supervisor Rick for his constant guidance and support throughout my candidature. He encouraged me to explore and formulate new ideas and concepts and helped me to look at them from different perspectives, while at the same time keeping me on track with his invaluable feedbacks. I have learnt so much from you and it has helped me become a better scientist. I could not have wished for better supervision.

I would also like to thank all members of the lab that have been here throughout my candidature. A special thanks to Tim and Michelle, who were always there to help. Whether it was a discussion on some metabolic pathway or some stubborn software that refused to work, you two were always there to help and guide. Your advices have been invaluable. Thank you to all others that were present, Josh, Peter, Susanne, Evan, Sabrina, and Liang, our discussions were fun and enlightening.

A very special thanks to my husband Gautam, without whose constant support I would never have been able to complete my thesis. I am forever grateful to my parents and my brother. I am lucky to have you all. You have always believed in me, even in times when I doubted myself. Your support and encouragement have kept me going through this major endeavour of my life, and they still fuel my dreams to accomplish more.

List of Publications

Tschitschko B, Erdmann S, DeMaere MZ, Roux S, **Panwar P**, Allen MA, Williams TJ, Brazendale S, Hancock AM, Eloe-Fadrosh EA, Cavicchioli R. *Genomic variation and biogeography of Antarctic haloarchaea.* Microbiome. 2018;6:113.

Panwar P, Allen MA, Williams TJ, Hancock AM, Brazendale S, Bevington J, Roux S, Páez-Espino D, Nayfach S, Berg M, Schulz F, Chen IMA, Huntemann M, Shapiro N, Kyrpides NC, Woyke T, Eloe-Fadrosh EA, Cavicchioli R. *Influence of the polar light cycle on seasonal dynamics of an Antarctic lake microbial community.* Microbiome. 2020;8:116.

Table of Contents

Abstract……ii

Acknowledgements……iv

List of Publications……..v

Table of Contents……vi

List of Figures……...xiv

List of Tables……...xviii

List of Python Scripts……..xxi

List of Abbreviations……...xxii

Disclosure Statement……...xxv

1. General introduction……..1

1.1 Antarctica…….……1

1.1.1 Antarctic lake biodiversity and ecology……1

1.1.2 Physical characteristics of stratified aquatic systems in the Vestfold Hills, Antarctica……...……...5

1.2 Ace Lake — a stratified lake in the Vestfold Hills, Antarctica……...6

1.2.1 Light penetration……...11

1.2.2 Biodiversity……...12

1.2.2.1 Mixolimnion……15

1.2.2.2 Oxycline……..16

1.2.2.3 Monimolimnion……...17

1.2.3 Water chemistry and nutrient cycling……...17

1.2.3.1 Carbon……17

1.2.3.2 Sulfur……...19

1.2.3.3 Nitrogen……...20

1.2.3.4 Other macronutrients and trace metals……22

1.3 *Chlorobium* species and their geographic distribution……...23
1.4 Metagenomics……28
1.4.1 Methods for metagenomic data analysis……...29
1.4.1.1 Sequence alignment……32
1.4.1.2 Taxonomic classification……34
1.5 Objectives……...37

2. Computational approaches to analyse metagenomic data……...…41

2.1 Introduction……41
2.1.1 Antarctic metagenomes……41
2.1.2 Computational software/approaches tested for metagenome data analysis……....42
2.1.3 Aims……..47
2.2 Method development……..48
2.2.1 Improving the preliminary Cavlab pipeline……..48
2.2.2 Taxonomic identification……..53
2.2.2.1 DIAMOND and MEGAN6……53
2.2.2.2 LAST and MEGAN-LR……..56
2.2.2.3 MetaPhlAn2……61
2.2.2.4 Kaiju……64
2.2.2.5 Protein taxonomy-based contig taxonomic classification and abundance estimation……65
2.2.3 Functional potential analysis……65
2.2.3.1 DIAMOND and MEGAN6……65
2.2.3.2 COG and KEGG functional potential analyses……...66
2.2.3.3 arCOG functional potential analysis……...67
2.2.4 Refining and verifying OTU taxonomy……70
2.2.4.1 Refining OTU bins……..70
2.2.4.2 Verifying OTU taxonomy……...74
2.2.5 Contig alignment and genome visualisation……76
2.2.5.1 Contig alignment……76
2.2.5.2 Genome visualisation……..77
2.2.6 Assessing OTU phylogeny……...78

2.2.7 Statistical analyses……78
2.3 Method test results and discussion……79
2.3.1 Metagenome taxonomic diversity and species abundance……...80
2.3.1.1 Read-based taxonomic diversity analysis……...80
2.3.1.2 Protein-based taxonomic diversity analysis……85
2.3.1.3 Contig-based taxonomic diversity analysis……87
2.3.1.4 Changes in metagenome assembly method and its impact on the development of Cavlab pipeline……93
2.3.2 OTU bin refinement and taxonomy verification……..97
2.3.3 Functional potential analysis of a system using metagenomes……..100
2.3.3.1 DIAMOND/MEGAN6 COG analysis……..100
2.3.3.2 IMG COG annotation data-based analysis……101
2.3.3.3 arCOG analysis……104
2.3.3.4 KEGG analysis……..105
2.3.4 Metagenome statistical analyses……107
2.3.5 Genomic analyses……...110
2.3.6 Development of a metagenome analysis pipeline……..110
2.4 Conclusion……115

3. Seasonal variation in Ace Lake biodiversity and the functional potential of its microbial community……..119

3.1 Introduction……..119
3.1.1 Ace Lake metagenomes……..120
3.1.2 Aims……120
3.2 Methods…….....121
3.2.1 Taxonomic classification, abundance calculation, and functional potential analyses using Cavlab pipeline v4……121
3.2.2 OTU bin refinement, taxonomy verification, and preparation of high-quality OTU bins……...123
3.2.3 Ace Lake metadata collection from various seasons and lake depths……124
3.2.4 Statistical analyses……..124
3.2.4.1 Assessing alpha diversity and OTUs contributing to seasonal variation……124

3.2.4.2 Assessing relationship between OTU abundance variation and changes in season……..125
3.2.4.3 Determining associations between specific OTUs, or virus and host……125
3.2.5 Unassigned data analyses……...125
3.2.6 Viral analyses……..126
3.2.6.1 Analysis of potential GSB viruses……127
3.2.6.2 Analysis of potential *Synechococcus* viruses……128
3.2.6.3 Analysis of algal viruses……...130
3.2.6.4 Analysis of viral contigs representing complete genomes……130
3.2.6.5 Analysis of viruses containing defence genes……...130
3.2.6.6 Analysis of abundant Ace Lake viral clusters……...131
3.3 Results and discussion……..132
3.3.1 Antarctic seasons: defining seasons in polar regions……132
3.3.2 Seasonal changes in Ace Lake environment……..137
3.3.3 Ace Lake biodiversity……138
3.3.3.1 Eukarya……..138
3.3.3.2 Bacteria……..140
3.3.3.3 Archaea……..143
3.3.4 Seasonal variations in OTU abundances……144
3.3.5 Ace Lake viruses……152
3.3.5.1 Viral contigs representing complete genomes……..154
3.3.5.2 Ace Lake ‘huge’ phage with defence genes……..155
3.3.5.3 The abundant Ace Lake viral clusters……...158
3.3.5.4 Algal viruses……..159
3.3.5.5 Ace Lake cyanophage……...161
3.3.5.6 Potential *Chlorobium* viruses……161
3.3.6 ‘Other’ taxa and unassigned contigs……...166
3.3.7 Overall functional potential of Ace Lake……...170
3.3.7.1 KEGG analysis……..173
3.4 Conclusion……178

4. Ace Lake *Synechococcus* — genomic variation and potential for defence against viruses……..182

4.1 Introduction……..182

4.1.1 Aims……184

4.2 Methods……184

4.2.1 Preliminary analysis of genomic variation within Ace Lake *Synechococcus* population using MAGs……...…….185

4.2.2 FR analysis of genomic variation within Ace Lake *Synechococcus* population ……...……..186

4.2.3 Phylogeny assessment……188

4.2.4 Analysis of *Synechococcus* defence system genes…….189

4.3 Results……..192

4.3.1 Overview of *Synechococcus* MAGs and SynAce01 genome…….192

4.3.2 *Synechococcus 16S rRNA* gene identity, ANI, AAI, and phylogeny…….193

4.3.3 Analysis of sequence variations between *Synechococcus* MAGs……..195

4.3.4 Comparative analysis of *Synechococcus* MAGs and SynAce01……195

4.3.5 FR analysis of SynAce01 in Ace Lake metagenomes……198

4.3.5.1 Variable coverage regions…….199

4.3.5.2 SNPs……..199

4.3.6 Defence genes in Ace Lake *Synechococcus*……...206

4.4 Discussion……214

4.4.1 *Synechococcus* genomic variation — phylotypes and potential ecotypes…….214

4.4.1.1 *Synechococcus* subpopulations representing a potential ecotype….214

4.4.1.2 *Synechococcus* subpopulations with varying cell wall composition.216

4.4.1.3 *Synechococcus* subpopulations with varying capacity for cell defence and immunity……..…….217

4.4.2 *Synechococcus* potential for defence against viruses…….218

4.5 Conclusion……223

5. Ace Lake *Chlorobium* — genomic variation, defence against viruses, and endemism in Vestfold Hills…….226

5.1 Introduction……226
5.1.1 Aims……..229
5.2 Methods……..230
5.2.1 *Chlorobium* OTU bin refinement and abundance calculation in Ellis Fjord and Taynaya Bay metagenomes……..230
5.2.2 Preliminary analysis of genomic variation within Ace Lake *Chlorobium* population using MAGs……231
5.2.3 FR analyses…….232
5.2.3.1 Determining genomic variation within Ace Lake *Chlorobium* population…….232
5.2.3.2 Determining genomic variation in Chlorobium populations from Ace Lake, Ellis Fjord and Taynaya Bay…….232
5.2.3.3 Subpopulation estimations……233
5.2.4 Analysis of *Chlorobium* endemicity to the Vestfold Hills…….……234
5.2.4.1 Comparative analysis of *Chlorobium* MAGs and C-phaeov genome……..234
5.2.4.2 Comparison of *Chlorobium* markers with marker sequences in IMG databases……...235
5.2.5 Analysis of potential *Chlorobium* viruses in Ellis Fjord and Taynaya Bay metagenomes…….235
5.2.6 Analysis of *Chlorobium* defence system genes and CRISPR spacers…...236
5.2.7 Phylogeny assessment……236
5.3 Results……..237
5.3.1 Overview of Ace Lake, Ellis Fjord, and Taynaya Bay *Chlorobium* MAGs and C-phaeov genome……...238
5.3.2 Analysis of genomic variation in Ace Lake *Chlorobium*……...238
5.3.2.1 Analysis of sequence variations between Ace Lake *Chlorobium* MAGs……...239
5.3.2.2 FR analysis of *Chlorobium* AL_ref MAG in Ace Lake metagenomes……239
5.3.3 *Chlorobium* relative abundance in Ace Lake, Ellis Fjord, and Taynaya Bay……248
5.3.4 Analysis of genomic variation in Ace Lake, Ellis Fjord, and Taynaya Bay *Chlorobium*……...251

5.3.4.1 *Chlorobium 16S rRNA* gene identity, BclA protein identity, ANI, AAI, and phylogeny……251
5.3.4.2 FR analysis of *Chlorobium* EF_ref MAG in Ace Lake, Ellis Fjord, and Taynaya Bay metagenomes……..255
5.3.4.3 Comparative analysis of C-phaeov and *Chlorobium* MAGs from Ace Lake, Ellis Fjord, and Taynaya Bay……263
5.3.5 Analysis of defence system genes, potential viruses, and CRISPR spacers……...266
5.3.5.1 Defence genes in *Chlorobium* MAGs……...266
5.3.5.2 Analysis of CRISPR spacers from Ace Lake, Ellis Fjord, and Taynaya Bay *Chlorobium*……..268
5.3.5.3 Potential viruses associated with Ellis Fjord and Taynaya Bay *Chlorobium*……..272
5.4 Discussion……274
5.4.1 Genomic variation in Ace Lake *Chlorobium* — potential phylotypes and ecotypes……274
5.4.1.1 Variations potentially associated with cold adaptation…….275
5.4.1.2 Variations potentially associated with cell defence and immunity……..276
5.4.1.3 *Chlorobium* subpopulations containing specific substrate transporters……...277
5.4.1.4 *Chlorobium* subpopulations capable of cobalamin biosynthesis and cobinamide salvaging……...285
5.4.2 *Chlorobium* endemism in Vestfold Hills……293
5.4.2.1 *Chlorobium* phylotypes and ecotypes in Ace Lake, Ellis Fjord, and Taynaya Bay……293
5.4.2.2 The endemicity of the Vestfold Hills *Chlorobium*.……...296
5.4.3 The Vestfold Hills *Chlorobium* potential for defence against viruses…...299
5.4.4 Biogeographic distribution of viruses associated with the Vestfold Hills *Chlorobium*……...301
5.5 Conclusion……302

6. Conclusion……307

6.1 The importance of the development of the Cavlab pipeline — an Antarctic metagenome analysis pipeline……...307
6.2 Microbial and viral population dynamics of Ace Lake……308
6.2.1 Microbial population…….....309
6.2.2 Viral population……..311
6.2.3 Seasonal variation in Ace Lake……..312
6.3 Ace Lake *Synechococcus* subpopulations — adaptation to the lake environment and a complex interplay with potential viruses…….313
6.4 Ace Lake *Chlorobium* subpopulations — adaptation and endemicity to the Vestfold Hills…….314
6.5 The Vestfold Hills *Chlorobium* viruses and their biogeographic distribution in East Antarctica……...317
6.6 The importance of manual annotation in the era of high-throughput functional auto-annotation……..319
6.7 Concluding remarks…….320

References…….323

Appendix A……...358

Appendix B……...372

Appendix C……...442

Appendix D……...566

Appendix E……...590

Appendix F……...600

Appendix G……..626

Appendix H……..637

Appendix I……686

List of Figures

Figure 1.1 Various factors affecting Antarctic lake ecology…….5

Figure 1.2 Ace Lake in Vestfold Hills, East Antarctica……..…..8

Figure 1.3 The physical, chemical, and biological structuring of Ace Lake……...10

Figure 1.4 Schematic showing metagenome preparation and analysis……...30

Figure 2.1 Software tested for the development of a pipeline for Antarctic metagenome analysis……43

Figure 2.2 DIAMOND and MEGAN6 output for Megahit-assembled Ace Lake 2008 metagenomes……...86

Figure 2.3 DIAMOND and MEGAN6 output for Megahit-assembled Deep Lake 2013-2015 time-series metagenomes……87

Figure 2.4 LAST and MEGAN-LR output for Megahit-assembled metagenomes from Antarctic meromictic lake systems — Ace Lake and Organic Lake……...89

Figure 2.5 LAST and MEGAN-LR output for Megahit-assembled metagenomes from Antarctic hypersaline lake systems — Deep Lake, Club Lake, and Rauer Island lakes…….91

Figure 2.6 Comparison of taxonomic classification methods used for relative OTU abundance estimation in Antarctic metagenomes……...95

Figure 2.7 RefineM taxonomy verification output of *Pseudomonas* and *Chlorobium* OTU bins from Spades-assembled Ace Lake metagenomes……...98

Figure 2.8 MEGAN6 COG data-based functional potential analysis of a Megahit-assembled Deep Lake metagenome……...101

Figure 2.9 COG and arCOG functional potential analyses of a Megahit-assembled Deep Lake metagenome……103

Figure 2.10 KEGG functional potential analysis of a Megahit-assembled Deep Lake metagenome…….106

Figure 2.11 PRIMER v7 analysis of Megahit-assembled metagenomes from Antarctic hypersaline lake systems — taxonomic diversity and sample clustering……..108

Figure 2.12 PRIMER v7 analysis of Megahit-assembled metagenomes from Antarctic hypersaline lake systems — species diversity analysis…….109

Figure 2.13 Cavlab pipeline v4.1 schematic……..111

Figure 3.1 Identifying potential *Chlorobium* viruses……128

Figure 3.2 Analysis of a cyanophage and some algal viruses……...129

Figure 3.3 Environmental data recorded at Davis Station, East Antarctica……..134

Figure 3.4 Ace Lake temperature, salinity, and dissolved oxygen profiles……...138

Figure 3.5 Relative abundances of eukaryal OTUs in the Upper zone of Ace Lake.…139

Figure 3.6 Relative abundances of bacterial OTUs throughout Ace Lake……141

Figure 3.7 Relative abundances of archaeal OTUs in the Lower zone of Ace Lake….144

Figure 3.8 Seasonal and depth-related variations in relative abundances of OTUs identified in Ace Lake……...145

Figure 3.9 Seasonal variation in Ace Lake alpha diversity……...147

Figure 3.10 Seasonal distribution of peak relative abundances of abundant OTUs in Ace Lake……..151

Figure 3.11 Relative abundances of viral OTUs in the Upper zone of Ace Lake…….154

Figure 3.12 Ace Lake *Chlorobium* and its potential viruses…….163

Figure 3.13 Depth and season distribution of unassigned contigs, low abundance OTUs, and OTUs with poor taxonomic assignments……167

Figure 3.14 COG category classification of proteins identified in Ace Lake metagenomes…….172

Figure 3.15 Enzymes/pathways involved in energy conservation and metabolism in Ace Lake……..176

Figure 3.16 Sulfur cycling between *Chlorobium* and some *Deltaproteobacteria* at the Ace Lake oxycline…….177

Figure 4.1 Differences in the sequence of *16S rRNA* genes from Ace Lake *Synechococcus*……...193

Figure 4.2 *16S rRNA* gene-based phylogenetic analysis of Ace Lake *Synechococcus*……...194

Figure 4.3 Alignment of *Synechococcus* MAGs against SynAce01 genome……196

Figure 4.4 *Synechococcus* abundance, coverage distribution, and genomic variation in Ace Lake metagenomes from different lake depths and time periods……..202

Figure 4.5 BREX defence system genes in SynAce01 and their coverage in Ace Lake merged metagenomes……208

Figure 4.6 The relative coverage of SynAce01 *asnB* gene in Ace Lake merged metagenomes…….216

Figure 4.7 Potential *Synechococcus* populations in Ace Lake……..222

Figure 5.1 Location of Ace Lake, Ellis Fjord, and Taynaya Bay in the Vestfold Hills……...227

Figure 5.2 *Chlorobium* abundance, coverage distribution, and genomic variation in Ace Lake Interface metagenomes from different seasons……242

Figure 5.3 *Chlorobium* abundance and coverage distribution in Ace Lake, Ellis Fjord, and Taynaya Bay……...250

Figure 5.4 Comparison of *Chlorobium* marker genes from Ace Lake, Ellis Fjord, Taynaya Bay, and C-phaeov……..253

Figure 5.5 BclA protein and *16S rRNA* gene based phylogenetic analyses of *Chlorobium* MAGs……255

Figure 5.6 Coverage pattern of EF_ref MAG in Ace Lake, Ellis Fjord, and Taynaya Bay metagenomes…….257

Figure 5.7 Sequence comparison of *Chlorobium* MAGs with C-phaeov genome……265

Figure 5.8 Comparison of spacers and repeats identified in *Chlorobium* from Ace Lake, Ellis Fjord, and Taynaya Bay….....269

Figure 5.9 Iron and cobalt concentrations in Ace Lake…….279

Figure 5.10 Cobalamin biosynthesis and cobinamide salvaging pathways……...286

Figure 5.11 Abundance and clustering of *Chlorobium* subpopulations containing low coverage genes of EF_ref MAG……294

Figure 5.12 Comparison of functional potential of the Vestfold Hills *Chlorobium* and C-phaeov……297

Figure 5.13 Biogeographic distribution of viral contigs with matches to the Vestfold Hills *Chlorobium* spacers……..302

List of Tables

Table 1.1 Changes in the position of the oxic-anoxic interface, halocline, and thermocline in Ace Lake over a period of more than three decades……...10

Table 1.2 The biodiversity of Ace Lake assessed over a period of more than three decades…….12

Table 1.3 Global distribution of some *Chlorobium* species……24

Table 2.1 Cavlab pipeline versions — issues identified and changes made to improve the pipeline……..48

Table 2.2 KO number databases for KEGG functional potential analysis…….67

Table 2.3 Read-based taxonomic diversity analysis of some Ace Lake metagenomes using MetaPhlAn2……...82

Table 2.4 Read-based taxonomic diversity analysis of some Ace Lake and Deep Lake metagenomes using Kaiju……83

Table 2.5 Comparison of contig statistics of Megahit- *vs* Spades-assembled metagenomes from some Ace Lake and Deep Lake samples…….93

Table 2.6 Verification of OTU taxonomy using RefineM, ANI, SSU rRNA identity, and matches to MetaBAT-generated MAGs……..99

Table 2.7 Cavlab pipeline v1.2 *vs* v4.1 — comparison of methods/software, input files (I) and UNSW Katana computer cluster resources (K)…….112

Table 3.1 Season description based on environmental data gathered during sample collection and at Davis Station, Antarctica……...135

Table 3.2 SIMPER analysis showing similarities between samples from a season and dissimilarities between samples from different seasons as well as the top contributing OTUs…….147

Table 3.3 Correlation between relative abundances of *Chlorobium* and members of *Deltaproteobacteria* at the Ace Lake Interface…….152

Table 3.4 Ace Lake unassigned contigs with relative abundance ≥1%.....156

Table 3.5 Algal virus OTUs identified in Ace Lake — their associated viral clusters and singletons and their correlation with potential host……...159

Table 3.6 Host analysis of two cluster 1024 (cl_1024) and one singleton (sg_14554) viral contigs……...164

Table 3.7 Unassigned contigs in Ace Lake metagenomes — their genetic and taxonomic composition……...168

Table 3.8 Turbidity values measured from different depths of Ace Lake in different time periods……...173

Table 4.1 List of Ace Lake metagenomes used for FR analysis of SynAce01…….187

Table 4.2 Marine cyanobacteria species used in the phylogenetic analysis of Ace Lake *Synechococcus*……...188

Table 4.3 Prokaryotic defence systems investigated in Ace Lake microbes……189

Table 4.4 Genes annotated on SynAce01 genomic regions associated with alignment gaps in *Synechococcus* MAGs……...197

Table 4.5 Genes annotated on variable coverage regions of SynAce01 genome…....202

Table 4.6 Defence genes annotated in *Synechococcus* MAGs……..209

Table 5.1 Ace Lake, Ellis Fjord, and Taynaya Bay metagenomes used for FR analyses of *Chlorobium* MAGs……233

Table 5.2 *Chlorobiaceae* family members used for phylogenetic analysis of Ace Lake, Ellis Fjord, and Taynaya Bay *Chlorobium*……236

Table 5.3 *Chlorobium* AL_ref MAG contigs……243

Table 5.4 Genes annotated on LCRs of AL_ref MAG……..244

Table 5.5 *Chlorobium* EF_ref MAG contigs…….258

Table 5.6 Genes annotated on variable coverage regions of EF_ref MAG……...259

Table 5.7 Defence genes annotated in *Chlorobium* MAGs……...266

Table 5.8 Spacers and repeats identified in CRISPR arrays of Ace Lake, Ellis Fjord, and Taynaya Bay *Chlorobium*……..270

Table 5.9 Host analysis of viral cluster and singletons with matches to Taynaya Bay *Chlorobium* spacers……...273

Table 5.10 Ace Lake *Chlorobium* low coverage genes associated with substrate transport……282

Table 5.11 Ace Lake *Chlorobium* low coverage genes associated with cobalamin biosynthesis and cobinamide and pseudocobalamin salvaging…….291

Table A1. List of Antarctic metagenomes…….359

Table A2. List of *Synechococcus* and *Chlorobium* MAGs from stratified systems in the Vestfold Hills…….367

Table E1. List of clade-specific markers added to MetaPhlAn2 database……591

Table F1. List of KO numbers associated with specific pathways……601

Table G1. List of abundant OTUs identified in Ace Lake……627

Table H1. List of specific viral contigs identified in Antarctic metagenomes……..638

Table H2. List of abundant viral clusters……..674

Table H3. List of *Chlorobium* spacer and repeat sequences…….676

Table I1. The physicochemical characteristics of Ace Lake and environmental characteristics of the Vestfold Hills, East Antarctica……687

List of Python Scripts

Code B1. Python code for Cavlab pipeline v1.2……...372

Code C1. Python code for Cavlab pipeline v4.1……...442

Code D1. Python code for arCOG pipeline v1.2……...566

List of Abbreviations

aa	amino acid
AADC	Australian Antarctic Data Centre
AAI	average amino acid identity
ABC	ATP-binding cassette
ABI	abortive infection
ANI	average nucleotide identity
arCOG	archaea COG
AsnA	ammonium-hydrolysing asparagine synthase
AsnB	glutamine-hydrolysing asparagine synthase
ATP	Adenosine triphosphate
BCAA	branched-chain amino acid
BclA	bacteriochlorophyll A
bp	base pair
BREX	bacteriophage exclusion
Cas	CRISPR-associated
CbiZ	adenosylcobinamide amidohydrolase
CNN	convolutional neural networks
CobA	corrinoid adenosyltransferase
CobP/CobU	adenosylcobinamide kinase/adenosylcobinamide-phosphate guanylyltransferase
COG	clusters of orthologous groups
CRISPR	clustered regularly interspaced short palindromic repeats
dbRDA	distance-based redundancy analysis
DIC	dissolved inorganic carbon
DISARM	defence island system associated with restriction–modification
distLM	distance-based linear model
DMB	5,6-dimethylbenzimidazole
DNA	deoxyribonucleic acid
DOC	dissolved organic carbon
EC	enzyme commission
ECF	energy-coupling factor

FmoA	Fenna-Matthews-Olson protein (bacteriochlorophyll A protein)
FR	fragment recruitment
GC	guanine cytosine content
GO	gene ontology
GSB	green sulfur bacteria
GTDB	Genome Taxonomy Database
GUI	graphical user interface
HGT	horizontal gene transfer
HTS	high-throughput sequencing
IGV	Integrative Genomics Viewer
IMG	Integrated Microbial Genomes
JCVI	J. Craig Venter Institute
JGI	Joint Genome Institute
kb	kilobase
KEGG	Kyoto Encyclopedia of Genes and Genomes
KO	KEGG Orthology
LCA	lowest common ancestor
LCR	low coverage region
MAG	metagenome-assembled genome
Mb	megabase
MEM	maximum-exact-match
N-ATPase	sodium-translocating ATPase
NCLDV	nucleocytoplasmic large DNA virus
OTU	operational taxonomic unit
PAR	photosynthetically active radiation
PCA	Principal component analysis
PCoA	Principal coordinates analysis
POC	particulate organic carbon
QC	quality control
R-M	restriction-modification
RefSeq	Reference Sequence
RNA	ribonucleic acid
rRNA	ribosomal RNA
rTCA	reverse tricarboxylic acid cycle

SIMPER	similarity percentage
SNP	single nucleotide polymorphism
SRB	sulfate-reducing bacteria
SSU	small subunit
T-A	toxin-antitoxin
TNF	tetra nucleotide frequency
tRNA	transfer RNA
UPGMA	unweighted pair group method with arithmetic mean

Disclosure Statement

Sections from this thesis have been published as:

1. Tschitschko B, Erdmann S, DeMaere MZ, Roux S, **Panwar P**, Allen MA, Williams TJ, Brazendale S, Hancock AM, Eloe-Fadrosh EA, Cavicchioli R. Genomic variation and biogeography of Antarctic haloarchaea. Microbiome. 2018;6:113.

2. **Panwar P**, Allen MA, Williams TJ, Hancock AM, Brazendale S, Bevington J, Roux S, Páez-Espino D, Nayfach S, Berg M, Schulz F, Chen IMA, Huntemann M, Shapiro N, Kyrpides NC, Woyke T, Eloe-Fadrosh EA, Cavicchioli R. Influence of the polar light cycle on seasonal dynamics of an Antarctic lake microbial community. Microbiome. 2020;8:116.

The parts of this thesis that were published and the contributions to the various chapters are as follows:

Chapter 2: The taxonomic diversity and abundance analysis of Megahit-assembled metagenomes from Deep Lake, Club Lake, and Rauer Island lakes using LAST and MEGAN-LR approach (section 2.3.1.3) were reported in publication **1**.

For the analyses in this chapter, a preliminary Cavlab pipeline v1.2 for metagenome analysis was provided by Bevington J (Appendix B). I tested various software and computational methods for the analysis of Antarctic metagenomic data; improved upon the overall code of the preliminary Cavlab pipeline; and generated Cavlab pipeline v4.1 (Appendix C) for Antarctic metagenome analysis. I wrote the arCOG pipeline v1.2 code (Appendix D) for the analysis of functional potential of archaeal sequences identified in metagenomes. Allen MA tested the PhyloSift runs on Katana as part of the preliminary Cavlab pipeline, assembled the JCVI-sequenced Ace Lake metagenomes using metaSPAdes, and provided the initial code for RefineM runs. Williams TJ helped in selecting the KO numbers associated with specific pathways for the KEGG analysis component of Cavlab pipeline v4.1 (Appendix F).

Chapter 3: The seasonal changes in Ace Lake environment (section 3.3.2), analysis of biodiversity and seasonal variation (sections 3.3.3 and 3.3.4), viral analyses (section 3.3.5), analysis of 'other' taxa and unassigned data (section 3.3.6), and KEGG-based functional potential analysis (section 3.3.7.1) were reported in publication **2**.

I performed the primary Ace Lake metagenome analyses using the Cavlab pipeline v4.1 to generate taxonomy, abundance, and COG and KEGG functional potential data. I also analysed taxonomic diversity, seasonal variations in OTU abundances, COG functional potential data, viral data (including identification of potential virus-host associations), 'unassigned contigs' data, and performed all statistical analyses. I provided the gene annotations of the abundant OTUs as well as the taxonomic affiliations of the KO numbers; Williams TJ analysed these sequences, verified their annotations, interpreted the functional potential of the OTUs, and identified the OTUs that contributed toward the abundance of KO numbers associated with specific pathways. Allen MA provided the MetaBAT-generated MAGs for taxonomy verification analysis as well as the VirSorter output of unassigned contig analysis. Páez-Espino D and Schulz F provided a list of viral contigs identified in the Antarctic metagenomes, including nucleocytoplasmic large DNA viral contigs and virophage contigs, as well as the IMG/VR spacer database containing matches of viral contigs to potential host spacers. Nayfach S and Roux S provided a list of viral contigs representing complete, circular phage genomes in metagenomes from Antarctic lakes (Appendix H: Table H1). Yau S and DeMaere M provided the cyanophage assembly (Appendix H: Table H1).

Chapter 4: The analysis of the presence/absence of CRISPR-Cas system genes in Ace Lake *Synechococcus* (section 4.3.6) was reported in publication **2**.

I performed all analyses in this chapter, including the verification of gene annotations and interpretation of data. The viral data were provided by Páez-Espino D, Schulz F, Nayfach S, and Roux S, as stated above for Chapter 3.

Chapter 5: The analysis of the presence/absence of CRISPR-Cas system genes in Ace Lake *Chlorobium* (section 5.3.5.1) as well as the analysis of seasonal variation in Ace Lake *Chlorobium* CRISPR spacer acquisition (section 5.3.5.2) were reported in publication **2**.

I performed all analyses in this chapter, including the verification of gene annotations and interpretation of data. Haque S extracted DNA from Taynaya Bay Sterivex samples

and performed QC filtration of the metagenomic reads. Allen MA performed read corrections and assembly of Taynaya Bay metagenomes. The viral data were provided by Páez-Espino D, Schulz F, Nayfach S, and Roux S, as stated above for Chapter 3.

1. General introduction

1.1 Antarctica

Antarctica is the coldest, driest continent on Earth and contains ~90% of the Earth's ice (AASSP, 2011). It covers nearly 14 million km^2 area, which increases to almost 20 million km^2 in winter. The Antarctic continent is divided into East and West Antarctica by the Transatlantic Mountains. Nearly 98% of the Antarctic continent is covered by an ice sheet of 2.2 km average thickness. Only ~0.4% of the continent, roughly 46,000 km^2, is ice-free (Cavicchioli, 2015; Chown et al, 2015). Antarctica is surrounded by Southern Ocean and most of its coastal ice is in the form of ice shelves (44%) and ice walls (38%) (Drewry, 1983). It is the southern-most continent of Earth containing the South Pole (90° S), with most of its coastal areas lying at lower latitudes reaching ~66° S; the northern-most tip of the Antarctic peninsula reaches ~63° S. The Antarctic light cycle includes a 24 h sunlight period in summer and a dark period with no sunlight in winter, with the duration of these light and dark periods varying from a few weeks to a few months depending on the latitude. Antarctica is also the windiest continent on Earth, with katabatic winds blowing off the continent at high velocity; the speed of wind gusts measured near the Antarctic coast sometimes exceed 200 kmh^{-1} (https://www.antarctica.gov.au/).

1.1.1 Antarctic lake biodiversity and ecology

Antarctica supports diverse life, including animals, plants, fungi, and a variety of microbes (Chown et al, 2015). Apart from penguins, albatrosses, and seals, which mainly inhabit the sub-Antarctic islands in the Southern Ocean, a rich assortment of lichens, bryophytes, and non-lichenised fungi can be found in Antarctica along with a few varieties of flowering plants, found only in the Antarctic peninsula (Peat et al, 2007; Bridge et al, 2008; Chown et al, 2015). Among the invertebrate organisms, tardigrades, nematodes, springtails, and mites are present in Antarctica (Stevens et al, 2006; Velasco-Castrillón et al, 2014). However, microbial communities show the most species diversity among all Antarctic life in a variety of habitats such as meltwater ponds (Archer et al, 2014), lake ice (Gordon et al, 2000), Antarctic soils (Cary et al, 2010; Fierer et al, 2012; Zablocki et al, 2014), stratified lakes (Lauro et al, 2011; Yau et

al, 2011; Yau et al, 2013; Laybourn-Parry and Bell, 2014), hypersaline lakes (Bowman et al, 2000a; DeMaere et al, 2013; Tschitschko et al, 2018), and other Antarctic aquatic systems (Laybourn-Parry and Pearce, 2007; López-Bueno et al, 2009; Wilkins et al, 2013; Cavicchioli, 2015).

In Antarctica, only ~0.4% of the total landmass is ice-free and harbours a variety of aquatic systems including lakes (both epiglacial and subglacial) and ponds (Cavicchioli, 2015; Chown et al, 2015). The Antarctic lake structure (microbial community types) and function (prevalent nutrient cycles) can depend on a number of factors such as availability of light, biotic factors (presence/absence of viruses), abiotic factors (availability of nutrients and oxygen, salinity, temperature), and other biogeographical and limnological factors (Figure 1.1; Cavicchioli, 2015). Single-celled eukaryotic algae and cyanobacteria are the most prominent primary producers in the photic zone of many Antarctic lakes, where they use light (energy source) and water (electron donor) for carbon fixation and oxygen production (Campbell, 1978; Williams, 1979; Wright and Burton, 1981; Franzmann et al, 1987; Rankin et al, 1999; Bowman et al, 2000a; Nadeau and Castenholz, 2000; Bell and Laybourn-Parry, 2003; Laybourn-Parry et al, 2005; Madan et al, 2005; Powell et al, 2005; Singh and Elster, 2007; Lauro et al, 2011; Kong et al, 2012; Yau et al, 2013; Williams et al, 2014). However, green sulfur bacteria (GSB) have been identified as important primary producers involved in anoxygenic photosynthesis at the oxic-anoxic interface of some Antarctic meromictic systems, where they use light (energy source) and hydrogen sulfide (electron donor) for carbon fixation, reducing hydrogen sulfide to elemental sulfur (Burke and Burton, 1988a; Bryant and Frigaard, 2006; Ng et al, 2010; Lauro et al, 2011). Due to cold temperature, the surfaces of many Antarctic aquatic systems are covered by ice for most of the year, which can impact the availability of light in the water column below the ice cover (described below in section 1.2.1). The amount of available light can further affect the abundance of microbial population, especially phototrophic microbes, as well as their function in an aquatic system. For example, the phytoflagellate *Pyramimonas gelidicola*, identified in two Antarctic lakes — Highway Lake and Ace Lake, has high abundance in summer when sufficient light is available for photoautotrophic growth, but has low abundance in winter when it resorts to phagotrophy for survival in the dark (Bell and Laybourn-Parry, 2003; Laybourn-Parry et al, 2005).

In Antarctic environments where light is not available, e.g., during winter or in the aphotic zones of lakes, chemoautotrophs have been identified as primary producers, utilizing inorganic compounds like nitrogen, sulfur or iron as energy sources, in place of light energy, for carbon fixation (Grzymski et al, 2012; Williams et al, 2012; Laybourn-Parry and Pearce, 2016). Chemoautotrophic archaea and bacteria including members of *Alphaproteobacteria*, *Betaproteobacteria*, *Gammaproteobacteria*, *Thaumarchaeota* have been reported in various Antarctic lakes (Sattley and Madigan, 2006; Mikucki and Priscu, 2007; Kong et al, 2012; Wilkins et al, 2013; Yau et al, 2013; Vick-Majors et al, 2014; Cavicchioli, 2015; Achberger et al, 2016; Laybourn-Parry and Pearce, 2016). Seasonal comparison of microbial diversity and function of some Antarctic environments (Lake Fryxell, Lake Bonney Western lobe, Antarctic peninsula coastal surface waters) showed a shift from photoautotrophy in summer when light is available to chemoautotrophy in dark winter (Grzymski et al, 2012; Williams et al, 2012; Vick-Majors et al, 2014). The chemoautotrophic archaea and bacteria in the Antarctic peninsula coastal waters use energy produced through oxidation of ammonia or nitrite, respectively, for carbon fixation (Grzymski et al, 2012; Williams et al, 2012). On the other hand, Lake Fryxell and Western lobe of Lake Bonney harbour chemoautotrophic bacteria that fix carbon using energy generated through sulfur oxidation (Sattley and Madigan, 2006; Kong et al, 2012).

Apart from photoautotrophs and chemoautotrophs, Antarctic aquatic systems contain heterotrophs that are involved in the conversion of complex organic compounds, including organic carbon generated by autotrophs, into inorganic molecules (Takacs et al, 2001; Mikucki and Priscu, 2007; Wilkins et al, 2013; Cavicchioli, 2015; Laybourn-Parry and Pearce, 2016). Unlike lower latitude lakes, Antarctic lakes receive very little exogenous nutrient input from their surrounding catchment areas, as the lakes are covered by ice for most of the year (Laybourn-Parry and Pearce, 2016). Therefore, the complex compounds utilised by heterotrophs mostly come from photoautotrophs and/or chemoautotrophs in the system (Matsumoto, 1989; McKnight et al, 1991; Laybourn-Parry and Pearce, 2016). Heterotrophic archaea and bacteria including members of *Actinobacteria*, *Alphaproteobacteria*, *Bacteroidetes*, *Betaproteobacteria*, *Chloroflexi*, *Gammaproteobacteria*, *Haloarchaea*, *Sphingobacteria* have been observed in various Antarctic lakes (Mikucki and Priscu, 2007; Mosier et al, 2007; Lauro et al, 2011;

DeMaere et al, 2013; Wilkins et al, 2013; Yau et al, 2013; Vick-Majors et al, 2014; Cavicchioli, 2015; Laybourn-Parry and Pearce, 2016).

As metazoan grazers of phytoplankton, bacteria and archaea are very few in the Antarctic lakes, viruses seem to play an important role in nutrient mobilization and in driving the evolution of hosts, thereby affecting lake ecology (Figure 1.1) (Kepner et al, 1998; Pearce and Wilson, 2003; Madan et al, 2005; Säwström et al, 2007; Anesio and Bellas, 2011; Lauro et al, 2011; Yau et al, 2011; Cavicchioli, 2015; Tschitschko et al, 2015; Laybourn-Parry and Pearce, 2016). The presence of strong wind in Antarctica has also been suggested to play a role in shaping microbial communities through the dispersal of microbes in the continent, just as it does in other ecosystems across the globe (Wilkins et al, 2013; Cavicchioli, 2015).

The chemical composition of the aquatic systems such as their salinity, oxygen content, and nutrient composition and concentration can govern Antarctic lake microbial communities (Figure 1.1; Cavicchioli, 2015). For example, Deep Lake, Organic Lake and Ekho Lake are three highly saline (hypersaline) lakes in the Vestfold Hills and have been shown to contain similar high abundance populations of *Gammaproteobacteria* and members of *Cytophaga-Flavobacterium-Bacteroidetes* group, along with low abundance populations of *Actinobacteria*, *Alphaproteobacteria* and *Firmicutes*; a majority of the *Gammaproteobacteria* belonging to the genus *Marinobacter* (Bowman et al, 2000a). Similarly, methanogenic archaea have been identified in the dark, anoxic waters of various stratified lakes in the Vestfold Hills (Bowman et al, 2000b; Lauro et al, 2011), whereas a diverse population of haloarchaea thrives in the hypersaline lakes from the Vestfold Hills and the Rauer Islands in East Antarctica (DeMaere et al, 2013; Tschitschko et al, 2018).

The biogeographic locations of Antarctic lakes have also been shown to affect the type of microbial communities observed in systems with similar physicochemical compositions. A dominant population of *Chlorobiaceae* family members (GSB) along with a low abundance population of *Chromatiaceae* family members (purple sulfur bacteria) are prevalent in various meromictic lakes in the Vestfold Hills (Burke and Burton, 1988a), but the meromictic Lake Fryxell in McMurdo Dry Valleys, East Antarctica supports members of *Chloroflexi* (green non-sulfur bacteria) and a diverse population of purple non-sulfur bacteria (Karr et al, 2003). Overall, the Antarctic lake microbial ecosystem is very diverse and can be shaped by environmental factors (e.g.,

temperature, light availability, wind) as well as biotic (e.g., presence/absence of viruses), physical (e.g., presence/absence of ice cover), chemical (e.g., salinity, nutrients), and geographical (location) characteristics of the aquatic systems in which the microbes reside (Figure 1.1).

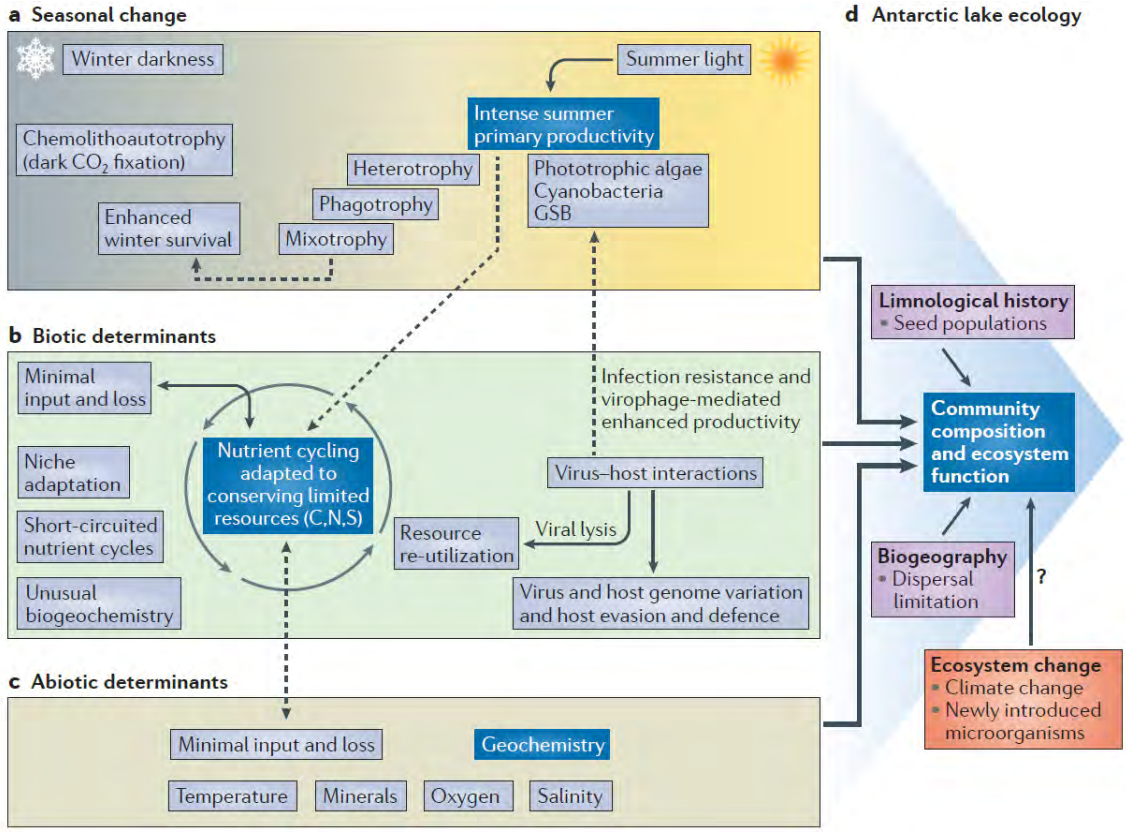


Figure 1.1 Various factors affecting Antarctic lake ecology. The availability of light **(a)**, and biotic **(b)**, abiotic **(c)** and other factors **(d)** that potentially impact Antarctic lake structure and function are shown. The schematic was taken from Cavicchioli (2015).

1.1.2 Physical characteristics of stratified aquatic systems in the Vestfold Hills, Antarctica

The Vestfold Hills lie in East Antarctica and are mostly free of ice and snow; they are classified as an Antarctic oasis. They were formed as a result of isostatic rebound (i.e., uplifting of the landmass) after the retreat of the continental ice sheet nearly 10,000 years ago, due to which the Vestfold Hills are riddled with thousands of supra- and sub-glacial water bodies including fresh water, saline, and hypersaline systems (Gibson, 1999; Cavicchioli, 2015; Siegert et al, 2016). The Vestfold Hills are well-known for their variety of stratified aquatic systems, with at least 34 stratified lakes and marine basins reported in one study (Gibson, 1999). Stratified systems are also referred to as meromictic systems — they have a well-mixed oxic mixolimnion, an oxic-anoxic

interface, and an anoxic monimolimnion. In the Vestfold Hills, the permanent stratification of lakes can be attributed to the presence of a protective ice cover that remains for most of the year and prevents wind-driven mixing of the lake waters (Burton and Barker, 1979; Burch, 1988; Burke and Burton, 1988a; Gibson and Burton, 1996). In summer, the melting ice cover and the inflow of melt water can create a layer of freshwater on the surface of a stratified lakes, as is seen in Ace Lake in the Vestfold Hills (Hand and Burton, 1981). The ice cover of most stratified lakes in the Vestfold Hills completely melts by the end of December, whereas in some low salinity lakes, the ice cover does not melt even by January (Gibson and Burton, 1996). Although the melting of the ice cover exposes the stratified lakes to wind-driven mixing, the lake waters mix to a depth of only a few metres, partly due to the stability provided by thermal and chemical stratification of the lakes and partly because of the presence of the additional fresher water layer on their surface (Walker, 1974; Burton and Barker, 1979; Burch, 1988). As the ice cover reforms with approaching winter, salt from the newly forming ice is exuded into the surrounding lake waters as brine, which sinks deeper and drives the convective mixing of the mixolimnion waters of the lake (Gibson and Burton, 1996; Swadling, 1998; Rankin et al, 1999). The depth to which the mixing occurs depends on the amount of salt excluded from the newly forming ice, which in turn is affected by the thickness of the ice cover formed (Gibson and Burton, 1996; Rankin et al, 1999). Due to the strong salinity gradient below the halocline, the sinking brine mingles with the monimolimnion waters mainly by diffusion, precluding any mixing (Canfield and Green, 1985; Rankin et al, 1999).

1.2 Ace Lake — a stratified lake in the Vestfold Hills, Antarctica

Ace lake is a marine-derived, stratified lake located in the Vestfold Hills in East Antarctica (68.473° S, 78.189° E) (Figure 1.2). With a maximum depth of 25 m, the Ace Lake water column is segregated into oxic mixolimnion and anoxic monimolimnion by an oxic-anoxic interface (Figure 1.3) (Burton, 1980; Gibson, 1999; Rankin et al, 1999). The lake surface is covered with ice for most of the year and melts only in summer (Burton, 1980). Apart from the oxygen gradient, Ace Lake also has a salinity gradient, which increases with lake depth — the lake salinity levels vary from 1.9 % at around 1 m depth to 4.2 % at around 24 m depth (Burton and Barker, 1979;

Hand and Burton, 1981; Burch, 1988; Bell and Laybourn-Parry, 1999; Rankin et al, 1999; Lauro et al, 2011; Panwar et al, 2020). The pH of Ace Lake waters steadily decreases with depth, being slightly alkaline in the oxic zone and nearly neutral in the anoxic zone (Burton and Barker, 1979; Hand and Burton, 1981; Rankin et al, 1999; Lauro et al, 2011). The lake pH also fluctuates with season in the oxic zone, with more variations observed at 2 m depth than at 10 m (Rankin et al, 1999). The seasonal heat transfer to and from the lake, i.e., gain of heat through solar radiation or surrounding rocks and loss of heat through the ice cover or into the lake sediment, has led to the thermal stratification of Ace Lake. However, temperature data collected from Ace Lake over more than three decades suggests that the thermal stratification of the lake is not as prominent as it was in the past (Burton and Barker, 1979; Hand and Burton, 1981; Burch, 1988; Bell and Laybourn-Parry, 1999; Rankin et al, 1999; Lauro et al, 2011; Panwar et al, 2020). Nonetheless, the presence of an ice cover and a strong salinity gradient allow for the permanent stratification of Ace Lake (Walker, 1974; Burton and Barker, 1979; Burch, 1988; Rankin et al, 1999).

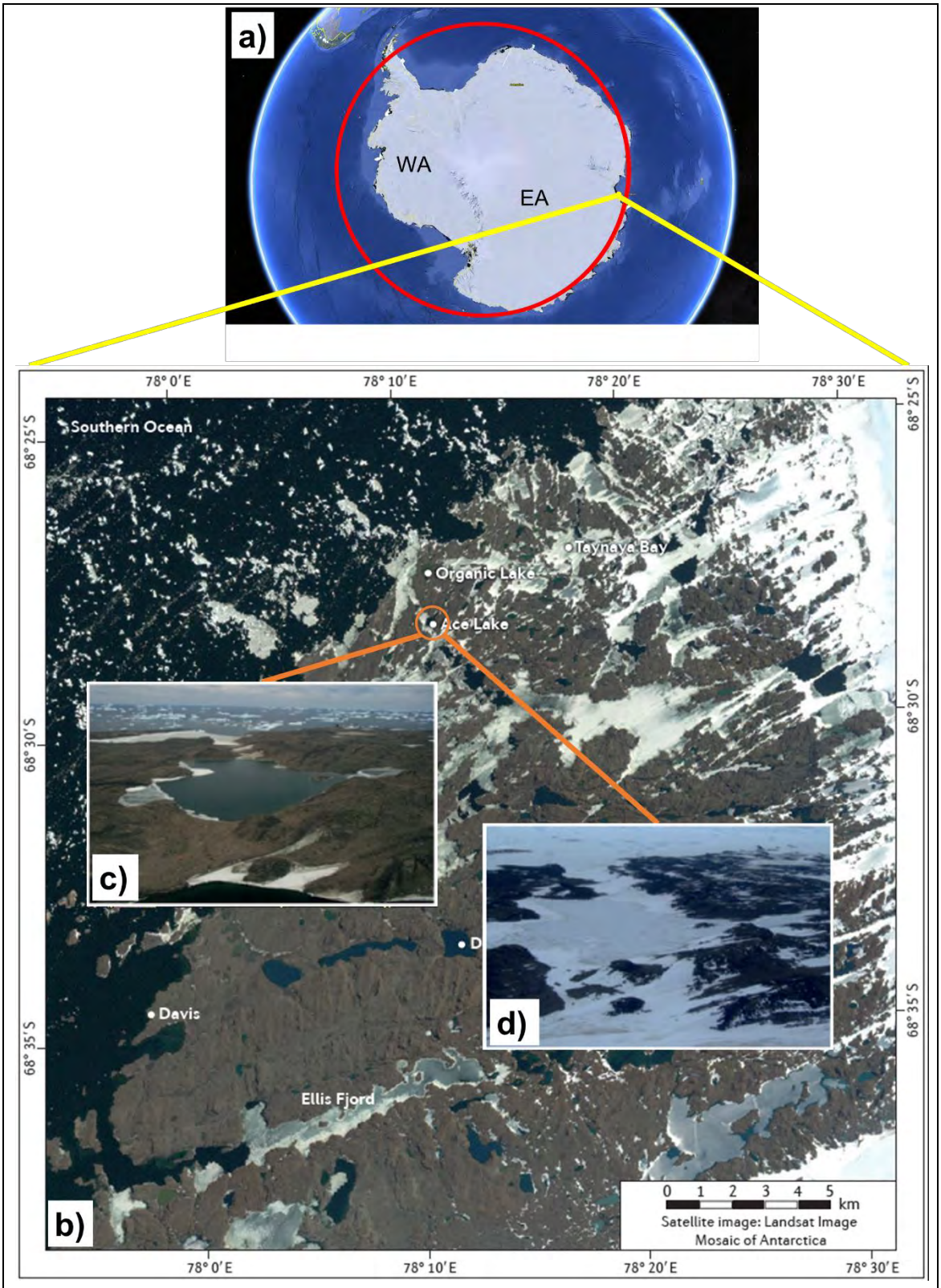


Figure 1.2 Ace Lake in the Vestfold Hills, East Antarctica. The Antarctic continent **(a)** is the southern-most landmass on Earth and mostly lies within the Antarctic Circle (~66° S; red circle in **a**). It is divided into East Antarctica (EA) and West Antarctica (WA) by the Transatlantic Mountains. The Vestfold Hills **(b)** lie along the coastal region of East Antarctica and cover approximately 411 km^2 area, containing ~3,000 aquatic systems including Ace Lake (orange

circle in **b**) (Cavicchioli, 2015). Ace Lake is covered by ice for most of the year **(d)**, which melts in summer **(c)**. The image of the Antarctic continent **(a)** was created in Google Earth Pro (https://www.google.com/earth/). The image showing the satellite view of majority of the Vestfold Hills **(b)** as well as the photographs of Ace Lake **(c, d)** were adapted from Cavicchioli (2015).

In Ace Lake, the oxycline, halocline, and thermocline lie across almost the same lake depths, but this has not always been the case (Table 1.1). Earlier reports show that the oxic-anoxic interface of Ace Lake was at 10 m depth in 1974 (Burton and Barker, 1979), at 12 m depth in 1992 (Rankin et al, 1999), and at 13 m depth 1996 (Bell and Laybourn-Parry, 1999). Moreover, based on the concentration of manganese and selenium in Ace Lake, two trace elements found in high concentration near the oxycline of meromictic Antarctic lakes (Masuda et al, 1988), it has been speculated that the Ace Lake oxic-anoxic interface could have been at a depth as low as 18 m at some point in the past (Rankin et al, 1999). On the other hand, the halocline and thermocline, which generally coincide in Ace Lake, were at depths higher up in the water column in the oxic zone until less than two decades ago, when the halocline/thermocline dropped to the level of the oxycline in Ace Lake (Table 1.1). These shifts in the physicochemical gradients in Ace Lake might be due to a change in lake water level, because of inflow of melt water, melting of ice cover, and/or evaporation of lake water, or might indicate the result of ice formation, which impacts the depth to which the mixolimnion extends. It has been speculated that if the halocline of a stratified lake were to be pushed down to a low enough depth, due to reduction in lake water level or during formation of an ice cover, the lake would completely mix removing any physicochemical gradients prevalent in the water column (Gibson and Burton, 1996; Rankin et al, 1999). Such a turn over event has been suggested to have occurred in Ace Lake in the past since its isolation, during which most of the sulfur (76%) was lost from the lake (Burton and Barker, 1979; Gibson and Burton, 1996; Rankin et al, 1999).

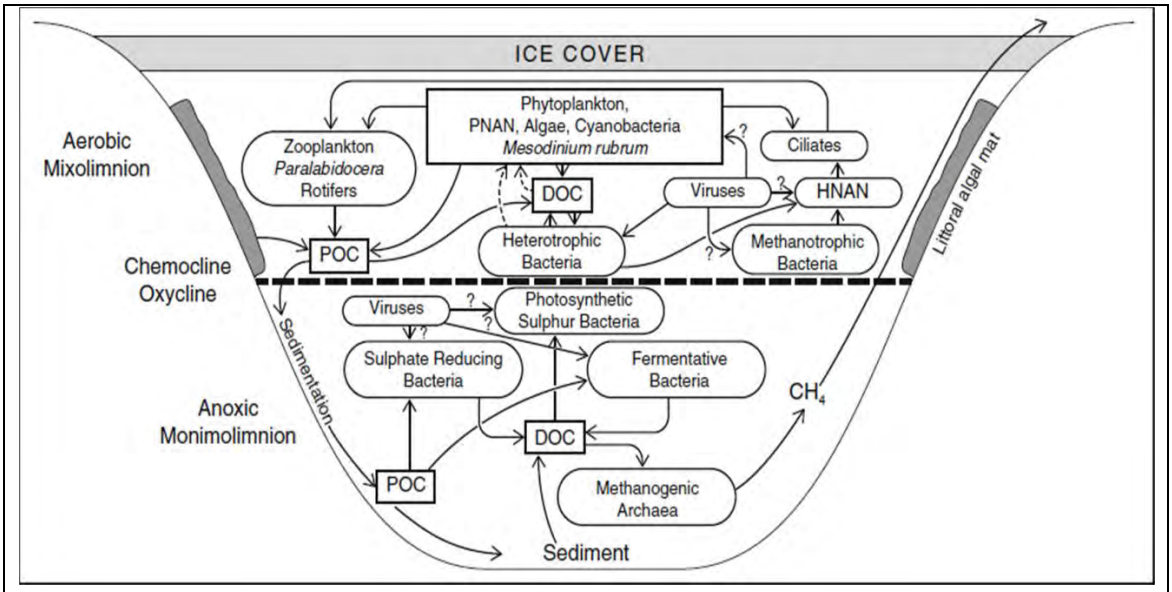


Figure 1.3 The physical, chemical, and biological structuring of Ace Lake. Ace Lake is a stratified lake of marine origin, with an upper oxic zone (Aerobic Mixolimnion), a chemocline/oxycline (Chemocline Oxycline), and a lower anoxic zone (Anoxic Monimolimnion). The schematic was taken from Laybourn-Parry et al (2014).

Table 1.1 Changes in the position of the oxic-anoxic interface, halocline, and thermocline in Ace Lake over a period of more than three decades. The values in the table represent the Ace Lake depths at which the oxic-anoxic interface, halocline, and thermocline of the lake waters were measured. [A] The data taken from various studies span more than three decades and are shown chronologically from top to bottom.

Data collection date	Ace Lake depths			Reference[A]
	Oxic-anoxic interface	Halocline	Thermocline	
Nov 1975	10 m	5–7 m	7–10 m	Burton and Barker, 1979
Dec 1977	9 m	-	7–10 m	Hand and Burton, 1981
Feb 1979	9 m	3–5 m	4–5 m	Burch, 1988
Aug 1979	9 m	3–7 m	4–7 m	
Nov 1992	12 m	7–9 m	7–9 m	Rankin et al, 1999
June 1994	12 m	7–9 m	7–9 m	
Feb 1996	12 m	6–8 m	6–8 m	Bell and Laybourn-Parry, 1999
Oct 1996	13 m	8.5–9.5 m	8.5–9.5 m	
Feb 2001 to Feb 2002	12 m (first summer) 14 m (winter)	-	-	Laybourn-Parry and Bell, 2014

	14 m (second summer)			
Dec 2002 to Dec 2003	12 m (summer and winter)	-	-	
Dec 2006	13 m	12–13 m	11–13 m	Lauro et al, 2011; Panwar et al, 2020
Nov 2008	13 m	12–13 m	12–13 m	Panwar et al, 2020
Nov 2013	14 m	12–14 m	12–14 m	
Aug 2014	15 m	12.5–15 m	12–14 m	
Oct 2014	14 m	12–14 m	12–14 m	
Dec 2014	14 m	12–14.5 m	12–14 m	

1.2.1 Light penetration

The polar light cycle is distinct from the light cycle experienced in non-polar cold environments, with 24 h of sunlight in summer and no sunlight in winter; the light and dark periods lasting a few weeks in the Vestfold Hills. In 1979, incident light as high as 1,225 $\mu Em^{-2}s^{-1}$ was measured in summer, and as low as 1.3 $\mu Em^{-2}s^{-1}$ was measured in winter at the surface of Ace Lake (Burch, 1988). As the lake is covered by ice for most of the year, the lake depth to which the incident light can penetrate depends on the opaqueness, thickness, and age of the ice cover (Kirk, 1994) as well as the presence/absence and thickness of a snow cover (Burch, 1988). In ice-free conditions in summer, only 10% of the incident light reaches 9 m depth in Ace Lake and light penetrates up to 11.5 m depth (Hand and Burton, 1981; Burch, 1988; Rankin, 1998). However, with a 2 m ice cover in spring, only 1% of the incident light reaches 9 m depth and no light is available past 10 m depth in the lake. The presence of dense populations of phototrophic bacteria in the mixolimnion and oxycline of Ace Lake also prevents light from penetrating beyond 11.5 m depth in ice-free conditions (Rankin et al, 1999). Apart from ice cover, the presence of a snow cover can further impede light penetration — the amount of incident light penetrating through a 1.6 m ice cover with a 30 cm snow cover is three times less than the light penetration through the same ice thickness but without a snow cover (Burch, 1988). Ice and snow cover not only affect the amount of light, but also the wavelength of light that penetrates the water column. Ice cover attenuates red light much more than blue or green light, whereas the presence

of a snow cover causes a stronger attenuation of green and blue lights but not red light (Burch, 1988). Generally, in the presence of a thick ice cover, green light penetrates deeper into the water column in Ace Lake, than blue or red light.

1.2.2 Biodiversity

Life in Ace Lake have been extensively studied by various research groups since the 1970s, (i) to understand the overall biological composition of this meromictic lake (Burton and Barker, 1979; Hand, 1980; Hand and Burton, 1981; Burch, 1988; Burke and Burton, 1988a; Bell and Laybourn-Parry, 1999; Bowman et al, 2000b; Laybourn-Parry et al, 2001; Coolen et al, 2004a; Coolen et al, 2004b; Laybourn-Parry et al, 2005; Madan et al, 2005; Powell et al, 2005; Coolen et al, 2006; Lauro et al, 2011) or (ii) to describe specific organisms identified in the system (Franzmann et al, 1991a; Franzmann and Rohde, 1991; Franzmann and Dobson, 1992; Franzmann et al, 1992; Bowman et al, 1997; Franzmann et al, 1997; Rankin, 1998; Bell and Laybourn-Parry, 2003; Ng et al, 2010). Generally, the biodiversity of each stratum of Ace Lake, i.e., the oxic mixolimnion, the oxycline, and the anoxic monimolimnion, is distinct (Table 1.2).

Table 1.2 The biodiversity of Ace Lake assessed over a period of more than three decades. Data were taken from various studies on Ace Lake, shown here chronologically from top to bottom. [A] The second column indicates the method used to analyse diversity. [B] The last column describes the types of life forms identified in each stratum of Ace Lake: U, upper oxic zone (mixolimnion); I, oxycline (oxic-anoxic interface); L, lower anoxic zone (monimolimnion); S, anoxic zone sediment.

Reference study	Method[A]	Biodiversity[B]
Burton and Barker, 1979	Microscopy	**U:** *Paralabidocera antarctica* and *Acartia* sp. (copepods); a branched filamentous *Chlorophyta* species; cyanobacteria; *Fragilaria* sp. and *Navicula* sp. (diatoms) **I and L:** high microbial cell count
Hand and Burton, 1981	Microscopy; cell culture and characterization	**U:** *P. antarctica* (a copepod); *Pyramimonas* sp. (a green alga); cyanobacteria; diatoms **I:** heterotrophs **L:** *Chromatium* sp. and *Rhodospirillum* sp. (phototrophic bacteria); *Desulfovibrio* sp.

		(sulfate-reducing bacteria); anaerobic heterotrophs; methanogens
Burch, 1988	Microscopy	**U:** *Pyramimonas gelidicola*; *Cryptomonas* sp.; two unknown species of flagellates; *Navicula* sp. and *Pinnularia* sp. (diatoms); cyanobacteria
Burke and Burton, 1988a	Microscopy; enrichment culture and morphology characterization	**I:** dominated by *Chlorobium vibrioforme* and *Chlorobium limicola* (green-coloured GSB); few members of *Chromatiaceae* family and *Rhodospirillaceae* family (purple photosynthetic bacteria) also identified
Mancuso et al, 1990	Lipid mass spectrometry	**U:** microeukaryotes **L:** *Desulfobacter* sp.; *Desulfovibrio* sp.; methanogenic bacteria
Franzmann et al, 1991a	Celle culture, isolation, and characterization; *16S rRNA* gene-based phylogenetic analysis	**L:** *Carnobacterium funditum*, *Carnobacterium alterfutulitum*
Franzmann and Rohde, 1991	Celle culture, isolation, and characterization	**L:** an obligate anaerobic coiled bacterium
Franzmann and Dobson, 1992	Cell culture and isolation; *16S rRNA* gene-based phylogenetic analysis	**L:** an anaerobic wall-less spirochete
Franzmann et al, 1992	Enrichment culture, isolation, and characterization; *16S rRNA* gene-based phylogenetic analysis	**L:** *Methanococcoides burtonii* (a methylotrophic methanogen)
Bowman et al, 1997	Enrichment culture, isolation, and characterization; *16S rRNA* gene-based phylogenetic analysis	**U (just above I):** *Methylosphaera hansonii* (a methanotroph)
Franzmann et al, 1997	Enrichment culture, isolation, and	**L:** *Methanogenium frigidum* (a hydrogenotrophic methanogen)

	characterization; *16S rRNA* gene-based phylogenetic analysis	
Bell and Laybourn-Parry, 1999	Microscopy	**U:** *P. antarctica*; *Notholca* (a rotifer); *Py. gelidicola*; *Cryptomonas* sp.; *Chlamydomonas* sp.; few diatoms; *Gyrodinium* sp. and *Gymnodinium* sp. (dinoflagellates); *Mesodinium rubrum*; *Euplotes* sp. (a ciliate) **I and L (just below I):** *P. antarctica*; phototrophic bacteria; sulfate-reducing bacteria
Rankin, 1998	Microscopy; cell culture and isolation; *16S rRNA* gene-based phylogenetic analysis	**U:** *Synechococcus* sp. (a marine cyanobacteria)
Bowman et al, 2000b	*16S rRNA* gene-based phylogenetic analysis	**S:** *Desulfosarcina*; *Syntrophus*; *Prochlorococcus* (probably dead cells settled from mixolimnion); a wall-less spirochete; other anaerobic bacteria; *Methanosarcina*; other members of *Euryarchaeota*
Laybourn-Parry et al, 2001	Microscopy	**U:** Viruses
Madan et al, 2005	Microscopy	**U:** *Py. gelidicola*; *Cryptomonas* sp.; *Chlamydomonas* sp.; *Gyrodinium lachrymal*, *Gonyaulax* sp., *Protoperidinium* sp., and *Gymnodinium* sp. (dinoflagellates), *M. rubrum*; viruses
Powell et al, 2005	Flow cytometry; cell culture and characterization; *16S rRNA* gene-based phylogenetic analysis	**U:** *Synechococcus* sp.; *M. rubrum*; *Py. gelidicola*; *Cryptomonas* sp.; a phototropic nanoplankter

Coolen et al, 2006	*16S rRNA* gene-based phylogenetic analysis; lipid chromatography	**I:** *Chlorobium* sp. **S:** members of *Methanosarcinales* (methanogenic archaea)
Ng et al, 2010	Metaproteogenomic analysis; *16S rRNA* gene-based phylogenetic analysis	**I:** *Chlorobium* sp. (referred to as C-Ace)
Lauro et al, 2011	Metaproteogenomic analysis; *16S rRNA* gene-based phylogenetic analysis; read taxonomic classification	**U:** *Mantoniella* sp. (a green alga); members of *Phycodnaviridae* (algal viruses); *Synechococcus* sp.; members of SAR11 clade, *Flavobacteria*, *Alphaproteobacteria,* and *Deltaproteobacteria* **I:** *Chlorobium* sp.; members of *Deltaproteobacteria* (sulfate-reducing bacteria) and *Gammaproteobacteria* **L:** members of *Gammaproteobacteria*, *Deltaproteobacteria*, *Epsilonproteobacteria*, Firmicutes, and *Euryarchaeota*; members of Candidate divisions OD1 and OP11 (bacterial candidate phyla); members of *Siphoviridae*, *Myoviridae*, and *Podoviridae* (bacteriophage)

1.2.2.1 Mixolimnion

In Ace Lake, the calanoid copepod *Paralabidocera antarctica* is the most prominent zooplankter (Burton and Barker, 1979; Hand and Burton, 1981; Bell and Laybourn-Parry, 1999; Lauro et al, 2011). Another calanoid copepod (*Acartia* sp.) and a harpacticoid copepod (*Idomene scotti*) were also identified in the Ace Lake oxic waters and benthic mats, respectively (Burton and Barker, 1979; Rankin et al, 1999). Eukarya capable of photosynthesis, including members of *Chlorophyta* (*Mantoniella*, *Chlamydomonas* sp., *Py. gelidicola*), the photosynthetic ciliate *Mesodinium rubrum*, and a *Cryptomonas* sp., are present in the oxic waters of Ace Lake (Burton and Barker, 1979; Hand and Burton, 1981; Burch, 1988; Bell and Laybourn-Parry, 1999; Madan et al, 2005; Powell et al, 2005; Lauro et al, 2011). Of these, *Py. gelidicola* is a mixotroph

and can survive in limited light by feeding on bacteria or dissolved organic carbon (DOC) in the lake waters (Bell & Laybourn-Parry, 2003; Laybourn-Parry et al, 2005; Madan et al, 2005), whereas the cryptophyte (*Cryptomonas* sp.) is likely to be preyed upon by *M. rubrum* (Nishitani and Yamaguchi, 2018). A large population of algal viruses (*Phycodnaviridae*) is also present in the oxic zone of Ace Lake, which probably preys on the green algae in the mixolimnion (Lauro et al, 2011). A few diatoms (*Fragilaria* sp., *Navicula* sp., *Pinnularia* sp.) have been identified, but their presence in the oxic lake waters has been attributed to diffusion from littoral algal mats (Burton and Barker, 1979; Hand and Burton, 1981; Burch, 1988). Other members of the benthic mat community of Ace Lake include green algae (*Urospora penicilliformis*, *Rhizoclonium implexium*), brown algae (*Ectocarpus* sp.), cyanobacteria, ciliates (a large tube dwelling member of *Folliculinidae* family), a platyhelminthe, nematodes, and three rotifer species (Dartnall, 2000). The bacterial population in the Ace Lake oxic zone is dominated by a cyanobacteria (*Synechococcus*), which is considered to be responsible for the oxygenation of the mixolimnion (Burton and Barker, 1979; Hand and Burton, 1981; Burch, 1988; Rankin, 1998; Powell et al, 2005; Lauro et al, 2011). A high abundance of the members of SAR11 clade (*Pelagibacterales*) has also been reported in this zone, which is consistent with the marine origin of Ace Lake (Lauro et al, 2011). Apart from these microbes, a methanotrophic bacteria, *Ms. hansonii*, is present in the oxic zone of Ace Lake, at a depth just above the oxycline (Bowman et al, 1997). The overall community structure of the Ace Lake mixolimnion is similar to that of marine surface environments, but with ten-fold lower species richness (Lauro et al, 2011).

1.2.2.2 Oxycline

The oxycline of Ace Lake is dominated by *Chlorobium* (green-coloured GSB), the most abundant organism in the lake (Burke and Burton, 1988a; Coolen et al, 2006; Ng et al, 2010; Lauro et al, 2011). Microscopy and cell culture studies have identified *Chlorobium vibrioforme* and *Chlorobium limicola* as the two most abundant *Chlorobium* species in the lake (Burke and Burton, 1988a), but a more recent metagenomic analysis of the Ace Lake oxyline waters has shown that the clonal population of a *Chlorobium* dominates this zone (Lauro et al, 2011). A few studies have indicated that the Ace Lake *Chlorobium* is closely related to *Chlorobium phaeovibrioides*, another GSB from a marine environment (Coolen et al, 2006; Ng et al, 2010). Other than *Chlorobium*, some members of the *Chromatiaceae* and

Rhodospirillaceae families have also been identified in the Ace Lake oxycline, although their abundance is much lower than *Chlorobium* (Hand and Burton, 1981; Burke and Burton, 1988a). A number of sulfate-reducing bacteria (SRB), such as *Desulfovibrio* sp., *Desulfobacter* sp., and other *Deltaproteobacteria* members, are present in the Ace Lake oxycline alongside the *Chlorobium* (Hand and Burton, 1981; Mancuso et al, 1990; Lauro et al, 2011).

1.2.2.3 Monimolimnion

The anoxic waters of Ace Lake support a diverse community of anaerobic bacteria, methanogenic archaea, and bacteriophages. The anaerobic bacteria identified in Ace Lake include *Carnobacterium funditum*, *Carnobacterium alterfutulitum*, members of *Gammaproteobacteria*, *Deltaproteobacteria*, and *Epsilonproteobacteria*, members of bacterial candidate phyla (Candidate divisions OD1 and OP11), as well as a coiled bacterium and a wall-less spirochete (Hand and Burton, 1981; Franzmann et al, 1991a; Franzmann and Rohde, 1991; Franzmann and Dobson, 1992; Lauro et al, 2011). SRB are also present in the anoxic zone of Ace Lake, although not in the deepest depths of the lake (Hand and Burton, 1981; Mancuso et al, 1990; Bell and Laybourn-Parry, 1999; Lauro et al, 2011). The monimolimnion supports a population of methanogenic archaea including *Methanococcoides burtonii* and *Methanogenium frigidum* (Hand and Burton, 1981; Mancuso et al, 1990; Franzmann et al, 1992; Franzmann et al, 1997; Coolen et al, 2004a; Coolen et al, 2006; Lauro et al, 2011). Studies of the Ace Lake sediment samples indicate that members of *Methanosarcinales* order (including a *Methanosarcina* sp.) are present in the anoxic zone of Ace Lake along with some *Deltaproteobacteria* such as *Desulfosarcina* and *Syntrophus* (Bowman et al, 2000b; Schouten et al, 2001; Coolen et al, 2004a; Coolen et al, 2006). A variety of bacteriophage belonging to the *Siphoviridae*, *Myoviridae*, and *Podoviridae* families of double-stranded DNA viruses have been identified in the anoxic zone of Ace Lake (Lauro et al, 2011).

1.2.3 Water chemistry and nutrient cycling

1.2.3.1 Carbon

In lakes, autotrophs fix dissolved inorganic carbon (DIC) such as carbon dioxide, carbonates and bicarbonates to produce organic carbon (dissolved and particulate) using light energy (photoautotrophs) or energy generated through inorganic nitrogen, sulfur,

and iron compounds (chemoautotrophs) (Alin and Johnson, 2007). The organic carbon compounds are used by autotrophs and heterotrophs for respiration resulting in carbon dioxide production (Alin and Johnson, 2007). Near lake surface the respired carbon dioxide can be lost to the atmosphere through gaseous exchange, but near lake bottom the carbon dioxide can remain stored as DIC for long periods, making it available to methanogens and chemoautotrophs (Alin and Johnson, 2007). Methane produced by methanogens can either be lost to the atmosphere through diffusion and gaseous exchange or it can be utilised by methane-oxidising bacteria (Bastviken et al, 2008; Hofmann et al, 2010). Excessive organic carbon often sinks to the lake bottom and is buried in sediments over time (Alin and Johnson, 2007).

Ace Lake is covered by ice for most of the year, leaving little chance for exogenous nutrient input, except during the ice-free periods in summer. The concentration of DIC is high throughout Ace Lake, such that the lake waters are supersaturated with inorganic carbon compared to the atmosphere (Burton, 1980). As significant levels of carbon monoxide dehydrogenase were present throughout Ace Lake, carbon monoxide oxidation might be an important pathway for energy production, which might explain the high concentration of DIC in the lake (Lauro et al, 2011). Due to this, it has been speculated that Ace Lake probably loses carbon to the atmosphere during ice-free periods (Burton, 1980; Rankin et al, 1999). In the mixolimnion, the green algae and cyanobacteria are the major primary producers capable of assimilating DIC (Rankin et al, 1999). On the other hand, the members of *Flavobacteria* and *Gammaproteobacteria* can degrade particulate organic carbon (POC) to DOC. The members of *Actinobacteria* and SAR11 clade in the oxic zone of Ace Lake have the capacity to use DOC (Lauro et al, 2011), which is consistent with the low concentration of DOC in the oxic zone compared to the anoxic zone of Ace Lake (Hand and Burton, 1981; Bell and Laybourn-Parry, 1999; Rankin et al, 1999; Lauro et al, 2011). The DOC concentration in Ace Lake has also been shown to fluctuate seasonally (Bell and Laybourn-Parry, 1999; Madan et al, 2005; Laybourn-Parry et al, 2007). At the oxycline of Ace Lake, anaerobic carbon fixation by the *Chlorobium* and the SRB contributes toward the carbon cycle in this zone (Lauro et al, 2011). SRB, together with fermentative bacteria and methanogens, are also involved in the anaerobic degradation of POC (sinking particulate matter produced in the mixolimnion) in the anoxic zone of Ace Lake (Burton and Barker, 1979; Burton, 1980; Franzmann et al, 1988, Mancuso et al, 1990;

Franzmann et al, 1991b; Franzmann and Dobson, 1992; Rankin et al, 1999; Lauro et al, 2011). Dissolved methane is absent in the oxic zone of Ace Lake, but its concentration increases around the oxycline and reaches very high concentration in the anoxic zone (Franzmann et al, 1991b). This accumulation of methane in the Ace Lake monimolimnion has been attributed to methane production by methanogenic archaea (members of *Euryarchaeota*) along with the absence of anaerobic methanotrophs in the anoxic zone and low potential for aerobic methane oxidation (Franzmann et al, 1991b; Lauro et al, 2011). Some of the methane diffuses to the lake surface and is lost to the atmosphere in ice-free periods, but some of it can be utilised by methanotrophic bacteria present in the oxic zone just above the oxycline (Bowman et al, 1997).

1.2.3.2 Sulfur

Microbial sulfur cycling involves redox reactions usually associated with sulfate, the most commonly found sulfur form in lakes (Holmer and Storkholm, 2001; Luo, 2018; Jørgensen et al, 2019). Some microbes can reduce sulfate to hydrogen sulfide and organic sulfur, and assimilate them for biosynthetic purposes (assimilatory sulfate reduction), while others such as anaerobic bacteria can reduce sulfate to hydrogen sulfide for energy production (dissimilatory sulfate reduction) (Hordijk, 1993; Holmer and Stockholm, 2001; Jørgensen and Kasten, 2006; Luo, 2018). Oxidation of hydrogen sulfide to organic sulfur, elemental sulfur, sulfite and/or sulfate can be driven by microbes such as sulfur oxidizing bacteria (Holmer and Stockholm, 2001; Jørgensen and Kasten, 2006; Luo, 2018; Jørgensen et al, 2019). Some chemoautotrophs use the energy generated during sulfur oxidation for carbon fixation (Sattley and Madigan, 2006; Kong et al, 2012). Apart from sulfur reduction and oxidation, various microbes, including some SRB, are capable of disproportionating inorganic sulfur into hydrogen sulfide and sulfate (Bak and Cypionka, 1987; Jørgensen et al, 2019). In lake sediments, chemolithotrophs have been found to utilise ferrous sulfide (FeS) and hydrogen sulfide to generate pyrite (FeS_2) and hydrogen (Rickard and Luther, 2007; Zopfi et al, 2008; Thiel et al, 2019). The hydrogen generated from this reaction can be potentially used as reductant for carbon dioxide conversion to methane or reduction of organic matter (Holmkvist et al, 2011; Thiel et al, 2019).

The overall concentration of sulfur in Ace Lake is much lower than what is observed in sea water of similar chlorinity (Burton and Barker, 1979). This depletion of sulfur in

Ace Lake, specifically in the anoxic zone, has been speculated to be the result of sulfate reduction by the SRB and the subsequent loss of most of the hydrogen sulfide to the atmosphere sometime in the past during a holomixis event (Burton and Barker, 1979; Cromer et al, 2005). A syntrophic relationship between the GSB (*Chlorobium*) and SRB in the Ace lake oxycline is a major component of sulfur cycling in the lake. *Chlorobium* oxidise hydrogen sulfide into sulfate during anoxygenic photosynthesis, but cannot utilise the sulfate they produce due to lack of genes associated with assimilatory sulfate reduction (Coolen et al, 2006; Ng et al, 2010; Lauro et al, 2011). On the other hand, the SRB require sulfate for anaerobic respiration and in the process convert sulfate back to hydrogen sulfide, which can be used by *Chlorobium* (Coolen et al, 2006; Ng et al, 2010; Lauro et al, 2011). This is consistent with the concentrated levels of sulfate in the oxic zone through to the anoxic zone just below oxycline as well as the concentrated levels of hydrogen sulfide in the anoxic zone of Ace Lake (Burton and Barker, 1979; Hand and Burton, 1981; Franzmann et al, 1991b; Rankin et al, 1999). The SRB are also present in the anoxic zone, close to the oxycline where the sulfate concentration is not completely depleted (Hand and Burton, 1981; Mancuso et al, 1990; Bell and Laybourn-Parry, 1999; Lauro et al, 2011). The hydrogen sulfide in the Ace Lake monimolimnion helps maintain the reduced environment in this zone (Rankin et al, 1999).

1.2.3.3 Nitrogen

Microbes play an important role in the cycling of nitrogen, the most abundant (80%) molecule in Earth's atmosphere that plants and animals cannot utilise in its gaseous form. Generally, nitrogen cycling involves nitrogen gas and its reduced (ammonia) and oxidised (nitrite, nitrate, etc) forms. Some microbes such as some cyanobacteria can reduce atmospheric nitrogen to its bioavailable form (ammonia) through nitrogen fixation (Bernhard, 2010). Other microbes mineralise organic nitrogen (like amino acids) to inorganic ammonia for energy production via ammonification (Strock, 2008; Bernhard, 2010). Ammonia, in turn, can be oxidised to nitrites and nitrates via nitrification (Bernhard, 2010). Some microbes such as chemoautotrophic bacteria and archaea can utilise the energy produced from nitrification for carbon fixation (Grzymski et al, 2012; Williams et al, 2012). Oxidised nitrogen such as nitrates, nitrites, nitric oxide and nitrous oxide can be converted to nitrogen via the denitrification process (Bernhard, 2010). However, some chemoheterotrophs have the capacity to reduce nitrates and nitrites directly to ammonia (nitrate/nitrite ammonification) as part of

anaerobic nitrate respiration (Kraft et al, 2011; Lam and Kuypers, 2011). Nitrite and ammonia can also be converted to nitrogen via an anaerobic synproportionation reaction termed as anammox, which has been identified as the major process for transforming bioavailable fixed nitrogen to its inert gaseous form (Devol, 2003; Francis et al, 2007; Bernhard, 2010). Anammox bacteria (members of *Planctomycetes* phylum) perform this reaction in special lipid bilayer membrane chambers called anammoxosome in their cytoplasm (Strous et al, 1999; Boumann et al, 2009; Jetten et al, 2009).

In Ace Lake, dissolved nitrogen gas is concentrated in the oxic zone through to a few metres into the anoxic zone, but is absent in the lower depths (>18 m) (Burton, 1980). A few microbes with the potential for nitrogen fixation have been identified in Ace Lake. Of these, the *Chlorobium* in the Ace Lake oxycline is suspected to drive nitrogen fixation in the lake (Lauro et al, 2011). Although nitrogenase proteins were not identified in the Ace Lake metaproteome from the oxycline, their absence might have resulted from the inhibition of nitrogenase genes by the ammonia present in the zone at the time (Ng et al, 2010; Lauro et al, 2011). Other than *Chlorobium*, the cyanobacteria in the algal mats of Ace Lake are capable of low levels of nitrogen fixation (Rankin et al, 1999; Lauro et al, 2011). However, *Synechococcus*, the most abundant cyanobacteria in Ace Lake mixolimnion waters, cannot fix nitrogen (Powell et al, 2005). In the monimolimnion, some of the methanogenic archaea, e.g., *M. burtonii*, contain nitrogenase genes and have the potential for fixing nitrogen (Allen et al, 2009). Other than nitrogen fixation, the potential for low levels of denitrification has been identified in Ace lake, but nitrification probably does not occur (Lauro et al, 2011). This is consistent with the absence of oxidised nitrogen (nitrates and nitrites) from the anoxic zone of Ace Lake where ammonia is concentrated, and its low concentration in the oxic zone of the lake compared to marine waters (Burton, 1980; Hand and Burton, 1981; Burch, 1988; Perriss et al, 1995; Gibson et al, 1997; Rankin et al, 1999). The oxygenated nitrogen concentration in the Ace Lake oxic zone increases in winter (Rankin et al, 1999). *Chlorobium* as well as members of *Actinobacteria* and SAR11 clade are potentially involved in nitrogen absorption and assimilation (Ng et al, 2010; Lauro et al, 2011). Moreover, members of *Planctomycetes* have been predicted to perform anaerobic ammonia oxidation (anammox) in the anoxic zone of Ace Lake (Lauro et al, 2011), where ammonia is concentrated around 15 m depth (Burton, 1980; Hand and Burton, 1981). The concentration of ammonia fluctuates seasonally, being

higher at the end of summer and in early winter (Bell, 1998; Bell and Laybourn-Parry, 1999; Laybourn-Parry et al, 2002; Madan et al, 2005; Laybourn-Parry et al, 2007). Considering the low concentration of oxidised nitrogen in Ace Lake as well as the low potential for denitrification and absence of nitrification in the lake, the overall lake community appears to depend on nitrogen fixation by the *Chlorobium* (Lauro et al, 2011).

1.2.3.4 Other macronutrients and trace metals

Phosphorus is essential to life on Earth, being a major component of genetic material (DNA, RNA), energy storage components (ATP) and biological membranes (phospholipids). In aquatic systems, it is generally found in the form of dissolved or particulate and organic or inorganic phosphorus (Wetzel, 2001). Organic phosphate is mainly present in the oxic zone of Ace Lake (Hand and Burton, 1981; Burch, 1988). On the contrary, soluble reactive phosphorus (an important algal nutrient) is concentrated in depths below the halocline of Ace Lake, reaching maximum concentration in the lower depths of the anoxic zone (Hand and Burton, 1981; Burton, 1980; Burch, 1988; Rankin, 1998; Bell and Laybourn-Parry, 1999). The concentration of silicate (an important nutrient for diatoms) in Ace Lake increases with depth in the oxic zone (Hand and Burton, 1981; Rankin et al, 1999).

Microbes generally require small amounts of trace metals to perform various metabolic functions. For example, iron is an essential component of GSB photosynthetic mechanism, iron and molybdenum are found in nitrogenase involved in nitrogen fixation, magnesium is a component of photosynthetic components like bacteriochlorophyll, and cobalt is found in the corrin rings of cofactors like adenosylcobalamin. The concentration of various trace metals has been previously measured in Ace Lake (Masuda et al, 1988). Most trace metals, such as potassium, calcium, magnesium, iron, strontium, chromium, cobalt, and antimony, have an increasing concentration gradient from oxic to anoxic waters of Ace Lake. Others like copper and nickel are present only in the oxic or anoxic zone, respectively. The concentration of aluminium, manganese, and selenium was reported to be highest at 18 m depth in Ace Lake, but zinc concentration was lowest at this depth (Masuda et al, 1988). The concentration of most of these trace metals in Ace Lake is much higher than their concentration in sea water, which might be due to inflow of aerosols and weathering of rocks surrounding the lake since its isolation (Masuda et al, 1988).

1.3 *Chlorobium* species and their geographic distribution

The genus *Chlorobium* belongs to the *Chlorobiaceae* family of GSB, which were first recognised as a distinct group of phototrophic bacteria by Pfennig and Trüper (1971). They are anaerobic photoautotrophs that fix carbon dioxide via reverse tricarboxylic acid cycle (rTCA) and perform anoxygenic photosynthesis using sulfide or other sulfur compounds as electron donors, eventually oxidising the sulfur compounds to sulfate (Sakurai et al, 2010; Tang and Blankenship, 2010). *Chlorobium* spp. also use a bacteriochlorophyll a-containing type I reaction centre placed in their chlorosomes (very sensitive light-harvesting antennae) for gathering light from low light environments (Eisen et al, 2002; Blankenship and Matsuura, 2003). As bacteriochlorophyll A (*fmoA*) gene is specific to GSB, it can be used for the phylogenetic analysis of the members of *Chlorobiaceae* family (Alexander et al, 2002; Alexander and Imhoff, 2006). The major pigments and carotenoids in the photosynthetic apparatus of GSB include bacteriochlorophyll c, d, or e and chlorobactene, isorenieratene, or γ-carotene, respectively (Schmidt, 1978; Gibson et al, 1984; Imhoff, 2014). GSB can be green- or brown-coloured depending on the carotenoids and pigments they contain — green-coloured GSB have chlorobactene and bacteriochlorophyll c or d, whereas brown-coloured GSB contain isorenieratene and bacteriochlorophyll e (Imhoff, 2014). The brown-coloured GSB are more sensitive to light and can outperform the green-coloured GSB under very low light conditions. Most GSB depend on vitamin B12 for growth and its deficiency can severely affect their bacteriochlorophyll content, precluding chlorosome formation (Sato et al, 1981; Fuhrmann et al, 1993).

As *Chlorobium* are obligate anaerobes and contain a photosynthetic apparatus (chlorosomes) that is very sensitive to low light, they grow in reduced environments (Van Gemerden and Mas, 1995). Overall, *Chlorobium* have been isolated from anoxic aquatic habitats from across the globe irrespective of the environmental temperature — from hydrothermal vents as well as lakes in the temperate and tropical zones and polar lakes (Table 1.3). Instead, availability of light and reduced sulfur compounds appears to be important requirements for the growth of these GSB. For example, a GSB isolated from a deep-sea hydrothermal vent was speculated to survive in the dark ocean depths using the geothermal radiation and effluents from the black smoker as sources of light

and reduced sulfur, respectively, for anoxygenic photosynthesis (Beatty et al, 2005). This notion was supported by the absence of the organism from the surrounding oxic ocean waters. In general, the *Chlorobium* from stratified lakes have been identified and/or isolated from the oxic-anoxic interface, where the available light is low and the waters are rich in reduced sulfur (Table 1.3).

Table 1.3 Global distribution of some *Chlorobium* species. [A] The background colour indicates the overall temperature conditions of the habitat — habitats such as hot springs and hydrothermal vents are shown in red; warm tropical habitats are shown in yellow (<30° N/S latitude); habitats in temperate zone (between 30 and 50° N/S latitudes) are shown in light blue; subpolar habitats (between 50 and 60° N/S latitude) are shown in blue; and polar habitats (>60° N/S) are shown in dark blue. [B] The last column indicates the methods applied for the identification of the *Chlorobium* species and the publications in which they were reported. All aquatic habitats, except hot springs and hydrothermal vents, described here are stratified systems. The *Chlorobium* species were identified in the oxycline of these systems. The data are arranged from top to bottom in the direction of north to south latitude.

Habitat[A]	*Chlorobium* species	Methods and references[B]
Lake A in Ellesmere Island, High Arctic Canada	*Chlorobium* sp.	Metagenomic analysis; read taxonomic classification (Comeau et al, 2012)
Lake Bolshye Khruslomeny in Oleniy Island, White Sea	*Chlorobium phaeovibrioides*; two strains — one green-coloured (*Gr*Khr17) and one brown-coloured (*Br*Khr17)	Cell culture, isolation, and characterization; *16S rRNA* gene-based phylogenetic analysis (Grouzdev et al, 2019)
Hot spring microbial mats in Greenland	GSB that clustered with uncultured *Chlorobium sp.*	PCR-DGGE analysis; *16S rRNA* gene-based phylogenetic analysis (Roeselers et al, 2007)
Lake Polden in Norway.	*Chlorobium luteolum*	Cell culture, isolation, characterization, and sequencing (https://genome.jgi.doe.gov/portal/pellu/pellu.home.ht

		ml; part of a project led by Donald A Bryant)
Sediment samples from freshwater creeks and ditches near Konstanz, Germany	*Chlorobium ferrooxidans*	Cell culture, isolation, and characterization; *16S rRNA* gene-based phylogenetic analysis (Heising et al, 1999)
Black Sea meromictic basin	*Chlorobium phaeobacteroides*, *C. phaeovibrioides*, *Chlorobium* sp.	Pigment chromatography (Repeta et al, 1989); Cell culture, isolation, and characterization; pigment chromatography (Overmann et al, 1992)
Lake Faro, Italy	*C. phaeobacteroides*	Radioisotopic analysis (Sorokin and Donato, 1975); Van Gemerden and Mas, 1995
Fayetteville Green Lake, New York	*C. phaeobacteroides*	Cell culture and isolation (Culver and Brunskill, 1969)
Lake Banyoles, Lake Vilar, Lake Cisó, Lake Nou, Lake Coromines, Lake Negre, Lake Estanya, and Lake Moncortes in Spain	*C. phaeobacteroides*, *Chlorobium limicola*	Cell culture, isolation, and characterization; pigment chromatography (Montesinos et al, 1983)
Lake Banyoles, Spain	*C. luteolum*	Metagenomic analysis; *16S rRNA* gene-based phylogenetic analysis (Llorens–Marès et al, 2017)
Cullera estuary, Spain	*C. phaeovibrioides*	Radioisotopic analysis; pigment chromatography (Miracle and Vicente, 1985); Van Gemerden and Mas, 1995

Cross Reservoir, Kansas	*C. limicola*, *Chlorochromatium aggregatum*	Fluorescence spectrophotometry; pigment chromatography; microscopy (Chapin et al, 2004)
Plumes of a deep-sea hydrothermal vent in East Pacific Rise	GSB that clustered with *Chlorobium* sp. and *Prosthecochloris* sp.	Cell culture, isolation, and characterization; pigment chromatography; microscopy; bacteriochlorophyll A and *16S rRNA* gene-based phylogenetic analysis (Beatty et al, 2005)
Bietri Bay, a stratified lagoon in western Africa	Two *Chlorobium* spp. that clustered with *Chlorobium vibrioforme* and *C. phaeobacteroides*	Cell culture, isolation, and characterization (Caumette, 1984)
Lake Fidler, Tasmania	*C. limicola*	Microscopy (Baker et al, 1985)
Hot spring microbial mats in New Zealand	*Chlorobium tepidum*	Cell culture and isolation (Castenholz et al, 1990); Cell culture, isolation, and characterization (Wahlund et al, 1991)
Sediment sample from Borge Bay in Signy Island, Antarctica	*C. limicola, C. vibrioforme*	Cell culture, isolation, and morphology characterization (Herbert and Tanner, 1977)
Ace Lake in the Vestfold Hills, Antarctica	*Chlorobium* sp. that clustered with *C. phaeovibrioides*	Microscopy; enrichment culture and morphology characterization (Burke and Burton, 1988a); *16S rRNA* gene-based phylogenetic analysis; lipid chromatography (Coolen et al, 2006);

		Metaproteogenomic analysis; *16S rRNA* gene-based phylogenetic analysis (Ng et al, 2010); Metaproteogenomic analysis; *16S rRNA* gene-based phylogenetic analysis; read taxonomic classification (Lauro et al, 2011)
Ellis Fjord, Taynaya Bay, Ace Lake, Burton Lake, Clear Lake, McCallum Lake, Abraxas Lake, Pendant Lake, and Fletcher Lake in the Vestfold Hills, Antarctica	*C. vibrioforme*, *C. limicola*	Microscopy; enrichment culture and morphology characterization (Burke and Burton, 1988a; Burke and Burton, 1988b)

In most studies listed in Table 1.3, the *16S rRNA* (ribosomal RNA) marker gene has been used for phylogeny assessment (Heising et al, 1999; Beatty et al, 2005; Coolen et al, 2006; Roeselers et al, 2007; Ng et al, 2010; Lauro et al, 2011; Llorens–Marès et al, 2017; Grouzdev et al, 2019). Some studies have also used *fmoA* gene for taxonomic classification (Beatty et al, 2005), and chromatography to differentiate between pigments from phototrophic bacteria and eukarya and/or to identify the *Chlorobiaceae* members (Montesinos et al, 1983; Miracle and Vicente, 1985; Repeta et al, 1989; Overmann et al, 1992; Chapin et al, 2004; Beatty et al, 2005; Coolen et al, 2006). A few metagenomic studies have also identified *Chlorobium* in habitats from different global locations — Lake A (Arctic; Comeau et al, 2012), Lake Banyoles (Spain; Llorens–Marès et al, 2017), Ace Lake (Antarctica; Ng et al, 2010; Lauro et al, 2011). Culture-based studies have identified *C. limicola* in multiple Antarctic lakes (Burke and Burton, 1988a; Burke and Burton, 1988b), in a Subantarctic bay (Herbert and Tanner, 1977), and a lake in Tasmania (Baker et al, 1985) in the Southern Hemisphere as well as in multiple lakes in Spain (Montesinos et al, 1983) in the Northern Hemisphere. Moreover, *C. phaeovibrioides* was identified in an estuary in Spain (Miracle and Vicente, 1985) as well as a Subarctic lake (Grouzdev et al, 2019), and a *Chlorobium* closely related to *C. phaeovibrioides* has also been identified in an Antarctic lake (Burke and Burton, 1988a; Coolen et al, 2006; Ng et al, 2010; Lauro et al, 2011). Considering the locations of their

habitats, the distribution of different species of *Chlorobium* does not appear to be restricted to specific geographic localities, except probably the thermophile *Chlorobium tepidum*.

1.4 Metagenomics

The advances in sequencing technology, such as high-throughput sequencing (HTS), have made it possible to efficiently sequence large quantities of DNA in a cost-effective manner. HTS is performed by next-generation sequencing technologies (also called second-generation sequencing technologies) and refers to massively parallel sequencing of DNA, which allows for sequencing of whole genomes within a time-frame of days. To put this in perspective, the sequencing of human genome (~3 billion bp long) using the first-generation sequencing technologies (also referred to as Sanger sequencing) took more than a decade and cost billions of dollars (Grada and Weinbrecht, 2013). The cost of sequencing a human genome using HTS methods would be less than a thousand dollars now (https://www.genome.gov/about-genomics/fact-sheets/). HTS can be used for direct sequencing of environmental DNA samples without the need for culturing, although it can be used for sequencing DNA from cultures as well. Some of the well-known HTS platforms include Illumina, 454 pyrosequencing, ABI SOLiD, Ion torrent, and Nanopore technologies.

Unlike genomic DNA sequences from a single isolate, metagenomes represent a snapshot of a microbiome. Metagenomics can be used: (i) to understand the overall community structure and functional potential of an environment; (ii) to compare the community structure and functional potential of different environments through comparative metagenomics; and (iii) to analyse the shift in community structure and functional potential over a period of time by utilising a time-series of metagenomes from the environment. Metagenomes have been found to be especially useful in studying environments that contain microbes that have not been cultured or cannot be easily cultured, such as the microbial communities in Antarctic lakes. Metagenomics-led studies of Antarctic lakes in the Vestfold Hills have provided insights into the lake communities and led to important discoveries, such as the inter-genera gene exchange among the haloarchaea in the hypersaline Deep Lake (DeMaere et al, 2013) and the presence of virophages in Organic Lake (Yau et al, 2011). Metagenomic studies of Ace

Lake and Organic Lake, two meromictic Antarctic lakes, have shown the niche adaptation of various microbial communities and their potential contributions to nutrient cycling in the lakes (Lauro et al, 2011; Yau et al, 2013).

Metagenome analysis can be challenging, and using metagenomes to understand microbial communities can have some limitations. Metagenomes are usually large datasets, making data handling and computational analysis a little difficult (Wooley and Ye, 2009). Efficient computational tools and approaches are required to analyse metagenomes, especially considering the high species diversity usually captured in them (Wooley and Ye, 2009). Metagenome sequence assemblers allow for generation of long contigs, which can be used to produce draft genome assemblies. However, a large portion of metagenomes, especially small contigs and unassembled reads, cannot be binned into genome assemblies, and their functional potential analysis can be tricky owing to their short lengths (Prakash and Taylor, 2012). Another limitation of using metagenomes is based on the reference databases available for analyses. Depending on the origin of the metagenome, a large number of the metagenome sequences can be unclassified because of unavailability of closely related reference organisms in the databases (Prakash and Taylor, 2012; Teeling and Glöckner, 2012). Such unclassified sequences can indicate data from novel organisms or proteins. This has been observed for environmental metagenomes such as those from oceans and soil, and it has been speculated that the bias in the number of database sequences from human-associated sources might contribute to it (Frias-Lopez et al, 2008; Prakash and Taylor, 2012). Another challenge of analysing metagenomes is that there is no single tool for the analysis of metagenomic data, therefore, results from various computational tools need to be generated separately and then combined and interpreted in a comprehensive manner.

1.4.1 Methods for metagenomic data analysis

As metagenome sequencing has become relatively easy and routine, various methods and software capable of handling large datasets have been developed for the analysis of metagenomes, so that meaningful data can be generated (Figure 1.4).

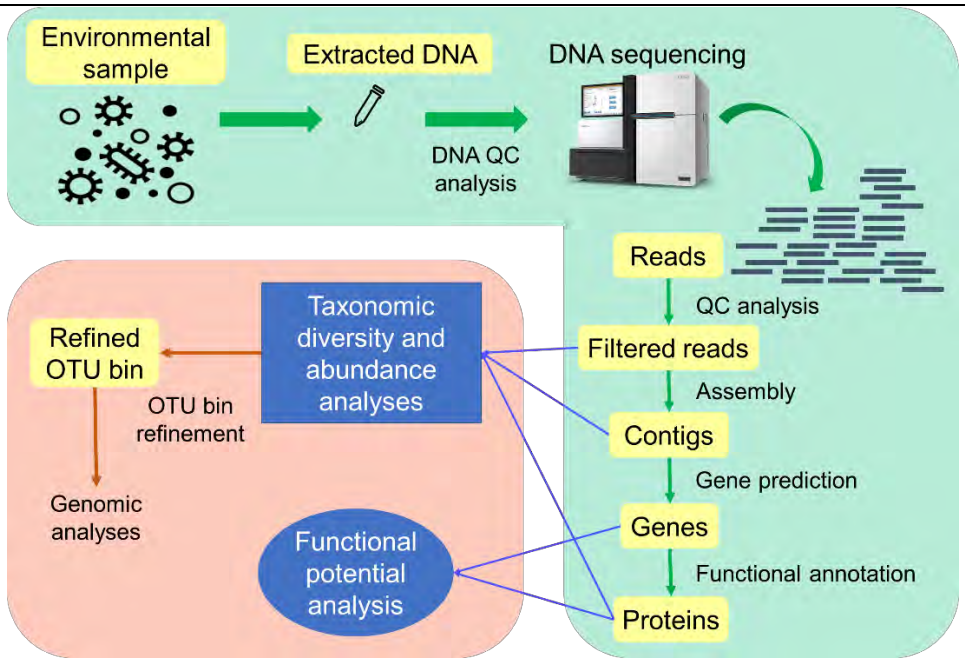


Figure 1.4 Schematic showing metagenome preparation and analysis. Metagenomes are prepared from environmental samples and can be used for the analysis of biodiversity and functional potential of the environment. The steps covered by the green background show metagenome sequencing and initial data preparation. The steps covered by the orange background show metagenomic data analyses. The sequencer image shown in the schematic was adapted from https://www.illumina.com/systems/sequencing-platforms/hiseq-x.html. The icons used in the figure were taken from The Noun Project website (https://thenounproject.com/).

The taxonomic classification of metagenomic data is a two-step process: (i) alignment of query read, contig, or protein sequences to the sequences in the reference databases and (ii) mapping of the alignment output to the taxonomy provided in the reference databases. Various software capable of performing either one or both steps are available. Some of the well-known alignment software such as BLAST (Altschul et al, 1990), LAST (Kielbasa et al, 2011), Bowtie 2 (Langmead and Salzberg, 2012), and DIAMOND (Buchfink et al, 2014) have been used in conjunction with mapping software like MEGAN6 (Huson et al, 2016), MetaPhlAn2 (Segata et al, 2012), and PhyloSift (Darling et al, 2014) (described below in sections 1.4.1.1 and 1.4.1.2). Other software such as Kaiju (Menzel et al, 2016) perform sequence alignment as well as taxonomy mapping. The sequence clusters generated by taxonomic binning of metagenome sequences are referred to as operational taxonomic units (OTUs), as they are produced based on sequence homology of query to closely related reference sequences. OTUs usually represent taxonomic levels like genus and species to which

metagenome sequences might belong. For species verification of bacterial and archaeal OTUs, it has been suggested that ≥99% 16S rRNA gene identity and >95% average nucleotide identity (ANI) to a reference species might be sufficient to establish their close relatedness (Chan et al, 2012; Kim et al, 2014). ANI of 95–96% is considered equivalent to the 70% cut-off of DNA-DNA hybridisation, previously used to distinguish between prokaryotic species (Brenner, 1973; Stackebrandt and Goebel, 1994; Goris et al, 2007; Richter & Rosselló-Móra, 2009; Chan et al, 2012; Kim et al, 2014).

The OTU abundance data generated using taxonomic classification software can be used for various types of statistical analyses. PRIMER v7 (Primer-e, NZ) is an efficient statistical tool for the analysis of multivariate data and is capable of handling large datasets. It was specifically designed for the analysis of environmental data including OTU abundances as well as biomass measures. PRIMER v7 has multiple subroutines that allow for a variety of analyses: (i) calculating similarity matrices; (ii) sample and variable clustering analyses; (iii) multi-dimensional scaling analyses (MDS, nMDS); (iv) analysis of similarity (ANOSIM) tests; (v) analysis of environmental factors; (vi) assessing relationship between environmental factors and abundance/biomass data; (vii) calculating biodiversity measures; (viii) plotting diversity curves; and more. Its interactive user interface makes PRIMER v7 a relatively easy to use software that does not require a high-level of technical expertise to start with. Other than this software, R language tools are often used for statistical analyses of environmental data, but that requires the user to know/learn R computing language.

The OTU bins, generated from binning of metagenomic sequences based on their taxonomic classification, can be filtered using RefineM (Parks et al, 2017). This software removes contamination from a bin by sifting out the outlier sequences based on their taxonomic classification, GC content, coverage, and/or tetra nucleotide frequency (TNF). For taxonomic classification of the sequences, RefineM uses the Genome Taxonomy Database (GTDB) as a reference database. RefineM and GTDB are often used with CheckM (Parks et al, 2015) to assess and improve the completeness of a bin. Various software are available for the genomic analyses of refined OTUs depending on the type of analysis to be performed. For example, JSpeciesWS (Richter et al, 2016), pyani (Pritchard et al, 2016), and fastANI (Jain et al, 2018b) can be used for calculating the ANI of an OTU against reference genomes, the genomic variation in

an OTU bin can be assessed using Artemis (Rutherford et al, 2000), Integrated Genome Browser (Nicol et al, 2009), Integrative Genomics Viewer (IGV; Robinson et al, 2011), or Mauve (Darling et al, 2004), and MEGA (Kumar et al, 2018) can be used for phylogenetic analysis.

1.4.1.1 Sequence alignment

BLAST

BLAST is the most commonly used alignment tool that allows for nucleotide-nucleotide sequence matching using blastn, megablast, and tblastx modules, protein-protein matching using blastp module, translated nucleotide-protein matching using blastx module, and protein-translated nucleotide matching using tblastn module. Sequence alignment with BLAST+ (an updated version of BLAST; Camacho et al, 2009) has three distinct phases: (i) reading the query sequence, applying any input filters, and preparing an index table using the 'perfect hashing' function; (ii) matching the reference sequences against the query sequences in search of hits; (iii) reassessing alignments in search of indels and mismatches and calculating alignment statistics, such as bit score and e-value. BLAST+ is capable of aligning very long query and reference sequences faster than the BLAST search tool. This is accomplished by splitting the query sequence into smaller overlapping sequences in the first phase of BLAST+ alignment and then merging the sequences and their alignments in the last phase (Camacho et al, 2009). Splitting the query sequence also reduces the amount of processor cache memory used for alignment, making it more efficient (Camacho et al, 2009).

LAST

LAST alignment tool can be used for the alignment of nucleotide as well as protein query sequences to a reference database. This tool uses the standard 'seed-and-extend' algorithm for alignment, but uses 'adaptive seeds' as opposed to 'fixed-length seeds' used by BLAST and other alignment tools (Kielbasa et al, 2011). Here, seeds refer to short stretches of query sequences (by default starts with 1 bp in LAST) that are picked for initial alignment to reference sequences. The 'adaptive seeds' approach allows variable length sequence matches between the query and reference sequences as long as the matching sequence occurs no more than a pre-defined number of times (by default 10) in the reference sequence. This method greatly improves the speed of LAST for the alignment of long sequences (by 10 to 100-fold) when compared with 'fixed-length-

seeds' approach, without compensating the alignment sensitivity (Kielbasa et al, 2011). Other versions of 'adaptive seeds', such as 'adaptive spaced seeds' (where some positions on the seed are allowed to have any mismatches) and 'adaptive subset seeds' (where some positions on the seed are allowed to have certain mismatches), also improve alignment speed of large sequences (Kielbasa et al, 2011).

Bowtie 2

Bowtie 2 is an alignment tool that allows for fast, sensitive, and accurate gapped alignment of reads to reference sequences (Langmead and Salzberg, 2012). As an initial step, it uses the 'full-text minute index' approach to index the reference sequences. The main alignment phase is divided into four steps: (i) selecting a seed sequence (by default starts with 20–25 bp) from the query sequence; (ii) aligning the seed to the indexed reference sequences in search of an ungapped alignment; (iii) prioritizing seed alignments and calculating their position on the reference sequence; and (iv) extending the seed alignments to full alignments using dynamic programming. When compared with other alignment tools like BWA, Bowtie, and SOAP2, the alignment speed of Bowtie 2 is better for both unpaired and paired-end reads of various lengths (100–400 bp) (Langmead and Salzberg, 2012). The accuracy of Bowtie 2 is also better than the other tools in case of unpaired reads, but is on par with BWA for paired-end reads (Langmead and Salzberg, 2012).

DIAMOND

DIAMOND alignment tool was specifically developed for the fast and sensitive alignment of reads to reference protein sequences, and is also capable of aligning protein query sequences (Buchfink et al, 2014). Like BLAST and LAST alignment tools, DIAMOND uses the 'seed-and-extend' algorithm for alignment, but uses 'spaced seeds' in place of 'adaptive seeds' (used by LAST) or 'fixed-length seeds' (e.g., used by BLAST) (Buchfink et al, 2014). 'Spaced seeds' are small stretches of query sequences where some positions are considered to be of low importance and are allowed to have mismatches, which can lead to faster alignment. The number of such positions and their layout in a seed sequence is referred to as the weight and shape of a 'spaced seed'. For improved sensitivity, DIAMOND uses specific combinations of seed weight and shape (sensitive: 12×15–24; most sensitive: 9×16) (Buchfink et al, 2014). Moreover, it uses a 'double indexing' approach in which the spaced seeds and their locations on query as

well as reference sequences are simultaneously parsed and arranged in dictionary order. A comparative analysis of DIAMOND and BLASTX using permafrost metagenomic reads shows that DIAMOND is more than four-orders of magnitude faster than BLASTX, and the two methods have similar sensitivity (Buchfink et al, 2014).

Overall, among these alignment software used with mapping software, LAST is often used for alignment of long sequences, due to its improved computing speed and ability to handle frame-shifts (Darling et al, 2014; Huson et al, 2018; Bağci et al, 2019). For large datasets of short sequences, such as metagenomic reads and proteins, DIAMOND is suggested due to its fast alignment speed, especially with mapping tools like MEGAN (Bağci et al, 2019).

1.4.1.2 Taxonomic classification

MEGAN6

MEGAN software was developed for the taxonomic classification of metagenomic reads and can be used for classification of protein sequences. Prior to using MEGAN, the metagenomic reads need to be aligned against a reference database; MEGAN can work with the aligned reads from BLAST as well as DIAMOND outputs (Huson et al, 2007). MEGAN software uses the 'lowest common ancestor' (naïve LCA) algorithm to assign taxonomy to the aligned reads and creates a phylogenetic tree from the output (Huson et al, 2007). This algorithm allows sequences specific to a species to be assigned to the species node in the phylogenetic tree, but the more conserved sequences, such as genus- or family-specific sequences, are assigned to higher taxa levels. A more recent version of MEGAN, namely MEGAN6, provides additional modules for functional potential analyses, such as COG (clusters of orthologous groups) analysis using eggNOG database, KEGG analysis using KEGG Orthology (KO) database (only available in the paid-version of MEGAN), and GO (gene ontology) analysis using InterPro database (Huson et al, 2016). These functional potential analyses of the query sequences generate: (i) COG annotations and categorisations as well as the abundances of functional groups; (ii) KO annotations and the abundances of metabolic pathways; and (iii) GO annotations and the abundances of protein families. An additional functionality recently added to the MEGAN graphical user interface (GUI) is the 'gene-centric assembly' module, which allows the user to request the assembly of all reads assigned to any taxonomic or functional node in MEGAN (Huson et al, 2017). The data

generated by MEGAN can be visualised in the MEGAN GUI and can be used to generate various plots, charts, clusters, and networks.

PhyloSift

PhyloSift software is a phylogenetic analysis pipeline for both genome and metagenome analyses using protein or nucleotide query sequences (Darling et al, 2014). Unlike most taxonomic classification tools that use reference databases of all protein or nucleotide sequences, PhyloSift uses a database of marker genes for taxonomic assignments. The PhyloSift marker gene sets include core markers, such as small subunit (SSU) rRNA genes, mitochondrial genes, and plastid genes, and extended markers, which are an extremely large dataset of clade-specific gene sequences. These marker gene sets are automatically updated on a regular basis to include markers from newly assembled genomes. PhyloSift uses a combination of LAST and hmmalign (http://hmmer.org/) for alignment of query DNA sequences of length <600 bp and query protein sequences and uses LAST and Infernal (Nawrocki and Eddy, 2013) for aligning query DNA sequences >600 bp to the reference marker genes. The taxonomic classification of the aligned queries is performed using pplacer (Matsen et al, 2010). An analysis of the computational resources needed to run PhyloSift shows that it requires roughly 5 h to analyse 10^6 reads on a single processor, using around 8 GB memory (Darling et al, 2014).

MetaPhlAn2

MetaPhlAn2 was originally created for the phylogenetic analysis of metagenome reads (Segata et al, 2012). It uses Bowtie 2 to align the reads to its database of clade-specific markers, which were prepared from taxa-specific genes coding for various functions. The relative abundances of the taxa identified in the metagenomes are calculated based on the number of reads assigned to the taxa and the length of the markers. A comparative analysis of MetaPhlAn and other phylogenetic analysis tools such as PhymmBL, Phymm, RITA, and NBC shows that MetaPhlAn is faster and more accurate in classifying reads (Segata et al, 2012). However, a drawback of using this software is that only well-characterised environments can be accurately analysed (Darling et al, 2014). All reads belonging to the taxa whose marker genes are not present in the MetaPhlAn2 database are grouped as ‘unclassified’ (Segata et al, 2012). Although MetaPhlAn2 provides the flexibility to add customised clade-specific markers to its

database, the process of adding the markers is exhaustive and requires technical expertise.

Kaiju

Kaiju is capable of fast and sensitive analysis of metagenomic read taxonomy and does not rely on external alignment software for matching metagenomic reads to a reference database. Like Bowtie 2, it uses the 'full-text minute index' approach to index the reference protein database (Menzel et al, 2016). The metagenomic reads are translated into six reading frames and are fragmented at their stop codons. Kaiju uses two algorithms to align the query fragments to the indexed database using a k-mer based approach: (i) 'maximum-exact-match' (MEM) algorithm which allows only exact matches — query fragments are first sorted by their lengths and then aligned against the indexed database until the longest exact match is obtained and (ii) 'Greedy' algorithm which allows substitution and thereby sequence extension — query fragments are arranged based on their BLOSUM62 score and then aligned against the indexed database until the best scoring match is obtained (Menzel et al, 2016). K-mers refer to all possible sub-sequences of length 'k' in a sequence, e.g., the sequence TCG has two possible 2-mers (TC and CG) and one 3-mer (TCG). Relative abundances are calculated as the number of reads assigned to a taxon relative to the total number of reads in the metagenome. A comparative analysis of Kaiju-MEM and Kaiju-Greedy as well as other k-mer based methods such Kraken and CLARK shows that Kaiju-MEM is fastest at computing the taxonomies of all types of read sequences tested — Illumina unpaired and paired-end reads (100 bp and 250 bp) and 454 unpaired reads (350 bp) (Menzel et al, 2016). However, Kaiju-Greedy has the highest sensitivity and accuracy of taxonomic assignments among the tested software. Moreover, a comparative analysis of Kaiju-MEM, Kaiju-Greedy, and Kraken using metagenomes from various environments, including human-associated, freshwater, seawater, soil, bioreactor samples, shows that Kaiju-Greedy is able to classify most reads in all metagenomes (24–73%) as compared to Kaiju-MEM (19–65%) and Kraken (3–46%) (Menzel et al, 2016).

MEGAN-LR

MEGAN-LR is the latest version of MEGAN and was specially designed for taxonomic classification of long reads and contigs using the 'interval-union LCA' algorithm (Huson et al, 2018). This algorithm is a variation of the naïve LCA algorithm and is

divided into a number of steps: (i) identifying intervals, where intervals refer to pieces of query sequences to which the reference proteins have alignments; (ii) identifying significant alignments to an interval — an alignment is considered significant if its bit score is within 10% (default value) of the best bit score of an alignment covering the same interval; (iii) union of intervals — for a query sequence, the various significant interval alignments of reference proteins from different taxa are put together and if two interval alignments overlap they are merged into one; and (iv) LCA assignment — for each query, the interval sets from different taxa are compared and the query is assigned to the taxon that covers ≥80% of the aligned query sequence, prioritizing the lowest-level taxa (e.g., species- and strain-level taxa) (Huson et al, 2018). The developers of MEGAN-LR suggest the use of LAST alignment tool for sequence alignment prior to taxonomic mapping, as LAST can handle frame-shifts and can align long sequences with high speed and sensitivity (section 1.4.1.1). A comparative analysis of the LAST/MEGAN-LR approach and Kaiju shows that the sensitivity and accuracy of LAST/MEGAN-LR is much better than that of Kaiju (Huson et al, 2018). However, Kaiju is much faster at computing the taxonomies.

1.5 Objectives

This is the first metagenomics-led seasonal study of Ace Lake in the Vestfold Hills, East Antarctica, using time-series samples spanning nearly a decade (Dec 2006–Jan 2015) and including samples from austral summer (Dec 2006, Feb 2014, Dec 2014, Jan 2015), winter (Jul and Aug 2014), and spring (Nov 2008, Nov 2013, Oct 2014). The overall aim of this thesis was to analyse the time-series of metagenomes from Ace Lake, to assess the impact of change in season on the microbial community structure and functional potential of the lake. Various software and computational methods were tested to improve upon a preliminary in-house metagenome analysis pipeline referred to as Cavlab pipeline (Chapter 2). The upgraded Cavlab pipeline, along with other computational methods required for specific analyses, were used for a comprehensive study of Ace Lake metagenomes, including analysis of biodiversity, functional potential, viruses and their potential hosts, and two key bacteria (*Chlorobium* and *Synechococcus*) (Chapters 3, 4, and 5). Furthermore, with the availability of metagenomes from three stratified marine systems, namely Ace Lake, Ellis Fjord, and

Taynaya Bay, in the Vestfold Hills, the endemicity of one key microbe (*Chlorobium*) from Ace Lake was analysed (Chapter 5).

The specific aims of this thesis were:

- To test various software and computational approaches for taxonomy, abundance and functional potential analyses of Antarctic metagenomes, to assess their reproducibility and robustness (Chapter 2). The methods that worked well with the Antarctic metagenomes annotated by Joint Genome Institute's Integrated Microbial Genomes (JGI's IMG) system were used to improve upon the preliminary Cavlab pipeline for metagenome analysis. The methods were tested because the Antarctic metagenomes represented data from environments that were not as well-characterised as other systems often used for building and testing software databases, especially clade-specific databases. Therefore, computational approaches had to be carefully selected for comprehensive analyses of Ace Lake seasonal data.
- To assess Ace Lake viral data, including viral contigs representing complete virus genomes, the most abundant viral contigs, and viral contigs potentially associated with some of the most abundant members of Ace Lake microbial community, such as *Micromonas* (a green alga), *Synechococcus* (a cyanobacterium), and *Chlorobium* (a GSB) (Chapter 3). These analyses were performed to determine the distribution and abundance of viral populations in various strata of Ace Lake (mixolimnion, oxycline, monimolimnion) in different seasons (summer, winter, spring). The association between virus and host abundances were also explored, to assess the potential impact of viral predation *vs* seasonal light availability, both of which can be responsible for changes in these phototrophic host abundances.
- To investigate genomic variation in the metagenome-assembled genomes (MAGs) of *Synechococcus* generated from metagenomes from different time periods and Ace Lake depths (Chapter 4). *Synechococcus* was identified throughout Ace Lake, in the oxic mixolimnion, oxic-anoxic interface and anoxic monimolimnion, and its abundance varied with season (described below in Chapter 3). Therefore, genomic variation analyses were performed to determine the presence/absence of different phylotypes, and potentially ecotypes, of *Synechococcus* in different seasons and Ace Lake depths.
- To investigate genomic variation in *Chlorobium* MAGs generated from metagenomes from different time periods (Chapter 5). *Chlorobium* abundance

varied with season (described below in Chapter 3), therefore genomic analyses were performed to identify potential phylotypes or ecotypes of Ace Lake *Chlorobium* from different seasons. With the availability of *Chlorobium* MAGs from two other Vestfold Hills stratified systems (Ellis Fjord and Taynaya Bay) that are known to harbour GSB (Burke and Burton, 1988a), the analysis of *Chlorobium* was expanded to assess its endemicity to the Vestfold Hills.

2. Computational approaches to analyse metagenomic data

2.1 Introduction

Analysis of large sequencing datasets, such as metagenomes, requires the use of software and methods capable of performing large-scale data analysis, some of which were discussed in Chapter 1. Although various computational tools for metagenome analysis are available, none of them can be used for complete analysis of a metagenome. Instead, different tools for metagenome analyses, such as taxonomy, abundance and functional potential, have to be used separately and their results have to be combined and interpreted for comprehensive metagenome analysis. A good way to combine the use of multiple software on one dataset is to create a pipeline that takes in the input data, calls specified software to perform various analyses, and generates results in specific formats. This also allows for time-efficient parallel runs of multiple software on a dataset. For the development of a pipeline for such comprehensive analysis of metagenomes, various aspects, such as the type of input data available, the purpose of analysis, and the type of output generated, need to be considered before selecting appropriate software/approaches. In this chapter, various software/approaches were tested to assess their capability to analyse the microbial community and functional potential of Antarctic metagenomes annotated by JGI's IMG system. The computational approaches found to be suitable for analysis of these IMG-annotated metagenomes were incorporated in a pipeline.

2.1.1 Antarctic metagenomes

The metagenomes discussed in this chapter were sequenced from samples collected from various Antarctic lakes. The water samples were sequentially extracted onto large format filters of sizes 20, 3, 0.8, and 0.1 μm, and some 0.1 μm filtrates were further concentrated through tangential flow filtration. DNA was extracted from the biomass on the filters as described previously (Ng et al, 2010; DeMaere et al, 2013; Tschitschko et al, 2018). The metagenomes were either sequenced at J. Craig Venter Institute (JCVI) using 454 and Sanger sequencing methods or at JGI using Illumina technology. JGI-sequenced metagenomes were initially assembled using Megahit (Li et al, 2015; Li et al, 2016) (referred to as Megahit-assembled metagenomes hereafter). With the change in

JGI's assembly method from Megahit to metaSPAdes (Nurk et al, 2013; Nurk et al, 2017), all Megahit-assembled metagenomes and any newly sequenced metagenomes were re-assembled/assembled using metaSPAdes (referred to as Spades-assembled metagenomes hereafter). JCVI-sequenced metagenomes were assembled in-house using metaSPAdes. All metagenomes were annotated by JGI's IMG system using their annotation pipeline (Huntemann et al, 2015). The descriptions of the metagenomes — their sample collection site, depth, and filter fraction; IMG genome ID; metagenome size; and total protein coding genes are provided Appendix A: Table A1. Ace Lake samples were collected from the surface (referred to as Upper 1 or U1); mixolimnion or upper oxic zone (referred to as Upper 2 or U2 and Upper 3 or U3); oxic-anoxic interface or oxycline (referred to as Interface or I); and monimolimnion or lower anoxic zone (referred to as Lower 1 or L1, Lower 2 or L2, and Lower 3 or L3).

2.1.2 Computational software/approaches tested for metagenome data analysis

Various software were tested for the development of an in-house pipeline that would perform taxonomic classification and explore functional potential of Antarctic metagenomes annotated by JGI's IMG system (Figure 2.1). These software were tested to assess the reliability and robustness of their analysis of Antarctic metagenomes (described below in section 2.2). Additional software were utilized for further genome-level analyses of the pipeline outputs and for statistical analyses (Figure 2.1).

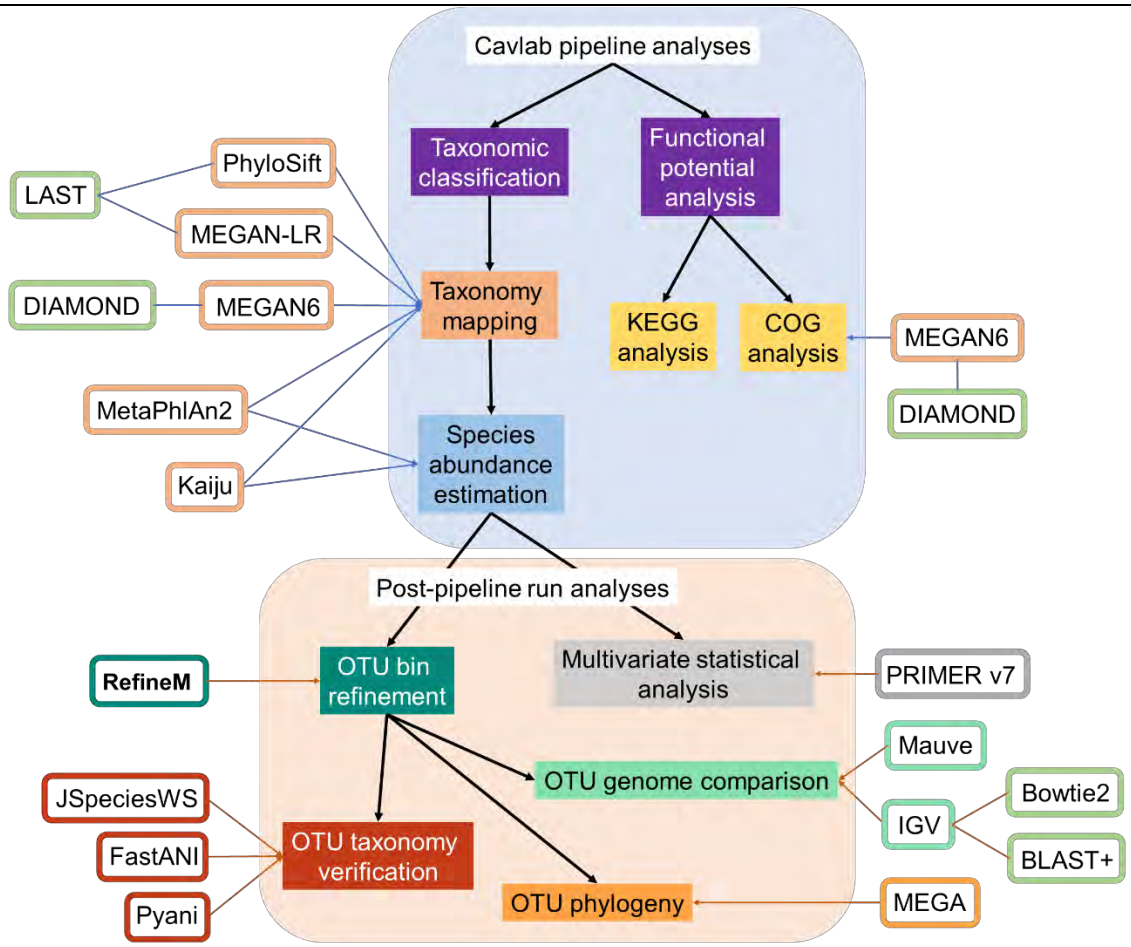


Figure 2.1 Software tested for the development of a pipeline for Antarctic metagenome analysis. The in-house metagenome analysis pipeline (Cavlab pipeline) was developed for taxonomic classification and functional potential analysis of Antarctic metagenomes. LAST and DIAMOND were tested for sequence alignment of reads, proteins, or contigs. PhyloSift, MetaPhlAn2, and Kaiju were tested for read taxonomic classification, whereas MEGAN-LR was tested for contig taxonomic classification. MEGAN6 was used for protein taxonomic classification as well as COG function analysis. The taxonomy output of the Cavlab pipeline was further examined at genome-level. RefineM was used for refining the OTU bins to produce high-quality OTU bins, whereas JSpeciesWS, fastANI, and pyani were used for calculating the ANI of the refined OTU bins. Specific OTUs of interest and MAGs were compared to their closest related reference genomes using BLAST+/Bowtie2 and Mauve alignment algorithm, and were visualised using IGV and Mauve, respectively. MEGA was used for assessing the evolutionary relationship between specific OTUs of interest and their closest related species. The abundance output of the Cavlab pipeline was used for statistical analysis of the metagenomes using PRIMER v7.

Described below are the software tested and/or used for analysis of Antarctic metagenomes (discussed in Chapter 1 section 1.4):

PhyloSift software was developed for the phylogenetic classification of metagenomes as well as genomes. It is dependent on other computing languages and software, such as Perl, HMMER (http://hmmer.org/), RAxML (Stamatakis, 2014), FastTree (Price et al, 2010), and Pplacer, for the taxonomic classification of DNA or protein sequences and for creating phylogenetic trees from the outputs. PhyloSift uses LAST alignment tool to align read, protein, or contig sequences against its database of core and extended set of markers. It uses hmmalign module of HMMER for taxonomic classification of DNA sequences <600 bp and protein sequences, and uses Infernal for taxonomic classification of DNA sequences >600 bp. The output is then converted to a phylogenetic tree using Pplacer from the Guppy software kit (Ueno et al, 2003).

DIAMOND is an alignment tool that can be used for aligning DNA or protein sequences to a protein database, such as the NCBI-nr protein database. For DNA to protein alignment, it translates the DNA prior to alignment. DIAMOND can align short read sequences much faster (20,000 times faster) than the BLASTX alignment tool, which also aligns translated DNA to protein databases (Buchfink et al, 2014).

LAST can be used for the alignment of DNA as well as protein sequences against a nucleotide or protein database. LAST tool uses an 'adaptive seed' algorithm (described in Chapter 1 section 1.4.1.1) that allows for better alignment of long sequences, such as contigs (Kielbasa et al, 2011).

MEGAN6 is a collection of tools that can be used to analyse metagenomes. Its command-line options allow for taxonomic classification and functional potential analyses of reads and proteins, which need to be aligned to a database prior to MEGAN6 classification using specific mapping files (available from https://software-ab.informatik.uni-tuebingen.de/download/megan6/welcome.html). Its most commonly used command-line modules include blast2rma, which uses a tab-delimited alignment output file, and daa2rma, which uses the DIAMOND alignment output file. It allows for some functional potential analyses using the mapping files for InterPro, eggNOG, and KO databases, which provide GO, COG, and KO annotations of proteins, respectively. MEGAN6 also has an interactive GUI, which can be used for taxonomic classification and functional analyses of reads and proteins, and for preparing bar charts, clusters, PCoA (Principal Coordinates Analysis) plots, networks, etc. The MEGAN6 GUI also allows for comparative analysis of multiple datasets, by using the 'Compare' mode with MEGAN6 outputs. MEGAN6 has a community edition that is free to use and an

ultimate edition, which is a paid version that includes the latest KO database mapping files for functional potential analysis.

MEGAN-LR is an off-shoot of MEGAN6 that was created specifically for the taxonomic assignment of contigs and long reads. Its command-line options for taxonomic classification and functional analyses are similar to MEGAN6, with an additional set of options for long reads and contigs. As with MEGAN6, MEGAN-LR can map only pre-aligned sequences, and LAST alignment tool has been recommended for use with MEGAN-LR (Huson et al, 2018). The latest version of the MEGAN6 community edition GUI includes the additional MEGAN-LR options.

MetaPhlAn2 is a software specifically designed for metagenome phylogenetic analysis. Apart from taxonomic classification, it calculates the relative abundances of the taxa identified in the metagenomes. It uses Bowtie2 for alignment of reads to its clade-specific marker database, which is available for download as part of the software.

Kaiju software was developed for the taxonomic classification of metagenome reads. It uses a k-mer based algorithm to align reads against a protein database, such as the NCBI-nr protein database, and calculates relative abundances by counting the number of reads assigned to a taxon relative to the total reads in the metagenome. Of the two algorithm modes of Kaiju alignment, namely MEM and Greedy (described in Chapter 1 section 1.4.1.2), the Greedy mode was reported to be more sensitive and precise in the taxonomic classification of 250 nucleotide long Illumina paired-end reads (Menzel et al, 2016).

RefineM software was developed to assess genome completion and contamination. It can also be used to assess the taxonomic composition of an OTU bin, to refine an OTU bin, and to generate high quality bins for genomic analyses. RefineM assesses the GC content, TNF, and coverage of the OTU bin contigs, based on which it identifies the outlier sequences that do not belong in the OTU bin. Additionally, it has modules that identify the genes on the OTU contigs and assign taxonomy to the genes and contigs in the OTU bin. RefineM taxonomy output, prepared using Krona (Ondov et al, 2011), can be visualised in an internet browser, showing the taxonomic composition of the OTU bin.

FastANI, ***pyani***, and ***JSpeciesWS*** are tools for measuring ANI. FastANI uses the MashMap (Jain et al, 2018a) algorithm for pairwise alignment of a query and a

reference sequence. It can be used to calculate the ANI of multiple OTUs at a time, by providing a list of OTU and reference sequence files in the command-line options. Pyani is a python module used for calculating ANI and TNF of sequences for genome comparison. Pyani can calculate ANI using three alignment methods — ANIb uses BLAST+, ANIm uses MUMmer (Kurtz et al, 2004), and ANIblastall uses legacy BLAST for alignment. It can be used for calculating ANI of multiple sequences at a time, by providing a list of all sequence files (query as well as reference) in the command-line. It computes an all-versus-all alignment, where each OTU/genome is aligned to all other OTUs/genomes provided as input. Pyani also calculates the alignment fraction, i.e., the percentage of query sequences that align a reference sequence. JSpeciesWS is an online service for ANI and TNF calculation (http://jspecies.ribohost.com/jspeciesws/#analyse). Similar to pyani, it can align sequences using ANIb or ANIm method, and it can either perform an all-versus-all alignment or a reference genome can be specified. It also provides a measure for alignment fraction.

The ***contig alignment*** and ***IGV*** approach can be used for the genomic analysis of specific OTUs identified in a system, allowing for direct comparison between an OTU and its closely matching reference genome. For contig alignment, Bowtie2 and blastn or megablast module of BLAST+ can be used for aligning the contigs from a metagenome, OTU, or MAG to a reference genome. Contig alignment can also be used to identify contigs of interest, e.g., viral contigs in metagenomes. IGV is a Java-based visualisation software that was developed for interactive analysis of large datasets. The contig alignment output files, namely, BAM and BAI format files, can be visualised using the IGV GUI, which shows a direct comparison between the reference and query sequences, highlighting mismatches and indels in the query sequence. Also, multiple alignment BAM files can be viewed simultaneously on IGV, making it easy to compare data from different samples.

Mauve is another alignment and visualisation tool that can be used for genomic analysis. During multiple sequence alignment, it identifies conserved regions between the reference and query sequences, and in the visual output, it represents them in the form of aligned segments, referred to as locally collinear blocks in Mauve. The 'progressiveMauve' algorithm of Mauve is recommended for sequence alignment (Darling et al, 2010).

MEGA is an alignment and visualisation tool that can be used for diverse purposes, including sequence alignment and constructing phylogenetic trees. The phylogenetic placement of an OTU can identify its closest related species group, which might provide insight into its probable function and interaction with its environment. Phylogenetic trees can be prepared using SSU rRNA genes or other clade-specific markers, such as bacteriochlorophyll A (BclA) protein sequence.

PRIMER v7 is a statistical analysis tool that provides many options for multivariate analysis of multiple samples/metagenomes, including options for assessing similarity/dissimilarity between samples, sample clustering, calculating a variety of species diversity measures, and preparing PCoA and PCA (Principal Component Analysis) plots. It can also be used for abundance analysis as well as to analyse environmental factors or even to assess the relationship between the two.

2.1.3 Aims

The main aim was to improve a preliminary metagenome analysis pipeline, named Cavlab pipeline, for the in-depth analysis of time-series metagenomes from Ace Lake in the Vestfold Hills. For this purpose, various software and computational methods were tested on Antarctic metagenomes, and the most reliable and robust approaches were incorporated in to the pipeline. The Cavlab pipeline was specifically developed to handle metagenomes generated by JGI's IMG system. The pipeline was organised to be used with IMG data — to efficiently use the input IMG folder structure, perform a list of analyses and generate outputs into specified folders that can be easily accessed and analysed using other in-house scripts. Additional software/methods were also used for the genomic and statistical analyses of the outputs generated by the pipeline. The software/approaches were tested on metagenomes from Ace Lake, Deep Lake, Club Lake, Organic Lake, and Rauer Island lakes.

Apart from improving the Cavlab pipeline, a specific aim was to develop a pipeline for analysis of archaea COG (arCOG) for the purpose of studying the functional potential contribution of archaea in metagenomes from archaea-rich environments (Appendix D). This pipeline relied on the output of DIAMOND and MEGAN6 protein taxonomy component of Cavlab pipeline and was tested on a Megahit-assembled metagenome from Deep Lake surface (Appendix A).

2.2 Method development

2.2.1 Improving the preliminary Cavlab pipeline

The preliminary in-house metagenome analysis pipeline (referred to as Cavlab pipeline v1.2 hereafter; Appendix B) was written using Python v3.5.2, for the analysis of metagenomes sequenced and annotated by JGI's IMG system. Cavlab pipeline v1.2 included specific metagenome analyses: (i) read taxonomic classification using PhyloSift, (ii) protein taxonomy and abundance analysis using DIAMOND and MEGAN6, (iii) functional potential analyses using the IMG COG and KO annotation files (hereafter referred to as metagenome COG and KEGG files), and (iv) initial steps for generating MAGs using MetaBAT (Kang et al, 2015). Each component of the pipeline was tested to assess its suitability for the analysis of Antarctic metagenomes (Appendix A) and changes were made to fix any errors incurred during pipeline runs (Table 2.1). The pipeline code was often modified to improve the output folder structure and input file verification step, to update software and database versions, and sometimes to reduce the computational resources required to run a pipeline component (Table 2.1). Additional analytical components were also added to the pipeline to generate specific outputs of use, and the components were tested on Antarctic metagenomes to assess their suitability (described in detail in the sections below).

Table 2.1 Cavlab pipeline versions — issues identified and changes made to improve the pipeline. ORF, open reading frame.

Cavlab pipeline version [development date]	Issues **(I)** and changes **(C)** in the Cavlab pipeline, which was run on UNSW Katana computer cluster
v1.3a [23 March, 2017]	**I:** The time taken to run the PhyloSift software on Katana (referred to as wall time) exceeded the maximum time limit (200 h) available for Katana runs. There were problems with the automatic updates of the PhyloSift database. **C:** PhyloSift v1.0.1 was downloaded to Katana scratch node and was used for the pipeline PhyloSift runs. Additionally, '*--config flag*' was added to the PhyloSift command-line to force it to use the config file *phylosiftrc*, which contained instructions for the software to use specific downloaded versions of the PhyloSift

	database. This prevented automatic update of PhyloSift database during Katana runs.
v1.3b [30 March, 2017]	**I:** The date in the output head folder name (Cav_LaunchDate) was not in a proper format and was confusing. **C:** The head folder naming format was modified to *Cav_YYMMDD*, i.e., 30 Mar 2017 would now be Cav_170330 in place of Cav_2017330.
v1.3b.1 [18 April, 2017]	**I:** PhyloSift software runs on Katana still exceeded the maximum wall time. **C:** PhyloSift runs in the pipeline were stalled.
v1.4.1 [20 April, 2017]	**I:** The COG and KEGG analyses run on some of the large metagenomes exceeded Katana maximum wall time. **C:** The COGKEGG python script wall time was increased from 48 h to 60 h, to accommodate for runs on large metagenomes. Additionally, the PBS job script (run on Katana) output and error files were merged by giving the command '#PBS -j oe' in the job script, to keep a track of the errors that spawn during Katana runs.
v1.4.2 [17 May, 2017]	**I:** Input file name error — due to similar file names, wrong protein files were being selected for analysis. The COG and KEGG analyses run on the large metagenomes still exceeded Katana maximum wall time. **C:** The protein file selection criteria was improved to ensure that protein files associated with assembled data were selected for analysis, and not the unassembled data protein files. The COGKEGG python script wall time was further increased to 96 h.
v1.4.2a [25 May, 2017]	**I:** The COG and KEGG analyses run on the large metagenomes still exceeded Katana maximum wall time. **C:** The COGKEGG python script wall time was increased to 120 h.
v1.5 [28 May, 2017]	**I:** The COG and KEGG analyses run on the large metagenomes still exceeded Katana maximum wall time. **C:** The COGKEGG python script was split into individual COG and KEGG scripts. COG runs were found to be more time-consuming and often prevented KEGG runs from initiating, if the wall time exceeded maximum limit during the COG run. COG wall time was kept at 120 h, but KEGG wall time was reduced to 12 h.
v2.0 [3 July, 2017]	**I:** The COG analysis run had high wall time (120 h).

	C: COG python script was modified so that the processing time was reduced from >100 h to <10 mins. This was achieved by using dictionaries, in place of lists, for storing and handling variable values. The wall time for COG analysis was changed to 12 h. NCBI COG conversion database file was updated using data available on ftp://ftp.ncbi.nih.gov/pub/COG/COG2014/data. PhyloSift component was removed, because the time taken to run it on a metagenome exceeded the maximum wall time (200 h) available on Katana at the time. CRISPR script was added to the pipeline that used the IMG CRISPR annotation file as input. The output would include all spacer sequences along with the IDs of the contigs in which they were identified. MEGAN script was updated to include KEGG mapping file (containing data from Feb 2015; prepared by the developers of MEGAN6), as an additional functional potential analysis.
v2.1 [13 July, 2017]	**I:** JGI IMG changed the output folder structure of newly sequenced and annotated metagenomes. **C:** For metagenomes with inconsistent IMG folder structure, new commands were added for identification of the correct folder containing the filtered read sequences, and associated changes were made to the MetaBAT preparation script.
v2.2 [7 August, 2017]	**C:** The latest NCBI-nr protein database (July 2017) was downloaded (from ftp://ftp.ncbi.nlm.nih.gov/blast/db/FASTA/), ~80 GB data. MEGAN6 version was also updated to v6.8.18, along with the accession ID-based taxonomy and function mapping files (downloaded from https://software-ab.informatik.uni-tuebingen.de/download/megan6/welcome.html) corresponding to the updated NCBI-nr protein database. The latest NCBI database included protein sequences and their corresponding accession IDs, but did not include protein GI numbers. Therefore, the GI numbers in the old MEGAN6 mapping files did not match the database.
v2.2a [8 August, 2017]	**C:** CRISPR script was updated to produce a FASTA file of the spacer sequences, with their position in a contig and corresponding contig ID mentioned in the header. The spacer sequences were to be used for aligning against viral contigs for virus-host analysis.

v2.2b [10 August, 2017]	**I:** MEGAN6 KEGG mapping file did not match the latest NCBI database and caused error. **C:** KEGG mapping file was removed from MEGAN6 analysis. It contained protein GI numbers, and not accession IDs, and therefore, did not match the updated, accession ID-based NCBI-nr protein database. The updated KEGG mapping file was only available to the users of MEGAN6 Ultimate Edition (paid version that provides the latest mapping files to KO database). MEGAN6 script was updated to include the InterPro mapping file for additional functional analysis based on GO data.
v3.0 [2 February, 2018]	**I:** In MEGAN6 script runs, abundance calculation was not performed, because the software did not accept the contig read depths mentioned in corresponding protein sequence headers. **C:** MetaPhlAn2 was added to the pipeline as a method for read-based taxonomic classification and relative abundance estimation. LAST and MEGAN-LR were added to the pipeline as a method for contig taxonomic classification. Protein sequence file pre-processing step was modified to add protein names to the protein sequence headers in place of contig read depths, to support functional potential analysis in place of abundance calculation. CRISPR script was removed from the pipeline, as a more in-depth virus analysis of the metagenomes would be available on IMG-VR (Páez-Espino et al, 2017). MetaBAT preparation script was also removed from the pipeline. Output folder structure was modified to make it more comprehensive. The updated NCBI-nr protein database (December 2017) was downloaded, ~87 GB data. Updated MEGAN6 mapping files were also downloaded.
v3.1 [15 February, 2018]	**I:** MetaPhlAn2 output excluded 99% of the filtered reads. **C:** MetaPhlAn2 was removed from the pipeline, due to issues with its database. The database did not include most of the species previously identified in Antarctic metagenomes, thereby, giving biased abundance estimations.
v3.1a [19 February, 2018]	**C:** Modifications were made to the script that picked data from the coverage and mapping files, to make the process slightly faster and the code more comprehensive.

v3.2 [27 February, 2018]	**C:** Modifications were made to the script that read the contig sequence file, to ensure that the scaffold sequence file in the QC_and_Genome_Assembly folder would be selected and read.
v3.3 [1 March, 2018]	**C:** COG and KEGG python scripts were altered to acquire the total coverage of ORFs that did not fall under any of the COG categories or specified KO numbers. The COG abundances were now represented as absolute abundances, and no longer represented as a fraction of the total abundance of all proteins (sum of corresponding contig read depths) in the metagenome. The formulae for KEGG pathway/enzyme abundance calculation were also changed. The head folder name would now include the pipeline version as well.
v3.3a [10 April, 2018]	**C:** The updated NCBI-nr protein database (March 2018) was downloaded, ~93 GB data. MEGAN6 mapping files corresponding to the updated NCBI-nr protein database were also downloaded.
v4 [28-29 May, 2020]	**C:** Output folder structure was updated, including changing some folder names. For contig taxonomy, LAST and MEGAN-LR methods were replaced by a method that uses the IMG protein taxonomy annotation file (Phylodist file). Additional KO numbers and pathways/enzymes were added to KEGG functional analysis. Four more KEGG database files were added to the pipeline — one associated with sulfur metabolism and three with methane oxidation or nitrification. Latest MEGAN6 mapping files were downloaded, corresponding to the NCBI-nr protein database downloaded in July 2019, ~132 GB data.
v4.1 [13 June, 2020]	**I:** JGI IMG changed the standard filenames of annotated data in the newly sequenced and annotated metagenomes. **C:** The latest file designations used by JGI for the metagenome annotation files were added to the script in the resource files verification component. The script was modified to accurately differentiate between coverage files that used Contig IDs and the ones that used Scaffold IDs, and to appropriately use both. The former was associated with metagenomes that were sequenced outside of JGI, but annotated by JGI IMG, and did not require a scaffold to contig mapping file.

	Minor issues with the latest metagenome COG files were also fixed. The second column in the COG files are blank, so the script prepares a new COG file for these, with the second column removed.

The latest version of the pipeline (referred to as Cavlab pipeline v4.1; Appendix C) has three main analytical components: (i) contig taxonomic classification using the IMG protein taxonomy annotation file (hereafter referred to as Phylodist file) and OTU abundance calculation using the contig coverage files; (ii) protein taxonomy and function analysis using DIAMOND and MEGAN6 with protein sequence file; and (iii) functional potential analyses using the metagenome COG and KEGG files.

2.2.2 Taxonomic identification

2.2.2.1 DIAMOND and MEGAN6

DIAMOND and MEGAN6 were a part of Cavlab pipeline v1.2, and were used for protein taxonomic diversity analysis (Appendix B). In Cavlab pipeline v2.0, KEGG analysis was added to the MEGAN6 runs using a MEGAN6 KEGG mapping file corresponding to data in KO database from February 2015 available at the time. DIAMOND/MEGAN6 module of Cavlab pipeline v2.0 was tested on Megahit-assembled metagenomes from Ace Lake 2008 and Deep Lake 2013–2015 (Appendix A), two very different Antarctic lake systems, to assess its reliability and robustness. Some taxonomy data were available from Ace Lake (Lauro et al, 2011) and Deep Lake (DeMaere et al, 2013), which were used as reference to assess the output of DIAMOND/MEGAN6 runs. The method would be considered robust, if its output was reliable and comparable to previous observations from different lake systems.

Cavlab pipeline v2.0 – DIAMOND and MEGAN6

```
export _JAVA_OPTIONS="-Xmx55g"

diamond makedb --in nr_Nov2016.fasta -d nr_Nov2016 -p 8

diamond blastp -d nr_Nov2016 -q ProteinSequenceFile.faa -a Output.daa -e 0.001 -p 8

diamond view -a Output.daa > -o Output.tab -f tab

blast2rma -r ProteinSequenceFile.faa -i Output.tab -o Output.rma -g2t gi_taxid_prot.dmp.gz -a2eggnog acc2eggnog-June2016X.abin -g2kegg gi2kegg-Feb2015X.bin -f BlastTab -mag -fun EGGNOG KEGG
```

Prerequisites (version): DIAMOND (v0.8.4); Java (v8u45); MEGAN6 (v6.4.5).

Katana resources: nodes = 1, processors = 8, memory = 64 GB, wall time = 48 h.
export _JAVA_OPTIONS is a Java command that was used to set the maximum heap size to 55 GB (-Xmx55g), to prevent MEGAN6 from running out of memory.
diamond makedb prepared an index file (-d) for the NCBI-nr protein database (--in) using parallel runs on 8 processors (-p 8). This needs to be run only once on a database, to create index files. All subsequent alignment runs used these indexed files.
diamond blastp aligned the query protein sequences (-q) against the database index files (-d) using parallel runs on 8 processors (-p 8). The output alignment file (-a) displayed only the alignments with e-value ≤0.001 (-e 0.001).
diamond view converted the DIAMOND output file (-a) to a tabular format (-f), to generate a tab-delimited file (-o) that could be used with blast2rma module of MEGAN6.
blast2rma mapped protein GI numbers to taxonomy (-g2t) and accession IDs to eggNOG database (-a2eggnog) to assign taxonomy and COG numbers to metagenome protein sequences (-r), respectively, based on the data in the input alignment file (-i).

- -f BlastTab specified the alignment file format.
- -fun EGGNOG KEGG specified that the eggNOG database-based COG (EGGNOG) and KO database-based KEGG (KEGG) functional analyses (-fun) should be performed.
- -mag specified that taxonomic assignments should be coverage-based. The read depth of contigs corresponding to the proteins were mentioned in the protein sequence headers.

To accommodate the change in the NCBI-nr protein database (the sequence headers were changed from GI numbers to accession IDs), the NCBI database and the MEGAN6 mapping files were updated in Cavlab pipeline v2.2. Additionally, InterPro database-based functional potential analysis was added to the MEGAN6 runs in Cavlab pipeline v2.2b, using the MEGAN6 InterPro mapping file.

Cavlab pipeline v2.2b – DIAMOND and MEGAN6

```
export _JAVA_OPTIONS="-Xmx55g"

diamond makedb --in nr_July2017.fasta -d nr_July2017 -p 8

diamond blastp -d nr_July2017 -q ProteinSequenceFile.faa -a Output.daa -e 0.001 -p 8

diamond view -a Output.daa -o Output.tab -f tab

blast2rma -r ProteinSequenceFile.faa -i Output.tab -o Output.rma -a2t prot_acc2tax-
May2017.abin -a2eggnog acc2eggnog-Oct2016X.abin -a2interpro2go acc2interpro-
```

Nov2016XX.abin -bm BlastP -f BlastTab -ram readMagnitude -fun EGGNOG INTERPRO2GO -v

Prerequisites (version): DIAMOND (v0.8.4); Java (v8u45); MEGAN6 (v6.8.18).

Katana resources: nodes = 1, processors = 8, memory = 96 GB, wall time = 48 h.

export _JAVA_OPTIONS, **diamond makedb**, **diamond blastp**, and **diamond view** were used as mentioned earlier in this section, except that an updated NCBI-nr database (downloaded in July 2017) was used in diamond makedb and blastp modules.

blast2rma was also used as described earlier in this section, except for the following changes in the options:

a) In place of -g2t used -a2t option, which allowed taxonomic classification using accession ID-based taxonomy mapping file.
b) Added InterPro database-based GO classification using the accession ID-based InterPro database mapping file (-a2interpro2go).
c) MEGAN6 version and all its mapping files were updated. In the new MEGAN6 version, -mag option had been replaced by -ram readMagnitude, which performed the same function.
d) -bm specified the blast module (blastp) used for generating the alignment file.
e) In the -fun option, INTERPRO2GO was added to indicate that InterPro database-based functional analysis should also be performed.
f) -v option printed MEGAN6 command-line options to the output log file, for future reference.

Additional command-line options were added to the DIAMOND blastp module, to further improve the protein-protein alignment. As per the recommendations of the developers of MEGAN6, the blast2rma module of MEGAN6 was replaced by daa2rma module, which can directly use the DIAMOND alignment output file.

Cavlab pipeline v3.0 – DIAMOND and MEGAN6

export _JAVA_OPTIONS="-Xmx64g"

diamond makedb --in nr_Dec2017.fasta -d nr_Dec2017 -p 8

diamond blastp --more-sensitive -d nr_Dec2017 -q ProteinSequenceFile.faa -o Output.daa -f 100 --algo 0 --index-mode 1 -p 8 -v

daa2rma -i Output.daa -o Output.rma -a2t prot_acc2tax-May2017.abin -a2eggnog acc2eggnog-Oct2016X.abin -a2interpro2go acc2interpro-Nov2016XX.abin -fun EGGNOG INTERPRO2GO -v

Prerequisites (version): DIAMOND (v0.9.10); Java (v8u91); MEGAN6 (v6.10.5).

Katana resources: nodes = 1, processors = 8, memory = 96 GB, wall time = 64 h.
export _JAVA_OPTIONS, **diamond makedb**, and **diamond blastp** were used as mentioned earlier in this section, except that the blastp module also included the following options:

- --more-sensitive was used for better alignment of long sequences, as recommended in DIAMOND v0.9.10 manual (also mentioned in DIAMOND v0.9.21 online manual; https://usermanual.wiki/Pdf/diamondmanual.1718530976/view).
- -f 100 set the output format to DIAMOND alignment archive (.daa), which was the default output format in previous versions.
- --algo and --index-mode are advanced options that were used for improved seed search, for reliable and faster alignments. The double-indexed seed search algorithm (–algo 0) was used, with a 4×12 index mode (--index-mode 1). In double-indexed algorithm, both query and reference sequences are indexed and are arranged as a dictionary of seed-location pairs, making the computations much faster (Buchfink et al, 2014).

daa2rma is a MEGAN6 module that works similar to blast2rma module, and was used to directly map the DIAMOND output file (.daa) to the NCBI-nr protein database.

In Cavlab pipeline v4.1, the updated versions of NCBI-nr protein database, DIAMOND, and MEGAN6 were used. The latest versions of the software have slightly different command-line options.

Cavlab pipeline v4.1 – DIAMOND and MEGAN6

```
diamond makedb --in nr_Jul2019.fasta -d nr_Jul2019 -p 16

diamond blastp --more-sensitive -d nr_Jul2019 -q ProteinSequenceFile.faa -o Output.daa -f 100 --algo 0 --index-mode 1 -p 16 -v

export _JAVA_OPTIONS="-Xmx96g"

daa2rma -I Output.daa -o Output.rma -a2t prot_acc2tax-Jul2019X1.abin -a2eggnog acc2eggnog-Jul2019X.abin -a2interpro2go acc2interpro-Jul2019X.abin -v
```

Prerequisites (version): DIAMOND (v0.9.31); Java (v8u121); MEGAN6 (v6.15.1).
Katana resources: nodes = 1, processors = 16, memory = 120 GB, wall time = 48 h.
export _JAVA_OPTIONS, **diamond makedb**, **diamond blastp**, and **daa2rma** were used as described earlier in this section, except that daa2rma no longer supports the -fun option, which used to highlight the functions to be performed; the program now automatically confirms this by reading the mapping files used.

2.2.2.2 LAST and MEGAN-LR

LAST and MEGAN-LR were introduced in Cavlab pipeline v3.0, as tools for assessing contig-based taxonomic diversity. MEGAN-LR has been specifically designed for use with long reads and contigs, and the developers recommended using LAST for contig alignment (Huson et al, 2018). As contig sequences are much longer than protein or read sequences, and much less likely to be shared through horizontal gene transfer (HGT), it was reasoned that their taxonomic classification would be more reliable. Therefore, the LAST/MEGAN-LR method was tested on Megahit-assembled metagenomes from different Antarctic lake systems, namely Ace Lake 2006 and 2008, Deep Lake 2006, 2008, and 2013–2015, Club Lake 2014, Organic Lake 2006, and Rauer Island lakes 1, 3, 6, 11, and 13 (Appendix A). The method was also tested on some of the Spades-assembled metagenomes from Ace Lake oxycline (0.1 µm-filter from 2008 and 2013) and Deep Lake surface (0.1 µm-filter from 2006 and <0.1 µm-filter from November 2014) available at the time (Appendix A). Spades-assembled Ace Lake oxycline metagenomes were selected to assess the population of GSB, a *Chlorobium*, previously observed at this depth of Ace Lake (Ng et al, 2010; Lauro et al, 2011). Deep Lake Spades-assembled metagenomes from 0.1 µm and <0.1 µm-fraction were randomly selected to assess the haloarchaeal population (members of *Halobacteria* class) population previously observed in Deep Lake (DeMaere et al, 2013). The method would be considered robust, if its output was reliable and comparable to the previous observations from both Ace Lake and Deep Lake.

Cavlab pipeline v3.0 – LAST and MEGAN-LR

```
export _JAVA_OPTIONS="-Xmx64g"

lastdb -vpcR01 -i10 -P16 nr_Dec2017 nr_Dec2017.fasta

lastal -P16 -fMAF -D10000 -R01 -F15 -pBL80 -v nr_Dec2017 ContigSequenceFile.fna Output.maf

maf2daa -i Output.maf -r ContigSequenceFile.fna -p 16 -o Output.daa -v

daa2rma -i Output.daa -o Output.rma -lg -alg longReads -ram readCount -a2t prot_acc2tax-Oct2017X1.abin -a2eggnog acc2eggnog-Oct2016X.abin -a2interpro2go acc2interpro-Nov2016XX.abin -fun EGGNOG INTERPRO2GO -v
```

Prerequisites (version): LAST (v914); Java (vu91); MEGAN-LR (v6.10.5).

Katana resources: nodes = 1, processors = 16, memory = 96 GB, wall time = 24 h.

lastdb is a LAST module that was used to create a database index, which was further used with the other modules of LAST, e.g., the lastal module. This needs to be run only once on a

database, to create index files. All subsequent alignment runs used these indexed database files. The command-line options were:

- -v printed the command-line options to the output log file, for future reference.
- -p specified that the input database sequences were proteins.
- -R01 was used to identify simple repeats in the database sequences and flag them. The first digit '0' allowed conversion of the input database sequences to uppercase, whereas the second digit '1' allowed conversion of simple repeats to lowercase.
- -c masked the lowercase letters in a sequence (simple repeats), so that during alignment these lowercase letters would be excluded from initial matches.
- -i10 instructed the program to perform at least 10 initial matches per query position.
- -P16 mentioned the number of processors used for parallel runs.

lastal is a LAST module that was used to align the query nucleotide sequences to the NCBI-nr protein database.

- -P16, -v, and -R01 worked the same as in lastdb module.
- -pBL80 used BLOSUM80 score matrix for aligning the query nucleotide sequences to the reference protein sequences.
- -D10000 defined the number of query letters to be used per random alignment.
- -fMAF specified the output file format.
- -F15 mentioned the frameshift cost, which also let the program know that the input query was a DNA sequence.

maf2daa is a LAST module that converted the input MAF alignment file (-i) to a DIAMOND alignment file (-o), using the contig sequence file as a reference (-r). The function ran parallelly on 8 processors (- p 8) and the command-line options were printed to the output log file (-v).

daa2rma was used as described earlier in section **2.2.2.1**. Additional MEGAN-LR options used for contig taxonomic assignment were:

- -lg specified that the input sequences were long sequences.
- -alg longReads used the algorithm specifically designed for long reads and contigs (Huson et al, 2018).
- -ram readCount allowed MEGAN-LR output to display the number of contigs assigned to a taxa node, when the output was visualised in MEGAN6 GUI. This was the default option in the previous versions of MEGAN6.

To utilise the output of LAST/MEGAN-LR to calculate OTU abundances in metagenomes, a python script was prepared. Prior to running the script, the LAST/MEGAN-LR output was used to prepare an input file containing the taxa identified in the metagenome and the contigs IDs assigned to the taxa. For this, the MEGAN-LR output file was opened in the MEGAN6 GUI and all the species nodes

were selected [**Select > Rank > Species**]. The taxa names and their corresponding contig IDs were exported to a document file [**File > Export > Text (CSV) Format**. *Choose data to export*: **taxonName_to_readName**; *Choose count to use*: **assigned**; *Choose separator to use*: **comma** > **OK** > FileA-species.doc > **Save**]. The data in the output document file was cleaned by removing the quotes from the taxa names, and the document was used for calculating OTU abundances in the metagenome using the python script below (text in red are comments and were not implemented by the python script):

```
#calculating OTU abundance using LAST/MEGAN-LR output
import csv
#### Reading data from the MEGAN-exported document file
d = {}
with open('FileA.doc', 'r') as datafile:
    for row in datafile:
        row = row.rstrip()
        row = row.split(',')
        d.setdefault(row[0],[]).append(row[1:]) #creates a dictionary with taxa names as keys and a list of Contig IDs; d = {species1:[[ContigA, ContigB, ContigC,…]], species2:[[ContigP, ContigQ, ContigR,…]],…}
species = list(d.keys()) #list of taxa names; species = [species1, species2,…]

#### Data prep: matching scaffold ID coverages to scaffold ID contig names (Contig IDs)
coverage = {}
with open(Metagenome contig coverage file, 'r') as covfile:
    covcsv = csv.reader(covfile, delimiter = '\t')
    next(covcsv)
    for row in covcsv:
        coverage[row[0]] = row[1]
maps = {}
with open(Contig ID to scaffold ID mapping file,'r') as mapfile:
    mapcsv = csv.reader(mapfile, delimiter = '\t')
    for row in mapcsv:
        maps[row[0]] = row[1]
covmap = {}
for i in range(len(maps)):
    covmap[maps[i]] = coverage[i]
```

```
#### Calculating taxa coverages
taxa = []
total = 0
issues = 0
sp_contigs = 0
for j in range(len(species)): # picking each taxon for abundance calculation
    contigs = []
    for sublilst in d[species[j]]:
        for y in sublilst:
            contigs.append(y) # list of contigs associated with a taxon
abund = 0
    for k in contigs:
        sp_contigs += 1 # count the total number of contigs associated with the taxa
        if k in covmap.keys():
            abund = abund + float(covmap[k]) # if a contig has coverage, sum it up
        else:
            issues += 1 # if a contig does not have a coverage, it is an error
    total = total + abund # calculate total abundance of species contigs
    taxa.append([species[j], abund])

#### Writing abundances to output file
with open(Output filename, 'w', newline = '') as writefile:
    csvfile = csv.writer(writefile, delimiter = '\t')
    for a in range(len(taxa)):
        csvfile.writerow([taxa[a][0], taxa[a][1]])
    csvfile.writerow(['Coverage_total', total])
    csvfile.writerow(['Contigs_without_coverage', issues])
    csvfile.writerow(['Contigs_analysed', sp_contigs])
```

For comparative metagenome analyses, OTU abundance files from different metagenomes were merged together using a MetaPhlAn2 python code:

MetaPhlAn2 file merge function

```
cd metaphlan2/utils

python3 merge_metaphlan_tables.py *-speciesAbn.txt > CombinedSpeciesAbn.txt
```

Prerequisites (version): Python (v3.6.5).

Katana resources: nodes = 1, processors = 1, memory = 16 GB, wall time = 1 h.
cd changed the current working directory to the 'utils' folder in MetaPhlAn2, where the 'merge_metaphlan_tables.py' python script for merging abundance data was stored.
python3 ran the merge_metaphlan_tables.py python script on the input OTU abundance files from multiple metagenomes and created a single text file containing abundances of all identified OTUs across all metagenomes. The asterisk (*) symbol allowed selection of multiple input text files with the suffix '-speciesAbn.txt' in the folder. Each input OTU abundance file had only two columns: (i) the taxa name column and (ii) an abundance value column. The output combined abundance file had a taxa name column followed by multiple abundance value columns (equal to the number of input abundance files). The abundance of an OTU was reported as zero in the metagenomes in which it was not identified.

The relative OTU abundances in a metagenome were calculated manually as below:

$$Relative\ OTU\ abundance\ (\%) = \frac{absolute\ OTU\ abundance}{total\ OTU\ abundance} \times 100$$

where, absolute OTU abundance was calculated by summing the read depths of contigs assigned to an OTU in a metagenome, using the python script mentioned above; total OTU abundance was calculated by summing the read depths of all contigs assigned to species-level in the metagenome.

The relative OTU abundances can also be calculated in PRIMER v7 software (section 2.2.7). The maximum relative abundance of an OTU in all metagenomes or in a specific set of metagenomes (from a depth or a time period) was described as the peak relative abundance of the OTU.

2.2.2.3 MetaPhlAn2

MetaPhlAn2 was introduced in Cavlab pipeline v3.0, alongside LAST and MEGAN-LR approach, for read-based taxonomy and relative abundance estimation. It was tested on seven Ace Lake 2008 metagenomes, at least one from each depth: U2_0.8 µm, U3_0.8 µm, I_0.8 µm, L1_3 µm, L1_0.8 µm, L2_0.8 µm, and L3_0.8 µm (Appendix A).

Cavlab pipeline v3.0 – MetaPhlAn2

```
cd metaphlan2

python3 metaphlan2.py FilteredReads.fastq.gz --input_type fastq --mpa_pkl
db_v21/mpa_v21_m200.pkl --bowtie2db db_v21/mpa_v21_m200 --nproc 8 --bowtie2out
BowtieOutput.bt2out.bz2 > RelativeAbn-allTaxaLevels.txt
```

```
python3 metaphlan2.py --input_type bowtie2out --mpa_pkl db_v21/mpa_v21_m200.pkl --nproc 8 -t reads_map BowtieOutput.bt2out.bz2 > ReadMap.txt

cd metaphlanOutput_file_path

grep -E "(s__)|(^ID)" RelativeAbn-allTaxaLevels.txt | grep -v "t__" | sed 's/^.*s__//g' > RelativeAbn-species.txt
```

Prerequisites (version): Bowtie (v2.3.2); Python (v3.5.2) with Numpy, Pandas, Biopython, SciPy, and Matplotlib packages installed.

Katana resources: nodes = 1, processors = 8, memory = 64 GB, wall time = 6 h

cd metaphlan2 changed the working directory to the folder where MetaPhlAn2 was installed.

python3 metaphlan2.py executed the MetaPhlAn2 python script run, which depends on python versions ≥3. The script first worked on the filtered reads FASTQ file to generate two outputs: a compressed Bowtie alignment file (--bowtie2out) and a taxa relative abundance text file. The relative abundance file contained all taxa (domain- to species-level) identified in the metagenome (taxa ID) and their corresponding relative abundances. In the second run of metahphlan2.py, the Bowtie alignment file was used to generate a text file containing filtered read IDs and their taxonomic assignments; only reads with matches to clade markers were mentioned in the text file. Other options included:

- --input_type specified the format of the input file (fastq or bowtie2out).
- --mpa_pkl mentioned the database used for MetaPhlAn2 taxonomic classification.
- --bowtie2db mentioned the reference database used for read alignment by Bowtie.
- --nproc specified the number of processors used for parallel runs.
- -t reads_map generated a file containing filtered read IDs and their taxonomy.

cd metaphlanOutput_file_path changed the working directory to the folder where the outputs of MetaPhlAn2 run were stored.

grep and **sed** are UNIX commands used to extract relative OTU abundances from the output file that contained relative abundances of all taxa levels, and store them in a separate file. Options included:

- grep -E "(s__)|(^ID)" selected all lines in the relative abundance output file that had species (s__) data in the taxa ID column (^ID), which was passed on to the next grep command.
- grep -v "t__" selected all data in the lines passed on to it that contained species data, and excluded any strain-level (t__) information. This species data, with strain name removed, was passed on to the next sed command.
- sed 's/^.*s__//g' edited the lines passed to it that contained species data. It removed all levels of taxonomy from the lines, except species name, by replacing ‘s__’ and everything before it with an empty string.

The MetaPhlAn2 database contains clade-specific marker genes identified from all three domains of life and viruses (Segata et al, 2012). However, the initial output of MetaPhlAn2 analysis of some of the Antarctic metagenomes showed that its database did not have marker genes for most of the species identified in Antarctic samples (Table 2.3). Therefore, the genetic markers for some of the species observed in DIAMOND/MEGAN6 outputs of Ace Lake and Deep Lake were added to the database using a python script prepared from the information provided on the software website (https://github.com/biobakery/MetaPhlAn). A few of the markers are mentioned in the python script below and a complete list of markers added to the MetaPhlAn2 database is provided in Appendix E.

```
#adding clade-specific markers to MetaPhlAn2 database
import pickle
import bz2
db = pickle.load(bz2.BZ2File('path/metaphlan2/db_v20/mpa_v20_m200.pkl', 'r'))
db['taxonomy']['k__Archaea|p__Euryarchaeota|c__Halobacteria|o__Haloferacales|f__Halorubraceae|g__Halohasta|s__Halohasta_litchfieldiae_tADL|t__GCF_900109065'] = 3332020
db['markers']['gi|1279136099|ref|CP024845.1|:42441-40969'] = {
                        'score': 0.0,
                        'len': 1473,
                        'taxon':
'k__Archaea|p__Euryarchaeota|c__Halobacteria|o__Haloferacales|f__Halorubraceae|g__Halohasta|s__Halohasta_litchfieldiae_tADL',
                        'clade': 's__Halohasta_litchfieldiae_tADL',
                        'ext': {}}
db['markers']['gi|645321082|ref|NR_118135.1|:1-1473'] = {
                        'score': 0.0,
                        'len': 1473,
                        'taxon':
'k__Archaea|p__Euryarchaeota|c__Halobacteria|o__Haloferacales|f__Halorubraceae|g__Halohasta|s__Halohasta_litchfieldiae_tADL',
                        'clade': 's__Halohasta_litchfieldiae_tADL',
                        'ext': {}}
ofile = bz2.BZ2File('path/ metaphlan2/db_v21/mpa_v21_m200.pkl', 'w')
pickle.dump(db, ofile, pickle.HIGHEST_PROTOCOL)
ofile.close()
```

2.2.2.4 Kaiju

Kaiju was tested on Ace Lake oxycline metagenomes (0.1 µm-filter from 2008 and 2013), to assess the *Chlorobium* population previously observed at this depth (Ng et al, 2010; Lauro et al, 2011). The software was also tested on Deep Lake surface metagenomes (0.8 µm-filter from 2006 and <0.1 µm-filter from November 2014), to assess previously observed haloarchaea population (DeMaere et al, 2013). These previously reported Ace Lake and Deep Lake microbial diversity data were used as references to assess the reliability of Kaiju output.

Kaiju

makeDB.sh -e -t 16

kaiju -t nodes.dmp -f kaiju_db_nr_euk.fmi -i FilteredReads1.fastq.gz -j FilteredReads2.fastq.gz -o Alignment.out -a greedy -e 5 -z 16 -v

kaijuReport -t nodes.dmp -n names.dmp -i Alignment.out -o SpeciesAbn.report -r species -v

Prerequisites (version): Perl (v5.20.1); Kaiju (v1.6.2).

Katana resources: nodes = 1, processors = 16, memory = 120 GB, wall time = 48 h

makeDB created an NCBI-nr protein database index for use with Kaiju. This needs to be run only once on a database, to create index files. All subsequent alignment runs used the indexed files. Options included:

- -e specified that NCBI-nr protein database must include sequences from fungal and microbial Eukarya, apart from Archaea, Bacteria, and Viruses.
- -t specified the number of processors used for parallel runs.

kaiju alignment tool was used to align paired-end filtered reads (-i and -j) against the NCBI-nr protein database index (-f). Options included:

- -a greedy, -e 5 specified that the greedy-5 alignment algorithm should be used for read alignment.
- -t specified the path to the nodes.dmp database file, which contained the taxon IDs of the NCBI database protein sequences.
- -z specified the number of processors used for parallel runs.
- -v printed the command-line options to the output log file, for future reference.

kaijuReport is a Kaiju module that mapped the taxon names (-n) in the names.dmp database file to the alignment output (-i), using the nodes.dmp database file as a reference (-t) for taxon IDs. Other options include:

- -r species specified that the relative abundances should be calculated and reported at species-level.

- -v was used as described in kaiju module above.

2.2.2.5 Protein taxonomy-based contig taxonomic classification and abundance estimation

Apart from the above-described methods for taxonomic classification and abundance estimation, a new python script was written for utilising data in the metagenome Phylodist files for contig taxonomy analysis. The script had two main components: (i) taxonomic classification of contigs based on protein taxonomies in the Phylodist file and (ii) calculating OTU abundances from contig lengths and read depths in the contig coverage file. Certain criteria were considered for protein taxonomy-based contig taxonomic classification:

(a) At least 30% of the genes identified on a contig must have a taxonomic assignment in the Phylodist file; if not, then the contig was unclassified.
(b) For a contig, the taxon to which most of its genes belonged was used as the taxonomic assignment of that contig.
(c) If all genes on a contig had different taxonomies in the Phylodist file, the contig was unclassified.

In the python script, which is now a part of the Cavlab pipeline v4.1 (Appendix C), the OTU abundances were calculated by summing the coverages (length × read depth) of all contigs assigned to the OTU. The coverages of contigs that were not assigned to any taxa, based on the above three criteria, were summed and referred to as 'unclassified abundance'. Additionally, the coverages of contigs that were not assigned to any taxa, because none of the genes identified on them had taxonomies in the Phylodist file, were summed and referred to as 'unassigned abundance'. The total metagenome abundance was the sum of coverages of all contigs in a metagenome, and included all OTU, unclassified, and unassigned abundances. The relative OTU abundances were calculated using PRIMER v7 (section 2.2.7).

2.2.3 Functional potential analysis

2.2.3.1 DIAMOND and MEGAN6

The options to include functional potential analyses, such as COG and InterPro, were added to MEGAN6 (section 2.2.2.1). MEGAN6 output file can be visualised in the MEGAN6 GUI and the functional output can be accessed through: **Window** > **Open**

EGGNOG Viewer/Open INTERPRO2GO Viewer for COG and GO data, respectively. For example, in the EGGNOG viewer, a dendrogram of COG categories would be available, with each node depicting the number of proteins assigned to it.

2.2.3.2 COG and KEGG functional potential analyses

A very specific set of methods were developed as part of the Cavlab pipeline v1.2, for the functional potential analysis of metagenomes annotated by JGI IMG, using the metagenome COG and KEGG files (Appendix B). These files contain metagenome protein IDs and their COG or KO number annotations, respectively.

The COG python script read the data in the metagenome COG file and assigned the COG numbers to their correct COG categories, using a COG conversion database file, which contained a list of all COG numbers and their respective COG categories. In Cavlab pipeline v1.2, the relative abundance of a COG category in a metagenome was calculated by summing the read depths of the contigs corresponding to the proteins assigned to the COG category and then dividing it by the sum of read depths of contigs corresponding to all the proteins annotated in the metagenome (Appendix B). Additionally, a COG category assignment ratio was calculated by dividing the number of proteins assigned to a COG category in a metagenome by the total number of proteins in the metagenome. In Cavlab pipeline v2.0, the COG conversion database file was updated using the COG data available on NCBI (ftp://ftp.ncbi.nih.gov/pub/COG/COG2014/data).

The COG abundance calculations were changed in Cavlab pipeline v3.3, such that the COG category abundances were reported as absolute abundances — sum of contig read depths corresponding to proteins assigned to the COG category, rather than relative abundances. This is because the denominator in the previous COG relative abundance calculations depended on the contig read depths corresponding to all proteins in the metagenome and did not compensate for multiple proteins from one contig being assigned to the same COG number/category, which inflated the value of the denominator and reduced the COG relative abundance. The COG category assignment was also no longer represented as a ratio of total proteins in the metagenome, but simply as the number of proteins assigned to a COG category.

The KEGG script was written to assess specific pathways/enzymes, using a specific set of KO numbers (Appendix F). The script read the data in the metagenome KEGG file

and selected proteins associated with specific KO numbers. The abundance of a KO number was calculated by summing the contig read depths corresponding to the proteins assigned to the KO number. In Cavlab pipeline v1.2, the abundance of a pathway/enzyme was calculated by averaging the abundance of the KO numbers associated with it (Appendix B). However, this calculation was changed in Cavlab pipeline v4, where the abundance of a pathway/enzyme was calculated by averaging the abundance of the KO numbers associated with the same reaction in the pathway, such as enzyme subunits, and by summing the abundance of KO numbers associated with different reactions in the pathway (Appendix C; also see Appendix F for a list of KO numbers).

Table 2.2 KO number databases for KEGG functional potential analysis. [A] Cavlab pipeline version specifies the pipeline version in which the KO number database was first added. All eight databases were part of the latest Cavlab pipeline v4.1 (Appendix C). [B] Each KO number had one database file associated with it. DSR, dissimilatory sulfate reduction.

Cavlab pipeline version[A]	Pathways (function)	KO number[B]	Enzyme name (EC number)	Number of protein sequences
v1.2	DSR (reduction); Sulfide oxidation (oxidation)	K00394	Adenylylsulfate reductase, subunit A (*apr*A; EC:1.8.99.2)	165
		K00395	Adenylylsulfate reductase, subunit B (*apr*B; EC:1.8.99.2)	156
		K11180	Dissimilatory sulfite reductase alpha subunit (*dsr*A; EC:1.8.99.5)	868
		K11181	Dissimilatory sulfite reductase beta subunit (*dsr*B; EC:1.8.99.5)	856
v4	DSR (reduction); Sulfide oxidation (oxidation)	K00958	Sulfate adenylyltransferase (*sat*; EC:2.7.7.4)	732
	Nitrification (Ammonia monooxygenase);	K10944	Methane/ammonia monooxygenase subunit A (*pmo*A-*amo*A;	208

Methane oxidation (Methane monooxygenase)		EC:1.14.18.3, EC:1.14.99.39)	
	K10945	Methane/ammonia monooxygenase subunit B (*pmo*B-*amo*B)	244
	K10946	Methane/ammonia monooxygenase subunit C (*pmo*C-*amo*C)	398

The KEGG component of Cavlab pipeline v1.2 depended on four KEGG database files associated with adenylylsulfate reductase (*aprA*, K00394; *aprB*, K00395) and dissimilatory sulfite reductase (*dsrA*, K11180; *dsrB*, K11181) that catalyse the sulfide oxidation and sulfate reduction redox reactions (Appendix B). These KEGG database files included protein sequences of the enzymes from various microbes, with their role (reduction or oxidation) in the microbes mentioned in the sequence headers. Four additional KEGG database files were prepared and added to the KEGG analysis script in Cavlab pipeline v4 (Table 2.2). One of these database files included protein sequences of sulfate adenylyltranserase (*sat*; K00958) from various microbes, with their role in sulfate reduction (reduction) or sulfide oxidation (oxidation) mentioned in the sequence headers. The other three database files were associated with K10944, K10945, and K10946, which represent subunits of the homologous enzymes ammonia and methane monooxygenase, and contained protein sequences from various microbes, along with their function as ammonia or methane monooxygenase mentioned in the sequence headers (Appendix C). The protein sequences of the KEGG database enzymes were downloaded from NCBI. Resources, such as the KEGG PATHWAY database (https://www.genome.jp/kegg/pathway.html), and various other online resources, were used to manually determine the role of the enzymes in the microbes they were sequenced from. The enzyme functions were added to the protein sequence headers using the python script below. The COG and KEGG analysis components of Cavlab pipeline v3.3 were tested on a 0.1 µm-filter Megahit-assembled metagenome from Deep Lake surface from Dec 2013 (Appendix A).

```
# adding enzyme functions to protein headers
recdata = {}
with open('InputFileA.txt', 'r') as mod: #file containing list of protein accession IDs, protein names, taxonomy, and the determined role
    modc = csv.reader(mod, delimiter = '\t')
```

```
    for row in modc:
        recdata[row[0]] = row[3] + '---' + row[1] + ' [' + row[2] + ']' #recdata = {'Accession ID':'Protein role---protein name---protein taxonomy'}

x = 0
with open('InputFileB.fasta', 'r') as seqs: #FASTA file containing protein sequences downloaded from NCBI
    with open('OutputFile.fasta', 'w') as outseqs:
        for rec in SeqIO.parse(seqs, 'fasta'):
            if rec.id in recdata.keys(): #check if the sequence header is among protein accesion IDs in File A
                recordID = recdata[rec.id].split('---')[0] + '$' + rec.id #change sequence header to 'Protein role$header'
                recordDesc = recdata[rec.id].split('---')[1] #add protein name as sequence description
                record = SeqRecord(seq = rec.seq, id = recordID, description = recordDesc)
                SeqIO.write(record, outseqs,'fasta')
                x += 1
            else:
                continue
```

2.2.3.3 arCOG functional potential analysis

The arCOG numbers were specifically considered for the analysis of metagenomes rich in archaea, such as metagenomes from Deep Lake, Club Lake, and some Rauer Island lakes (DeMaere et al, 2013; Tschitschko et al, 2018). A python script was developed to assess the arCOG number-based COG categorisation of proteins, and the script was tested on a 0.1 µm-filter Megahit-assembled metagenome from Deep Lake surface from December 2013 (Appendix A). The script was run on potential archaeal protein sequences that were gathered from the output of DIAMOND/MEGAN6 runs on the metagenome protein sequences. For this, the DIAMOND/MEGAN6 output file was opened in MEGAN6 GUI and the 'Archaea' node was selected. The protein IDs assigned to the 'Archaea' node were exported to a text file (namely Samplename.archaea.txt) — **File > Export > Text (CSV) Format** > *Choose data to export*: **readName_to_taxonName**; *Choose count to use*: **summarised**; *Choose separator to use*: **tab** > Samplename.archaea.txt. Note that the 'summarised' option not only considered the proteins assigned to the 'Archaea' node itself, but also all proteins

assigned to the nodes that fall under the 'Archaea' node (all archaeal phyla, class, order, family, genus, and species nodes). A folder named 'arCOGs' was created in the metagenome head folder (the folder containing IMG_Data subfolder) and the Samplename.archaea.txt file was uploaded to it. The arCOG pipeline v1.2 script was run from the metagenome head folder (Appendix D).

As a first step in the arCOG script, the Archaea protein IDs in Samplename.archaea.txt file were read and an archaeal protein sequence FASTA file was created from the sequences in the IMG protein annotation file. The archaea proteins were aligned against arCOG protein sequence databases (available from ftp://ftp.ncbi.nih.gov/pub/wolf/COGs/arCOG) using PSI-BLAST (Altschul et al, 1997). Based on the alignment output, the Cognitor module of COGsoft (Kristensen et al, 2010) assigned arCOG numbers to the protein sequences. These arCOG numbers were assigned COG categories based on their comparison with an arCOG conversion database file, which contained a list of all arCOG numbers and their corresponding COG categories. For comparison, a COG number-based COG categorisation of the archaeal proteins was performed, using the COG number annotations in the metagenome COG file. COG category abundances were calculated by summing the contig read depths corresponding to the archaea proteins assigned to a COG category. The number of archaea proteins assigned to a COG category were also calculated.

2.2.4 Refining and verifying OTU taxonomy

2.2.4.1 Refining OTU bins

The species identified using taxonomic classification methods are often the closest available matches in the reference database used for classification. Therefore, the species were referred to as OTUs, until their taxonomies were verified. RefineM was used to filter the OTU bins identified from the output of Cavlab pipeline v4.1 runs, specifically the output of contig taxonomy and abundance estimation component on Spades-assembled Ace Lake metagenomes. Among the OTUs identified in a metagenome, only the ones with abundance >1% (relative to the total metagenome abundance) were considered for further analysis. The contig sequences belonging to these abundant OTUs were gathered in their respective OTU FASTA files and were called OTU bins. Before running RefineM, the bin contigs were aligned against the

filtered reads from the metagenomes from which the bin contigs originated, using BBMap and Samtools.

BBMap alignment

```
export _JAVA_OPTIONS="-Xmx112g"

bbmap.sh ref=OTUcontigs.fasta

bbmap.sh in=Metagenome1.filtered.fastq ambig=all out=Output1.sam

samtools view -bS Output1.sam > Output1.unsorted.bam

samtools sort Output1.unsorted.bam -l 9 -@ 16 > Output1.bam

samtools index Output1.bam
```

Prerequisites (version): Java (v8u121); BBMap (v38.51); Samtools (v1.9).
Katana resources: nodes = 1, processors = 16, memory = 120 GB, wall time = 48 h
bbmap.sh aligned the filtered read sequences in the metagenome (in) against the OTU bin contigs (ref) and prepared a SAM alignment file as output (out). Note that multiple metagenomes were aligned against the OTU contig sequences using a single script, by adding the bbmap and samtools steps to the script for each additional metagenome. Similarly, multiple OTUs were aligned against multiple metagenomes using a single script, by merging contigs from multiple OTUs into a single FASTA file and using it as a reference (ref). The ambig=all option retained all top-scoring sites in an ambiguously aligned read.
samtools view prepared an unsorted BAM file (a binary version of a SAM file) from a SAM alignment file. The -bS option specified that the program should create an output BAM file (-b) from the input SAM file (S).
samtools sort module sorted and compressed the alignment data in the unsorted BAM file, using 16 parallel processors (-@ 16) and maximum level of compression (-l 9).
samtools index generated a BAI file (.bam.bai) that contained the BAM indexes.

The BAM files generated from the BBMap alignments were used for the RefineM runs. A folder structure was created within the folder where the BAM files were stored, i.e., a 'refineM' folder with a 'Contig_files' subfolder (FolderWithBAMfiles/refineM/Contig_files) was created. The OTU contig sequence FASTA files were placed in the Contig_files subfolder for RefineM reference.

RefineM

```
export PATH=$PATH:/krona/bin

export MALLOC_ARENA_MAX=1
```

```
rsync -a FolderWithBAMfiles/${TMPDIR}

cd ${TMPDIR}

refinem scaffold_stats -x fasta -c 16 --cov_all_reads --silent OTUcontigs.fasta
FolderWithBAMfiles/refineM/Contig_files ./Outputs FolderWithBAMfiles/*.bam

refinem outliers --silent ./Outputs/scaffold_stats.tsv ./Outputs/Outliers --cov_perc 1000000000 -
-no_plots

refinem call_genes -x fasta -c 16 –silent FolderWithBAMfiles/refineM/Contig_files
./Outputs/Genes

refinem taxon_profile -c 16 --silent ./Outputs/Genes ./Outputs/scaffold_stats.tsv
gtdb_r80_protein_db.2017-11-09.faa.dmnd gtdb_r80_taxonomy.2017-12-15.tsv
./Outputs/taxon_profile

refinem taxon_filter -c 16 --silent ./Outputs/taxon_profile
./Outputs/taxon_profile/taxon_filter.tsv

rsync -a ${TMPDIR}/ FolderWithBAMfiles/Outdata_copy
```

Prerequisites (version): Python (v2.7.15) with RefineM (v0.0.24) package installed; Prodigal (v2.6.3); HMMER (v3.2.1); BLAST+ (v2.6.0); DIAMOND (v0.9.10); Krona.

Katana resources: nodes = 1, processors = 16, memory = 120 GB, wall time = 48 h

export PATH added the path to the folder where Krona software was installed.

export MALLOC_ARENA_MAX=1 instructed the program to use a single memory pool for the RefineM runs, regardless of the number of processors used. This command is especially useful for programs that tend to use excessive memory.

rsync is a Linux tool that was first used to sync the BAM files to a temporary directory (TMPDIR). After the BAM files were used as input for RefineM runs and the outputs were generated in the temporary directory, rsync was used to copy the data from the temporary folder to the original folder with the BAM files (FolderWithBAMfiles). The -a option ran rsync in 'archive' mode, which not only synced the data, but also all associated attributes and permissions.

cd changed the current working directory to the temporary directory (TMPDIR).

refinem scaffold_stats module calculated contig statistics in the reference contig file (OTUcontigs.fasta), by comparing them with the BAM alignment files stored in the FolderWithBAMfiles folder and the individual OTU contig FASTA files in FolderWithBAMfiles/refineM/Contig_files. Note that when only one OTU was being refined, the contigs in the reference contig file (OTUcontigs.fasta) were identical to those in the individual OTU contig FASTA file in refineM/Contig_files folder. Other options included:

- -x fasta specified the input file format.
- -c 16 specified the number of processors used for parallel runs.
- --cov_all_reads used all reads for coverage estimation and not just the ones in proper pairs.
- --silent prevented the program from printing the output in the Katana run instance, which can be a large amount of data.

refinem outliers module assessed the output of 'refinem scaffold_stats' and identified contigs whose genomic characteristics, such as GC, coverage, and TNF, did not match those of other contigs in the OTU bin. The output was stored in the refineM/Outputs/Outliers folder. Other options were:

- --silent worked as described above.
- --cov_perc 1000000000 prevented the program from differentiating between contigs based on their read depths, since the read depths of contigs in an OTU can be different if they are from different metagenomes.
- --no_plots prevented the program from generating plots for contig genomic characteristics.

refinem call_genes module predicted genes on the OTU contigs and wrote the output to the refineM/Outputs/Genes folder. Other options were:

- -x fasta specified the input file format.
- -c 16 and --silent worked as described above.

refinem taxon_profile module performed taxonomic classification of the genes identified in the OTU contigs and stored the output in refineM/Outputs/Genes folder. It used DIAMOND to align the genes against GTDB and a GTDB taxonomy mapping file for taxonomic classification. The output was stored in the refineM/Outputs/taxon_profile/taxon_filter.tsv file. The '-c 16' and '--silent' options worked as mentioned above.

refinem taxon_filter module assessed the output of 'refinem taxon_profile' and identified OTU contigs whose taxonomy did not match the overall OTU taxonomy. The output was written to the refineM/Outputs/taxon_profile folder. The '-c 16' and '--silent' options worked as described above.

The OTU bins were refined using the output of the 'refinem outliers' module, stored in the refineM/Outputs/Outliers/outliers.tsv, which contained a list of the contigs that probably did not belong to the OTU along with the list of attributes (either GC or TNF or both) responsible for their exclusion. A python script was prepared to read this output and automatically decide which contigs should be included in the bin, because the value of their outlying attribute was quite close to the upper or lower boundary of the OTU bin's attribute (see script below). Contigs that were too different from the other bin

contigs were listed in a text file ‘outlierContigsList.txt’ and were excluded from the bin. The OTU bins were also refined based on the output of the ‘refinem taxon_profile’ module, which generated gene and scaffold taxonomy Krona outputs (HTML files) that were visualised on Google Chrome internet browser, showing the taxonomic composition of the OTU bin. This further allowed removal of spurious contigs that did not belong in the OTU bin.

```
#assessing outlier bin contig characteristics to decide whether to keep or remove them
with open('refineM/Outputs/Outliers/outliers.tsv', 'r') as inf:
   with open('outlierContigsList.txt', 'w', newline = '') as outf:
      infc = csv.reader(inf, delimiter = '\t')
      outfc = csv.writer(outf, delimiter = '\t')
      next(infc)
      for row in infc:
         if row[3] == 'GC': # check if the outlying attribute is GC content
            if float(row[4]) < float(row[6]) + 0.5 or float(row[4]) > float(row[7]) + 0.5: # check whether the contig GC is less than the OTUs ‘lower GC boundary + 0.5’ or more than its ‘upper GC boundary + 0.5’
               outfc.writerow([row[0]]) # if yes, then the contig is an outlier; add it to the outlier list
         elif row[3] == 'TD': # check if the outlying attribute is tetranucleotide frequency
            if float(row[8]) > float(row[10]) + 0.05: # check whether the contig TD is more than the OTUs ‘upper TD boundary + 0.5’

               outfc.writerow([row[0]]) # if yes, then the contig is an outlier; add it to the outlier list
         elif row[3] == 'GC,TD': # check if the outlying attributes both GC and TD
            if float(row[4]) < float(row[6]) + 0.5 or float(row[4]) > float(row[7]) + 0.5 or float(row[8]) > float(row[10]) + 0.05: # check whether the contig GC is less than the OTUs ‘lower GC boundary + 0.5’ or more than its ‘upper GC boundary + 0.5’ or the contig TD is more than the OTUs ‘upper TD boundary + 0.5’
               outfc.writerow([row[0]]) # if yes, then the contig is an outlier; add it to the outlier list
         else:
            continue
```

2.2.4.2 Verifying OTU taxonomy

The taxonomic classification of the OTUs identified in the metagenomes were verified in three steps. The first step relied on the Krona outputs of 'refinem taxon_profile' module of RefineM that were visualised in Google Chrome internet browser, which gave an idea of the overall taxonomic composition of the OTU bin. The taxonomies displayed in the Krona output were from GTDB, unlike the previous methods that used NCBI databases as reference for protein and contig taxonomic classifications. In the second step, the *16S/18S rRNA* gene identities of the OTUs were assessed against the SSU rRNA genes of their closest related taxa, which were manually downloaded from NCBI and were aligned against the OTU contigs using blastn module of BLAST+ v2.9.0. The OTUs were considered to belong to the same species as reference, if the SSU rRNA gene identity was >99% (Kim et al, 2014) and to the same genus as the reference, if the identity was >95% (Stackebrandt and Goebel, 1994).

The third step was to calculate the ANI of the OTUs against their closest related reference genomes. The complete or draft genomes of the reference taxa were manually downloaded from NCBI. ANI was initially calculated using the JSpeciesWS online service, where the query OTUs were uploaded one at a time and the reference genomes were added from GenomeDB, a genome database that is part of the JSpeciesWS service. ANI was calculated using the ANIb module that used BLAST+ for query and reference sequence alignment (Richter et al, 2016). However, this service had a file upload limit, and only OTU bins whose file size was between 0.02 to 15 MB could be uploaded. Therefore, other software, like fastANI and pyani, were used for ANI calculation of all OTUs. FastANI did not provide an estimate of the fraction of the query sequence that aligned to the reference (% alignment fraction). Moreover, its fragment length option (--fragLen) limited the ANI calculation to only contigs that were larger than the specified fragment length. Considering these issues, pyani was used for ANI calculation of the OTUs. The pyani output included ANI and alignment fraction measures of all OTUs and reference genomes mentioned in the input file.

ANI calculation using fastANI

```
fastANI --ql queryList.txt --rl referenceList.txt -o ANI-output.out -t 16 --fragLen 500
```

Prerequisites (version): MashMap (v2.0), FastANI (v1.1).

Katana resources: nodes = 1, processors = 16, memory = 120 GB, wall time = 12 h

fastANI was used for calculating the ANI of the OTUs against the reference genomes downloaded from NCBI. Other options include:

- --ql specified that the query was a list of FASTA files and their file names were mentioned in queryList.txt.
- --rl specified that the reference was a list of FASTA files and their file names were mentioned in referenceList.txt.
- --t 16 mentioned the number of processors used for parallel runs.
- -o specified the output file name.
- --fragLen specified the minimum fragment length used while aligning the query and reference sequences.

ANI calculation using pyani

```
average_nucleotide_identity.py -i InputOTUs -o ANIb_out -m ANIb -g -v --noclobber --nocompress --gformat jpg --scheduler multiprocessing --workers 16
```

Prerequisites (version): Python (v3.6.5) with Biopython, NumPy, Pandas, SciPy, Matplotlib, and Seaborn packages installed; R (v3.5.3); BLAST+ (v2.9.0).

Katana resources: nodes = 1, processors = 16, memory = 120 GB, wall time = 12 h

average_nucleotide_identity.py is a python script that is part of the pyani module of Python. It was used for calculating ANI using the options below:

- -i specified the input file name that contained a list of the OTU and reference FASTA file names.
- -o specified the output file name.
- -m ANIb used BLAST+ for aligning the sequences in the FASTA files.
- -g generated ANI heatmaps.
- --gformat jpg specified the ANI heatmap file format.
- --noclobber prevented the program from deleting any existing output files.
- --nocompress prevented the program from compressing or deleting the comparison outputs.
- --scheduler multiprocessing allowed the program to use multiple processors for parallel runs.
- --workers 16 specified the number of processors used for parallel runs.
- -v printed the command-line options to the log output file, for future reference.

2.2.5 Contig alignment and genome visualisation

2.2.5.1 Contig alignment

To assess how similar or dissimilar an OTU was from its closest related reference, the OTU contigs were aligned against the reference genome.

```
makeblastdb -in ReferenceGenome.fasta -dbtype nucl -out ReferenceGenomeDB
```

```
blastn -task megablast -query OTUcontigsSeqs.fasta -db ReferenceGenomeDB -out OTUalignment.sam -outfmt "17 SQ SR" -evalue 0.001

samtools view -bS OTUalignment.sam > OTUalignment.unsorted.bam

samtools sort -l 9 OTUalignment.unsorted.bam > OTUalignment.bam

samtools index OTUalignment.bam
```

Prerequisites (version): Blast+ (v2.6.0), Samtools (v1.5).

Katana resources: nodes = 1, processors = 16, memory = 96 GB, wall time = 12 h

makeblastdb is a module of BLAST+ that was used to prepare database index files. This needs to be run only once on a database, to create index files. All subsequent alignment runs used the indexed files. The -dbtype nucl option specified that the output database index file should be a nucleotide sequence file.

blastn is the alignment module of BLAST+ that aligned query nucleotide sequences (-query) against a reference nucleotide database index (-db). Other options include:

- -task megablast was used for finding highly similar sequences (closely related species).
- -outfmt "17 SQ SR" instructed the program to prepare a SAM output file (17) that included the sequence data (SQ) and displayed the reference sequence as the subject (SR).
- -evalue 0.001 set the maximum permissible e-value of an alignment to 0.001. Only alignments with e-value ≤0.001 were reported in the output file.

samtools view, **samtools sort**, and **samtools index** were used as described above in section 2.2.4.1.

2.2.5.2 Genome visualisation

The IGV GUI was used to visualise the contig alignment output files (BAM and BAI files), to assess how similar the OTU or MAG was to the genome of its closest related species. The reference genome FASTA file was uploaded to IGV (**Genomes** > **Load Genome from File** > select file > **Open**), along with the alignment BAM files (**File** > **Load from File** > select the BAM file(s) only > **Open**). IGV creates a reference genome index file (fasta.fai) in the folder where the reference file is stored. Importantly, the BAI files need to be stored in the same folder as the BAM files, although they are not directly uploaded to IGV, instead IGV automatically reads the BAI files in the folder (used for analyses in Chapters 4 and 5).

Apart from the BLAST and IGV approach, Mauve was used for both alignment and genome visualisation. The recommended 'Align with progressiveMauve' algorithm was

used for the alignment of two or more input FASTA files; each FASTA file can have multiple sequences (used for analyses in Chapters 4 and 5).

2.2.6 Assessing OTU phylogeny

The phylogenies of OTUs were assessed using MEGA. For multiple sequence alignment, the DNA or protein sequences were first uploaded to the software. DNA sequences were aligned using ClustalW algorithm, whereas protein sequences were aligned using MUSCLE algorithm. The alignment files were then used for generating phylogenetic trees using 500 or more bootstrap replicates, to assess the evolutionary relationship between an OTU and its closest related taxa/clades (used for analyses in Chapters 4 and 5).

2.2.7 Statistical analyses

PRIMER v7 software was used for the statistical analyses of Megahit-assembled metagenomes from Deep Lake surface samples collected in Dec 2006, Nov 2008, Dec 2013, Jun 2014, and Dec 2014; Club Lake surface samples collected in Nov 2014; and surface samples from Rauer Lakes 1, 3, 6, and 13 collected in Jan 2015 (Appendix A). This analysis was performed to assess variations in the relative abundances of OTUs identified in the metagenomes from hypersaline lakes in the Vestfold Hills (Deep Lake and Club Lake) and the Rauer Islands (Rauer Lakes 1, 3, 6, and 13). The seasonal variations in the relative abundances of OTUs were also assessed using some of the Deep Lake time-series samples. The OTU abundances were calculated using the LAST/MEGAN-LR output (section 2.2.2.2). For metagenomes from each lake and time period, the OTU abundances from all filter fractions were averaged, and the merged metagenome datasets were used for various PRIMER v7 analyses, including calculation of relative OTU abundances, alpha diversity and other diversity measures, and sample clustering based on relative OTU abundances.

To calculate relative OTU abundances using PRIMER v7, an Excel sheet containing OTU abundance data was prepared, with merged metagenomes as sample columns and OTUs as variable rows. Sample factors, such as lake and season name, were added to the Excel sheet, below the last OTU name, leaving one row empty between the species abundances and sample factors. The Excel sheet was uploaded to PRIMER v7 and the percentage relative OTU abundances were calculated using PRIMER v7. The diversity measures were calculated from the relative OTU abundances. Alpha diversity, species

richness, and species evenness were measured using Simson's index of diversity (1-λ′), total species-level OTUs in the merged metagenome, and Pielou's evenness index, respectively. All other measures were unchecked before producing the result. Sample clustering was also performed on relative OTU abundances. The relative abundances were square root transformed and used for preparation of a resemblance matrix of percentage similarities between the samples. The resemblance matrix data was utilised for creating a dendrogram using the UPGMA (unweighted pair group method with arithmetic mean) clustering method.

2.3 Method test results and discussion

To improve the preliminary metagenome analysis pipeline (Cavlab pipeline v1.2), a number of software and methods were tested on metagenomes from meromictic lakes (Ace Lake, Organic Lake) and hypersaline lakes (Deep Lake, Club Lake, Rauer Lakes), to assess their suitability for the analysis of Antarctic metagenomes. Some methods were tested on metagenomes from both types of lakes (meromictic as well as hypersaline), to determine if they were robust enough to produce results from very different lake systems. In order to verify the reliability of the software or methods, specifically for taxonomic classification and OTU abundance estimation, their outputs were compared to previously reported data from Ace Lake (Rankin et al, 1997; Rankin, 1998; Powell et al, 2005; Ng et al, 2010; Lauro et al, 2011), Deep Lake (DeMaere et al, 2013), and Organic Lake (Bowman et al, 2000a; Yau et al, 2013).

Ace Lake is a stratified lake with an upper oxic zone, an oxyline/halocline, and a lower anoxic zone. A high abundance of a *Synechococcus* has been reported in Ace Lake upper oxic zone, just above the oxycline, using a combination of flow cytometry techniques and *16S rRNA* gene sequence comparisons (Rankin et al, 1997; Rankin, 1998; Powell et al, 2005). This observation was supported by additional *16S rRNA* gene-based and metagenome read-based analyses of biodiversity in the Ace Lake upper oxic zone (Lauro et al, 2011). Moreover, the high abundance of a GSB, closely related to *Prosthecochloris vibrioformis* DSM 265 (now *C. phaeovibrioides* DSM 265), has been reported in Ace Lake oxycline (Ng et al, 2010). The researchers assembled the draft genome (nine scaffolds) of the GSB from a 0.1 μm-filter Ace Lake oxyline metagenome and found that 77% of all metagenome reads belonged to this microbe.

This was supported by a similar report of a clonal population of a *Chlorobium* in Ace Lake oxycline, based on the high score matches of *16S rRNA* gene fragments (e-value ≤ 10^{-5}) and high identity matches of metagenome reads (>60% identity; e-value ≤ 10^{-5}) (Lauro et al, 2011). Deep Lake, a hypersaline oxic lake, is abundant in haloarchaea, and is marked by the intergenera exchange of long (~35 kb), high identity (100%) DNA sequences (DeMaere et al, 2013). Among the haloarchaea, three were reported to be highly abundant — 44% *Halohasta litchfieldiae*, 18% halophilic archaeon DL31, and 10% *Halorubrum lacusprofundi* in Deep Lake samples. The researchers used a combination of *16S rRNA* gene sequence comparison for taxonomy identification and stringent fragment recruitment (FR) of metagenome reads (>98% match identity and >90% alignment fraction) for abundance calculation. Organic lake, a stratified, hypersaline lake, was reported to mainly support populations of heterotrophic bacteria, such as *Psychroflexus*, *Marinobacter, Halomonas*, and *Roseovarius*, based on *16S rRNA* gene sequence comparison (Bowman et al, 2000a; Yau et al, 2013).

Overall, the taxonomic composition of three lakes have been verified by multiple research groups using *16S rRNA* gene sequences, which is the most commonly used marker gene. Additionally, some of the researchers have used either flow cytometry or stringent FR of reads for OTU abundance calculations. Therefore, these previously reported taxonomies and their abundances were used as references for the validation of the output of software/methods tested on Antarctic metagenomes.

2.3.1 Metagenome taxonomic diversity and OTU abundance

Software tested for the analysis of metagenome taxonomic diversity included PhyloSift, MetaPhlAn2, and Kaiju using filtered read sequences (section 2.3.1.1 below), DIAMOND and MEGAN6 using protein sequences (section 2.3.1.2 below), and LAST and MEGAN-LR using contig sequences (section 2.3.1.3 below). Apart from these, the protein taxonomies in the Phylodist file were used to generate contig taxonomies, which were used for assessing metagenome taxonomic diversity and for calculating OTU abundances (section 2.3.1.4 below).

2.3.1.1 Read-based taxonomic diversity analysis

Read-based taxonomic classification methods are often useful, because they allow for easy and straight forward calculation of OTU abundances. A taxon abundance can be calculated by simply counting the number of reads assigned to it, and its relative

abundance can be calculated by dividing the abundance by the total number of reads in the metagenome.

PhyloSift

PhyloSift was a part of Cavlab pipeline v1.2 and was used for read-based taxonomic classification and abundance estimation (Appendix B). PhyloSift database comprises of a core and an extended set of marker genes, and for metagenomic analysis it is preferable to use the extended set of marker genes to capture as much taxonomic information as possible. However, one of the limitations of running PhyloSift on metagenomic data was the size of the dataset; PhyloSift runs on metagenomes using the extended set of marker genes did not complete within the maximum available wall time (200 hours) on the UNSW Katana computer cluster. Apart from this, each PhyloSift run automatically downloads the latest version of the online PhyloSift database, as part of the program run. However, the updates in the online PhyloSift database interrupted any on-going PhyloSift runs and caused errors. To avoid this issue, the software, along with its latest databases, were downloaded to the Katana scratch node and the PhyloSift program was provided paths to the offline versions of the PhyloSift databases (Cavlab pipeline v1.3a). Despite fixing the PhyloSift database issue, the sheer size of the metagenome datasets prevented successful runs and the software was removed from the Cavlab pipeline in v2.0.

MetaPhlAn2

MetaPhlAn2, a software for profiling the taxonomic composition of metagenomes, was tested for read-based taxonomic diversity analysis and relative abundance estimation and was added to Cavlab pipeline in v3.0. MetaPhlAn2 database is composed of clade-specific markers, which allows for more accurate taxonomic assignment of reads. However, this limits the analysis to systems with well-characterised taxonomic diversity, as was observed during the analysis of Antarctic metagenomes that tend to have uncommon and some unique taxa, most of which have never been cultured. MetaPhlAn2 runs on a few Ace Lake metagenomes, to assess the viability of its read-based taxonomic diversity analysis of Antarctic metagenomes, showed that its database (DBv20) was not useful for the analysis of Antarctic metagenomes — less than 0.4% of the total filtered reads in any metagenome were assigned a taxonomy (Table 2.3).

Table 2.3 Read-based taxonomic diversity analysis of some Ace Lake metagenomes using MetaPhlAn2. [A] The samples in the first column refer to the metagenomes from Ace Lake (Appendix A). [B] The number of reads that were assigned a taxonomy using MetaPhlAn2 database v20 (DBv20) are mentioned in the last column. The percentages were calculated by dividing the number of assigned reads by the total number of reads in a metagenome.

Metagenome (collection date; depth; filter fraction)[A]	Number of OTUs identified	Total number of filtered reads	Number of read assignments (% read assignments)[B]
19/11/2008; 5 m; 0.8 µm	16	58,374,702	180,050 (0.31%)
21/11/2008; 11.8 m; 0.8 µm	17	70,074,842	209,257 (0.30%)
21/11/2008; 12.8 m; 0.8 µm	3	53,708,884	191,171 (0.36%)
21/11/2008; 14.1 m; 3 µm	6	59,786,200	7,471 (0.01%)
21/11/2008; 14.1 m; 0.8 µm	6	60,749,386	7741 (0.01%)
21/11/2008; 18 m; 0.8 µm	5	71,060,470	38,230 (0.05%)
23/11/2008; 23 m; 0.8 µm	8	59,324,648	7,459 (0.01%)

To improve the MetaPhlAn2 database (DBv20) for use with Antarctic metagenomes, clade-specific markers associated with some of the more known, abundant taxa in Ace Lake and Deep Lake were added to the database (Appendix E). Testing the manually updated database on one of the Ace Lake metagenomes (Nov 2008_5 m_0.8 µm-filter) showed slight, but not significant, improvement in the taxonomic assignment of the filtered reads — % read assignments: 0.31% using DBv20 vs 0.32% using updated database. Therefore, the MetaPhlAn2 database needs to be updated with markers for Antarctic and other polar and cold environment species. However, this can be an exhaustive and time-intensive task. Before adding a marker to the database, which needs to be done manually (section 2.2.2.3), it must first be matched and scored against all other markers already in the database, which is important for proper abundance estimation. This was not done for the markers listed in Appendix E, since the purpose of adding those markers was to first assess the improvement in read taxonomic assignment, which is unaffected by the marker score. Considering the MetaPhlAn2 database issues, the software was removed from the Cavlab pipeline in v3.1.

Kaiju

As the trials of PhyloSift and MetaPhlAn2 with Antarctic metagenomes were unsuccessful, a simpler, yet efficient, taxonomic classification tool, namely Kaiju, was

tested for the read-based taxonomic diversity analysis of the Antarctic metagenomes. The NCBI-nr protein database was used as reference for Kaiju runs, since it is more exhaustive than marker-based databases, although less specific. Kaiju runs were performed on a few metagenomes from Ace Lake and Deep Lake, to assess the viability and robustness of its taxonomic classification of Antarctic metagenome reads (Table 2.4). For this, the greedy-5 algorithm of Kaiju was used, as it is better at taxonomic classification of environmental samples than the Kaiju MEM algorithm (Menzel et al, 2016). Additionally, its overall precision and sensitivity in genus and phylum-level classification of paired-end Illumina reads (250 nt) is better than that of the other Kaiju algorithms (Menzel et al, 2016).

Table 2.4 Read-based taxonomic diversity analysis of some Ace Lake and Deep Lake metagenomes using Kaiju. The relative abundances of the taxa were calculated by Kaiju — total reads assigned to a taxon divided by the total reads in the metagenome. [A] The samples in the first column refer to the metagenomes from Ace Lake and Deep Lake (Appendix A). [B] The reads that were not assigned to any taxa were referred to as 'unclassified' or 'unassigned' in Kaiju output.

Metagenome (collection date; depth; filter fraction)[A]	OTUs with relative abundance ≥1%	Relative abundance of Viruses	Number of OTUs with relative abundance <1%	Number of unclassified/unassigned reads[B]
Ace Lake				
21/11/2008; 11.8 m; 0.8 µm	*Candidatus* Pelagibacter ubique (3%) *Candidatus* Aquiluna sp. IMCC13023 (3%) Pelagibacteraceae bacterium BACL20 MAG-120920-bin64 (2%) Microbacteriaceae bacterium BACL28 MAG-120531-bin53 (1%)	16%	23,859	52%
21/11/2008; 12.8 m; 0.1 µm	*Chlorobium phaeovibrioides* (50%) *Pelodictyon luteolum* (4%)	0.2%	17,025	36%

	Chlorobium phaeobacteroides (2%)			
21/11/2008; 14.1 m; 0.1 µm	None. Highest relative OTU abundance was 0.3% (archaeon BMS3Abin17)	0.8%	25,552	79%
23/11/2008; 23 m; 0.1 µm	None. Highest relative OTU abundance was 0.2% (archaeon BMS3Abin17)	1%	25,568	82%
25/11/2013; 12.5 m; 0.1 µm	*Chlorobium phaeovibrioides* (19%) *Pelodictyon luteolum* (2%)	0.8%	23,090	57%
Deep Lake				
1/12/2006; 0 m; 0.8 µm	halophilic archaeon DL31 (16%) *Halorubrum lacusprofundi* (7%) *Natrinema* sp. CBA1119 (1%)	0.8%	21,140	58%
24/11/2014; 0 m; <0.1 µm	*Halorubrum lacusprofundi* (6%) halophilic archaeon DL31 (4%)	0.7%	7,554	80%

Kaiju output showed that taxonomic diversity varied with depth in Ace Lake, and that in Deep Lake, haloarchaea were highly abundant; these findings were consistent with previous observations from Ace Lake and Deep Lake (Lauro et al, 2011; DeMaere et al, 2013). However, the OTU abundances calculated by Kaiju in metagenomes from the two lake systems were very different from previously reported abundances. For example, in samples from Deep Lake, the relative abundance of *Hht. litchfieldiae* has been reported to be as high as 44% (DeMaere et al, 2013), but Kaiju calculated its relative abundance to only 0.8%. As horizontal transfer of long, high identity regions (~35 kb in length) has been reported among the haloarchaea in Deep Lake (DeMaere et al, 2013) and because Kaiju algorithm is a k-mer based method that relies on exact matches of short sequences, it is possible that most of the reads matched multiple haloarchaea genomes and were assigned to no taxa. Another issue with the Kaiju output was the high percentage of total reads in a metagenome that could not be assigned a taxonomy and were reported as unassigned or unclassified reads (36–82%). Additionally, Kaiju v1.6.2 used for this analysis did not calculate relative abundances of

viruses at lower taxa levels (family, genus, species), and rather aggregated the abundances of all viruses under 'Viruses' taxon. Therefore, based on these issues Kaiju was not added to the Cavlab pipeline for read-based taxonomic classification and abundance analysis.

Read-based taxonomy is based on the alignment of very small sequences and has a higher probability of being wrongly assigned to a taxon than a protein or a contig sequence. Additionally, most of the reads in the analyses above could not be assigned a taxonomy and were reported as unassigned or unclassified reads. Therefore, protein and contig taxonomic classification methods were also explored alongside read-based methods.

2.3.1.2 Protein-based taxonomic diversity analysis

DIAMOND and MEGAN6 component of Cavlab pipeline v2.0 was tested for the taxonomic classification of protein sequences in Megahit-assembled metagenomes from Ace Lake 2008 and Deep Lake 2013–2015 time-series metagenomes (Appendix A). The outputs of the DIAMOND/MEGAN6 runs showed that the method was reliable at higher taxa levels, as it corroborated previously observed data (Lauro et a, 2011; DeMaere et al, 2013). It showed high abundance of bacteria throughout the Ace Lake, with a few algae in the upper, oxic zone and some archaea in the lower, anoxic zone (Figure 2.2), and high abundance of haloarchaea in Deep Lake, irrespective of change in season (Figure 2.3). The method was also robust, because it worked on metagenomes from very different lake systems. However, DIAMOND/MEGAN6 method was not able to effectively assign proteins to lower taxa levels, such as genus and species. For example, in the 0.1 μm-filter metagenome from Ace Lake 2008 oxycline, 85% of the total proteins in the metagenome were classified to Bacteria domain (Figure 2.2), but only 13% of the total proteins were classified to *C. phaeovibrioides*. This was inconsistent with previous reports of a very high abundance of a *Chlorobium* closely related to *C. phaeovibrioides* in the Ace Lake oxycline (Ng et al, 2010; Lauro et al, 2011).

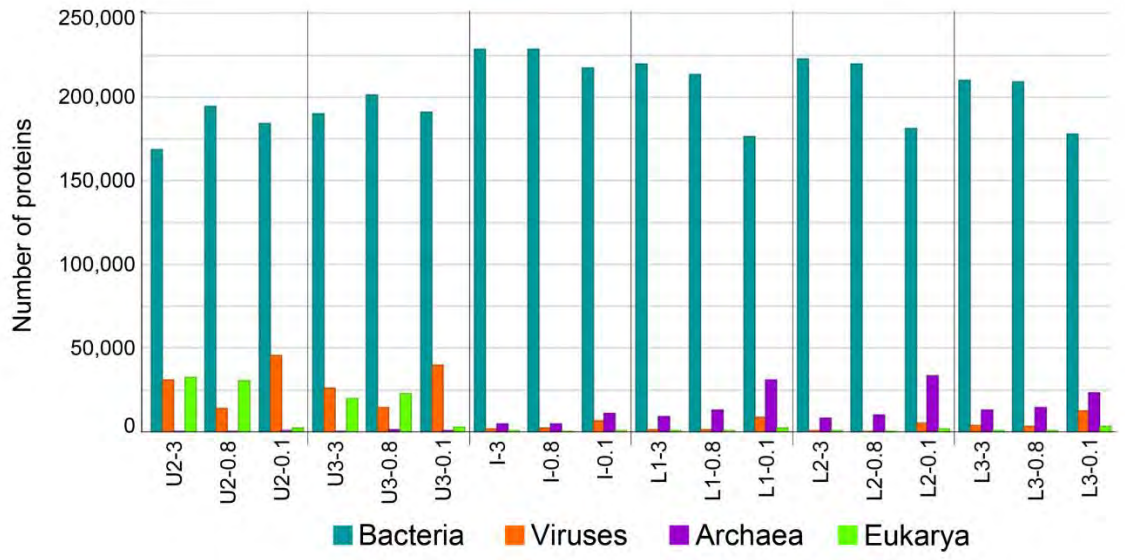


Figure 2.2 DIAMOND and MEGAN6 output for Megahit-assembled Ace Lake 2008 metagenomes. The bar chart shows the number of proteins assigned to Bacteria (dark green), Viruses (orange), Archaea (purple), and Eukarya (green) in samples collected from six Ace Lake depths (U2, U3, I, L1, L2, L3) on three filters (3, 0.8, 0.1) in Nov 2008. Depths: U2, upper 2; U3, upper 3; I, interface; L1, lower 1; L2, lower 2; L3, lower 3. Filter fractions: 3, 3–20 µm; 0.8, 0.8–3 µm; 0.1, 0.1–0.8 µm.

This issue with species-level classification was also observed in the DIAMOND/MEGAN6 output of Deep Lake 2013–2015 time-series metagenome analysis, where more proteins were assigned to *Halobacterium* sp. DL1 than to *Hrr. lacusprofundi*, although the dominant species was *Hht. litchfieldiae* along with DL31. This was inconsistent with previous observations of haloarchaea species abundance in Deep Lake (44% *Hht litchfieldiae,* 18% DL31, 10% *Hrr. lacusprofundi*, and 0.3% DL1; DeMaere et al, 2013). Similar to what was observed in Deep Lake Kaiju output (Table 2.4), it is possible that most Deep Lake metagenome proteins had matches to proteins from multiple haloarchaea genomes, due to known HGT among the Deep Lake haloarchaea (DeMaere et al, 2013). Hence, MEGAN6 could not confidently assign the proteins to species-level, thereby classifying them to higher taxa levels instead. Another possibility was that the deviations observed in the DIAMOND/MEGAN6 outputs, compared to previously reported observations, reflected actual changes in the systems over time. If so, then all taxonomic diversity analyses performed on these metagenomes should yield similar results. However, the contig-based taxonomy analysis of the Deep Lake metagenomes using LAST/MEGAN-LR approach did not support these observations from the DIAMOND/MEGAN6 output (described below in section

2.3.1.3). Therefore, DIAMOND/MEGAN6 output from the Cavlab pipeline was not used for assessing the taxonomic composition of Antarctic metagenomes, and was limited to COG-based functional potential analysis (described below in section 2.3.3.1).

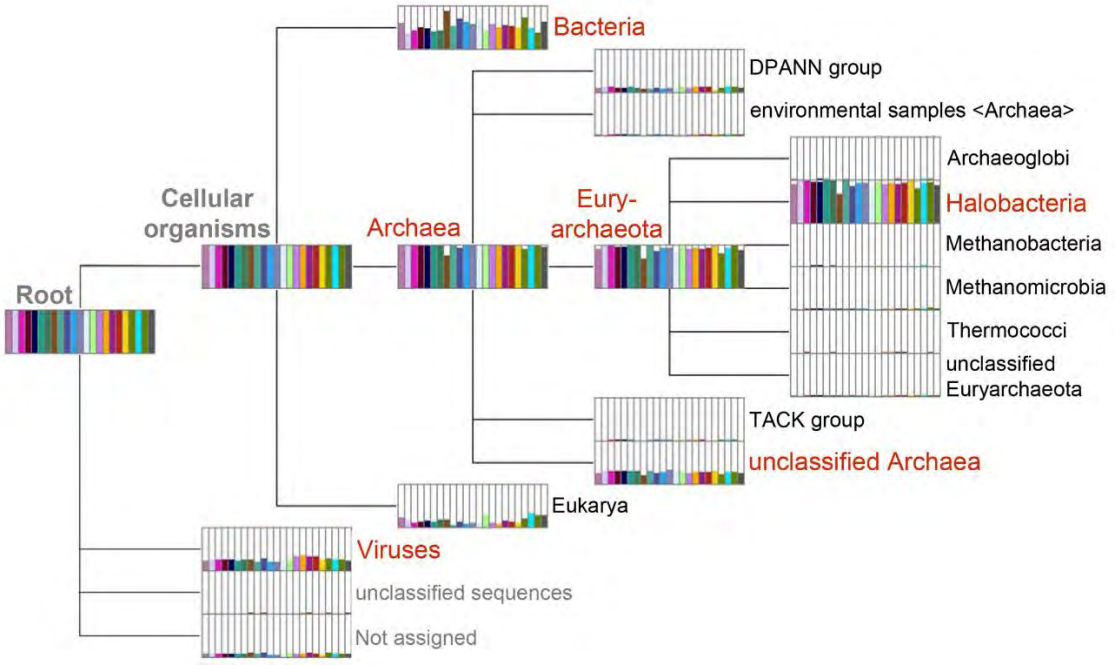


Figure 2.3 DIAMOND and MEGAN6 output for Megahit-assembled Deep Lake 2013–2015 time-series metagenomes. The phylogenetic tree shows the higher-level taxa (domain to class) to which most of the metagenome proteins were assigned (red-highlighted). For each taxon, from left to right, the number of proteins assigned to it are shown in metagenomes from 3–20 µm-filter collected in Dec 2013 (■), Feb 2014 (■), Jun 2014 (■), Aug 2014 (■), Nov 2014 (■), Dec 2014 (■), and Jan 2015 (■); followed by metagenomes from 0.8–3 µm-filter collected in Dec 2013 (■), Feb 2014 (■), Jun 2014 (■), Aug 2014 (■), Nov 2014 (■), Dec 2014 (■), and Jan 2015 (■); metagenomes from 0.1–0.8 µm-filter collected in Dec 2013 (■), Feb 2014 (■), Jun 2014 (■), Aug 2014 (■), Nov 2014 (■), and Dec 2014 (■); and metagenomes from <0.1 µm-filter collected in Nov 2014 (■), Dec 2014 (■), and Jan 2015 (■). DPANN group, *Diapherotrites*, *Parvarchaeota*, *Aenigmarchaeota*, *Nanoarchaeota* and *Nanohaloarchaeota* group; TACK group, *Thaumarchaeota*, *Aigarchaeota*, *Crenarchaeota*, and *Korarchaeota*.

As a result, contig-based taxonomic diversity analysis of Antarctic metagenomes was considered, because contigs have a lower probability of incorrect taxonomic assignment than proteins, considering their longer lengths. This would also reduce the chance of any bias that might be caused due to HGT.

2.3.1.3 Contig-based taxonomic diversity analysis

To assess the robustness of the LAST/MEGAN-LR method for contig-based taxonomic diversity analysis, the method was tested on various Megahit-assembled metagenomes from Ace Lake 2006 and 2008, Organic Lake 2006, Deep Lake 2006, 2008, and 2013–2015, Club Lake, and Rauer Lakes 1, 3, 6, 11, and 13 (Figures 2.4 and 2.5). In MEGAN6 GUI, the taxa names along with the names of the contigs assigned to them were exported to a file and this exported data, in concert with the data in the metagenome contig coverage files, containing contig IDs and their read depths, was used for calculating relative OTU abundances (section 2.2.2.2).

LAST/MEGAN-LR output of the Ace Lake meromictic system showed that the Upper zone mainly harboured phototrophic bacteria and algal viruses (53% and 35% peak relative abundances, respectively), whereas the Interface was dominated by a high relative abundance of *Chlorobium* (91% peak relative abundance), along with members of *Deltaproteobacteria* (59% peak relative abundance), which were also abundant in the Lower zone (Figure 2.4); these corroborated previous findings (Bowman et al, 2000b; Ng et al, 2010; Lauro et al, 2011). Members of candidate phyla, such as *Cloacimonetes* and *Atribacteria* (18% and 10% peak relative abundances, respectively), were also observed in Ace Lake Lower anoxic zone. Among the phototrophs, a *Synechococcus* was one of the predominant cyanobacteria in Ace Lake oxic zone (39% peak relative abundance) just above the oxycline, which has also been reported before (Rankin et al, 1997; Rankin, 1998; Powell et al, 2005; Lauro et al, 2011). Therefore, the LAST/MEGAN-LR runs on Ace Lake were successful, and the method was inferred to be reliable for Ace Lake analysis.

Apart from some members of *Alphaproteobacteria*, *Actinobacteria*, and *Bacteroidetes*, and some algal viruses and bacteriophages that were common to Organic Lake and Ace Lake, the surface metagenomes from Organic Lake had a distinct population of members of *Flavobacteriia* (*Bacteroidetes*) (Figure 2.4). *Psychroflexus*, members of *Flavobacteriia* class, were highly abundant in 3–20 and 0.8–3 μm-filter fractions from Organic Lake surface (average of relative abundance in 3–20 and 0.8–3 μm filter fractions: 25% *Psychroflexus torquis* and 14% *Psychroflexus gondwanensis*), similar to previous reports (Bowman et al, 2000a; Yau et al, 2013). Therefore, the LAST/MEGAN-LR runs on Organic Lake were successful, since the output taxonomic diversity conformed to reference data.

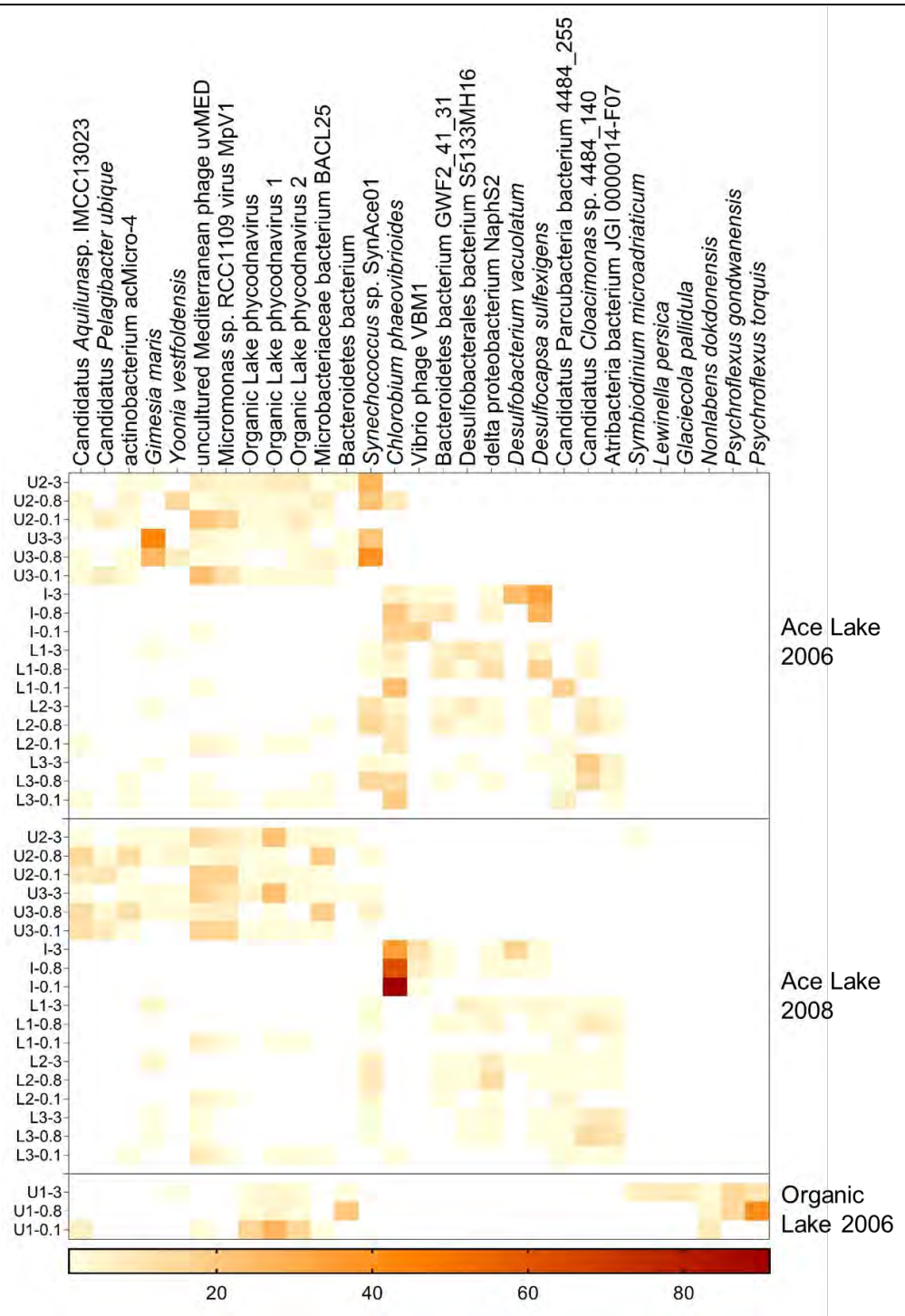


Figure 2.4 LAST and MEGAN-LR output for Megahit-assembled metagenomes from Antarctic meromictic lake systems — Ace Lake and Organic Lake. The heat map represents the relative abundance of OTUs identified in Megahit-assembled metagenomes from three filter fractions (3, 0.8, 0.1) and six depths (U2, U3, I, L1, L2, L3) of Ace Lake, samples collected in 2006 and 2008, as well as Organic Lake surface (U1), samples collected in 2006. The gradient bar represents the percentage relative OTU abundances. The OTUs shown here are some of the most abundant OTUs observed in Ace Lake and Organic Lake, with relative abundance >5% in at least one metagenome. The white portions of the heat map represent relative abundances

<1%. Depths: U1, upper 1; U2, upper 2; U3, upper 3; I, interface; L1, lower 1; L2, lower 2; L3, lower 3. Filter fractions: 3, 3–20 μm; 0.8, 0.8–3 μm; 0.1, 0.1–0.8 μm.

LAST/MEGAN-LR output of the analysis of Megahit-assembled metagenomes from Deep Lake hypersaline system showed dominant presence of haloarchaea in the surface waters (Figure 2.5). *Hht. litchfieldiae* (64% peak relative abundance) was the most abundant haloarchaea species in Deep Lake, followed by DL31 and *Hrr. lacusprofundi* (43% and 17% peak relative abundances, respectively), similar to previous observations in Deep Lake samples from 2006 (44% *Hht litchfieldiae,* 18% DL31, and 10% *Hrr. lacusprofundi*; DeMaere et al, 2013). The relative abundances of the haloarchaea were unaffected by the change in season according to the LAST/MEGAN-LR output (Figure 2.5), which was similar to the observations from DIAMOND/MEGAN6 output of Deep Lake (Figure 2.3). Further comparison of LAST/MEGAN-LR with DIAMOND/MEGAN6 outputs of Deep Lake showed that the contig-based taxonomic diversity analysis was more reliable than protein-based analysis, which did not corroborate previous findings. Therefore, the LAST/MEGAN-LR runs on Deep Lake were successful and reliable.

Club Lake, which is also a hypersaline system, showed a similar taxonomic diversity of haloarchaea as was observed in Deep Lake: *Hht. litchfieldiae* (51% peak relative abundance) being the most abundant haloarchaea species, followed by DL31 and *Hrr. lacusprofundi* (30% and 15% peak relative abundances, respectively) (Figure 2.5). Both hypersaline systems showed presence of a similar population of haloarchaea viruses (peak relative abundance: 15% in Deep Lake and 7% in Club Lake), but not halophilic bacteria (Figure 2.5). All halophilic bacteria identified in Club Lake had relative abundance <1% in all metagenomes from the lake, supporting the dominance of haloarchaea in the system. As Deep Lake and Club are both hypersaline systems and lie in close proximity (less than 2 km apart), it was interesting that their taxonomic composition was so similar, and it raised questions about the effect of salinity and distance between the lakes on their taxonomic composition. As the taxonomic composition of Club Lake had not been analysed prior to these LAST/MEGAN-LR runs, there was no way to assess the reliability of the output. However, the LAST/MEGAN-LR output for Club Lake metagenomes was considered to be successful, because the method had reliable outputs from Deep Lake, which is another hypersaline system in close proximity to Club Lake.

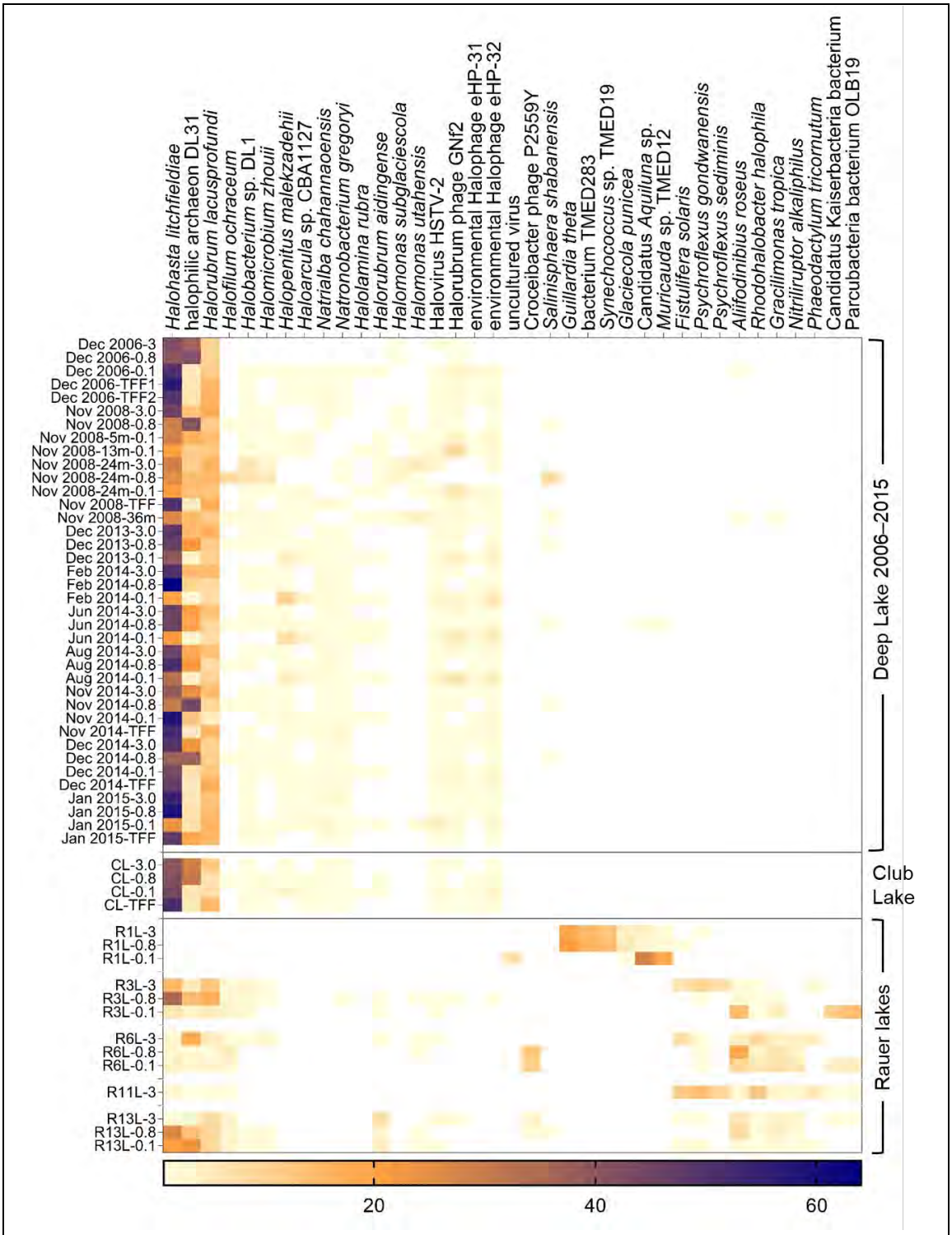


Figure 2.5 LAST and MEGAN-LR output for Megahit-assembled metagenomes from Antarctic hypersaline lake systems — Deep Lake, Club Lake, and Rauer Island lakes. The heat map shows the relative abundance of OTUs in Megahit-assembled metagenomes from three filter fractions (3, 0.8, 0.1) from the surface of Deep Lake (samples from 2006, 2008, and 2013–2015), from four depths of Deep Lake (2008), and from the surface of Club Lake (2014) and Rauer Lakes 1, 3, 6, 11, and 13 (2015). The gradient bar represents the percentage relative OTU abundances. The OTUs shown here are some of the most abundant OTUs observed in

Deep Lake, Club Lake, and Rauer Island lakes, with relative abundance >2% in at least one metagenome. The white portions of the heat map represent relative abundances <1%. Lakes: CL, Club Lake; R1L, Rauer 1 Lake; R3L, Rauer 3 Lake; R6L, Rauer 6 Lake; R11L, Rauer 11 Lake; R13, Rauer 13 Lake. Filter fractions: 3, 3–20 μm; 0.8, 0.8–3 μm; 0.1, 0.1–0.8 μm; TFF (tangential flow filtration), <0.1 μm. *Candidatus* Kaiserbacteria bacterium, *Candidatus* Kaiserbacteria bacterium RIFOXYD1_FULL_42_15; *Candidatus* Aquiluna sp.; *Candidatus* Aquiluna sp. UB-MaderosW2red.

Lastly, the LAST/MEGAN-LR method was tested on metagenomes from hypersaline lake systems in the Rauer Islands, Antarctica (Figure 2.5). Unlike Rauer 1 Lake, which mainly harboured a bacterial population, the Rauer Lakes 3, 6, 11, and 13 showed high abundance of haloarchaea that were also present in Deep Lake and Club Lake in the Vestfold Hills, albeit their abundances were not as high as in Vestfold Hill lakes. However, unlike Deep Lake and Club lake, Rauer Lakes 3, 6, 11, and 13 supported a larger bacterial population, including members of *Bacteroidetes*, *Balneolaeota*, *Actinobacteria*, and candidate phyla, such as *Parcubacteria* and *Kaiserbacteria*. The taxonomic diversity of Rauer 1 Lake was almost completely different from that observed in Rauer Lakes 3, 6, 11, and 13, except for a *Flavobacteriia* (*Psychroflexus gondwanensis*) and a diatom (*Fistulifera solaris*) that were present in all Rauer Island lakes tested (Figure 2.5). A pigment-based high-performance liquid chromatography study on Rauer Island lake samples identified presence of certain eukarya and cyanobacteria in some of the lakes (Hodgson et al, 2001). Other than this, there were no previous studies on the taxonomic composition of Rauer Island lakes at the time. Therefore, due to the lack of taxonomic data from Rauer Lakes 1, 3, 6, 11, and 13, the reliability of the LAST/MEGAN-LR output from Rauer Island lake metagenomes could not be verified. However, the LAST/MEGAN-LR method was considered to work on Rauer Island lake metagenomes, as with Club Lake metagenomes, since it had worked reliably on metagenomes from Deep Lake, which is also a hypersaline lake system.

MEGAN-LR was specifically developed for the taxonomic classification of contigs and long reads and uses the 'interval-union LCA' algorithm, which allows for stringent and reliable mapping of contigs to taxa (Chapter 1 section 1.4.1.2). In this algorithm, a contig is assigned to a taxon only if the proteins from the taxon cover 80% or more of the contig sequence, considering only the protein alignments with a significant bit score (Huson et al, 2018). Based on the outputs of LAST/MEGAN-LR runs on metagenomes

from Ace Lake, Deep Lake, Club Lake, Organic Lake, and Rauer Island lakes 1, 3, 6, 11, and 13, the method was found to be reliable, since the output data from runs on some of these lake systems corroborated previously reported data, and was inferred to be robust, because the method worked on metagenomes from very different lake systems — meromictic, hypersaline, or both. Consequently, LAST/MEGAN-LR was added to the Cavlab pipeline v3.0 for taxonomic diversity analysis and OTU abundance estimation.

2.3.1.4 Changes in metagenome assembly method and its impact on the development of Cavlab pipeline

All Megahit-assembled metagenomes from the Antarctic samples were re-assembled by JGI using metaSPAdes assembler, as part of changes to their assembly pipeline. All samples sent to JGI for sequencing after the change in their assembly pipeline were assembled using the metaSPAdes assembler. A comparison of contig statistics of Megahit- and Spades-assembled metagenomes showed that Spades assemblies had more contigs of longer sequence length, but their overall assembly size was smaller than that of Megahit assemblies (Table 2.5).

Table 2.5 Comparison of contig statistics of Megahit- *vs* Spades-assembled metagenomes from some Ace Lake and Deep Lake samples. The contig statistics mentioned in the table were taken from the data generated by JGI after contig assembly using Megahit (M) or SPAdes (S) assembler. [A] These metagenomes were used to assess the viability of Kaiju, LAST/MEGAN-LR, and Phylodist file-based methods for taxonomic diversity analysis in Figure 2.6. LAST/MEGAN-LR was tested on contigs from these Megahit- and Spades-assembled metagenomes. [B] The percentage genome in contigs >50 kb length was calculated by dividing the sum of length of contigs >50 kb by the sum of length of all contigs in the metagenome (total contig length). [C] N50 and L50 statistics are indicative of assembly quality. N50 is the length of the shortest contig at 50% of the total length of all contigs in a metagenome, whereas L50 is the smallest number of contigs whose total length contributes to at least 50% of the total contig length of the metagenome. For example, if a metagenome has 7 contigs of lengths 10, 23, 34, 44, 56, 61, and 78 bp, then the total length of all contigs in the metagenome would be 306 bp and 50% of this total length would be 153 bp. To calculate N50 and L50, the contigs first need to be arranged in the order of decreasing contig lengths (78 bp, 61 bp,…). In this case, N50 would be 56 (78 + 61 = 139 + **56** = 195 > 153) and L50 would be 3 contigs (78, 61, and 56 bp), as the total length of the contigs ≥56 bp contributes to at least 50% of the total contig length of

the metagenome. [C] This metagenome was assembled after the changes to JGI's assembly pipeline, and was not assembled using Megahit. NA, not applicable.

Lake system		Ace Lake		**Deep Lake**	
Sample (collection date; depth; filter fraction)[A]		21/11/2008; 12.8 m; 0.1 µm	25/11/2013; 13.5 m; 0.1 µm[C]	1/12/2006; 0 m; 0.1 µm	24/11/2014; 0 m; <0.1 µm
Contigs >50 kb length	M	35	NA	33	27
	S	56	71	57	64
Contigs <1 kb length (%)	M	311,340 (93%)	NA	215,310 (93%)	126,639 (93%)
	S	119,695 (89%)	138,631 (81%)	68,383 (87%)	45,085 (87%)
Total contigs	M	335,960	NA	230,986	136,229
	S	134,843	170,162	78,454	51,989
Assembly size (total contig length)	M	167 Mb	NA	120 Mb	86 Mb
	S	93 Mb	154 Mb	69 Mb	51 Mb
% Genome in contigs >50 kb[B]	M	2%	NA	2%	3%
	S	8%	5%	8%	12%
Contig N50[C]	M	51,155 bp	NA	31,253 bp	17,876 bp
	S	14,478 bp	19,645 bp	4,628 bp	2,296 bp
Contig L50[C]	M	612	NA	626	684
	S	1,028	1,388	1,873	2,984

In view of the better contig length statistics of the Spades assemblies and for the purpose of consistency, the Spades-assembled metagenomes were considered for taxonomic and functional potential analyses of Ace Lake, and the software that had been finalised for use with the Megahit assemblies were re-tested on the Spades assemblies (Figure 2.6).

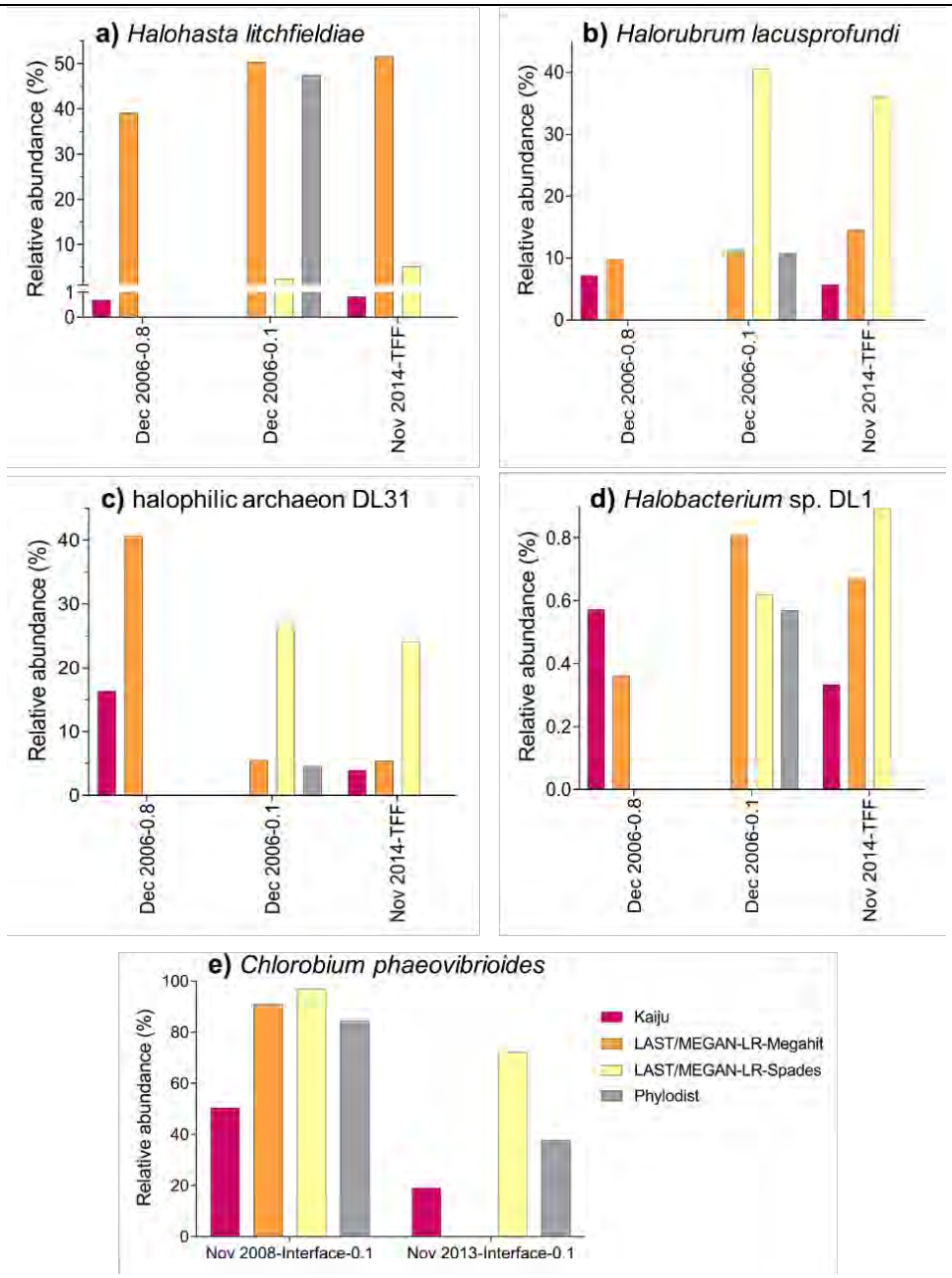


Figure 2.6 Comparison of taxonomic classification methods used for relative OTU abundance estimation in Antarctic metagenomes. The bar charts show relative abundances of (**a**) *Halohasta litchfieldiae*, (**b**) *Halorubrum lacusprofundi*, (**c**) halophilic archaeon DL31, and (**d**) *Halobacterium* sp. DL1 in three metagenomes from Deep Lake surface and (**e**) *Chlorobium phaeovibrioides* in two metagenomes from Ace Lake oxycline (Interface). Kaiju (red bars) used filtered reads for taxonomic classification and abundance estimation. LAST/MEGAN-LR method was used for taxonomic classification of contigs from both Megahit-assembled metagenomes (LAST/MEGAN-LR-Megahit, orange bars) and Spades-assembled metagenomes (LAST/MEGAN-LR-Spades, yellow bars), and a python script was used to calculate the relative OTUs abundances (section 2.2.2.2). The IMG protein taxonomy data (Phylodist, grey bars) was used to assign taxonomy to Spades-assembled contigs and relative OTU abundances were

calculated using a python script (Appendix C). Missing data bars indicate that some methods were not tested on some of the metagenomes. Filter fractions: 0.8, 0.8–3 μm; 0.1, 0.1–0.8 μm; TFF (tangential flow filtration), <0.1 μm.

LAST/MEGAN-LR runs and relative abundance calculations on the Spades-assembled metagenomes from the different lake systems (hypersaline Deep Lake or meromictic Ace Lake) did not always yield results that were consistent with previously reported findings (Figure 2.6). A comparison of the relative abundances of certain key species from Deep Lake showed that LAST/MEGAN-LR method did not work well on the Deep Lake Spades-assembled metagenomes (Figure 2.6). For example, in the Dec 2006 0.1 μm Deep Lake surface sample, the relative abundance of *Hht. litchfieldiae* in Spades-assembled metagenome was much lower than that in the Megahit-assembled metagenome (Spades 2% vs Megahit 50%), whereas the relative abundances of *Hrr. lacusprofundi* (Spades 41% vs Megahit 11%) and DL31 (Spades 27% vs Megahit 5%) were much higher, and did not match previously reported findings (44% *Hht. litchfieldiae,* 18% DL31, and 10% *Hrr. lacusprofundi*; DeMaere et al, 2013). Contrarily, the relative abundance of *C. phaeovibrioides*, closest related species to the key microbe in Ace Lake oxycline, calculated using LAST/MEGAN-LR output, was comparable in Megahit- (91%) and Spades-assembled (97%) metagenomes (Figure 2.6). This difference in relative OTU abundances in Megahit- *vs* Spades-assembled metagenomes was probably because MEGAN-LR could not reliably assign the Spades-assembled long contigs to species-level taxa. The algorithm used by MEGAN-LR for the taxonomic assignment of contigs is very stringent — a contig can be assigned to a taxon only if the proteins from the taxon match at least 80% of the contig length, with each protein alignment having a significant bit score, i.e., a bit score must be within 10% of the best bit score observed for that part of the contig sequence (Huson et al, 2018). Additionally, the contig would be assigned to the lowest common ancestor in cases where multiple taxa cover 80% or more of the contig. For example, if a species-, genus-, and order-level taxa cover >80% of the contig length, then the contig would be assigned to species-level. Therefore, the simplest explanation for the higher taxa-level assignment of most of the Spades-assembled contigs would be that the proteins from the species-level taxa could not cover more than 80% of the contig sequence. To fix this issue with taxonomic classification of Deep Lake Spades-assembled metagenomes, various options in the MEGAN-LR daa2rma module, such as read assignment mode (-

ram), lowest common ancestor coverage percentage (-lcp), and minimum support (-sup), were changed and tested, but the problem persisted. As LAST/MEGAN-LR method did not work on Spades-assembled Deep Lake metagenomes, i.e., the output haloarchaea abundances were not comparable to their previously reported abundances (DeMaere et al, 2013), it was removed from Cavlab pipeline in v4.

Instead, a new approach involving the use of protein taxonomies in the Phylodist file was developed for contig-based taxonomic classification and OTU abundance estimation, and the method was added to Cavlab pipeline v4 (Appendix C). The relative OTU abundances in Ace Lake and Deep Lake Spades-assembled metagenomes calculated using the data in the Phylodist file were comparable to the relative OTU abundances in their corresponding Megahit-assembled metagenomes calculated using the LAST/MEGAN-LR output (Figure 2.6); these data were also comparable to previously reported OTU abundances in Ace Lake and Deep Lake (Ng et al, 2010, Lauro et al, 2011; DeMaere et al, 2013). Consequently, the Phylodist file-based method was used for taxonomic diversity analysis of Ace Lake, Ellis Fjord and Taynaya Bay (described in Chapters 3 and 5), and select Organic Lake metagenomes.

2.3.2 OTU bin refinement and taxonomy verification

Among all the methods for taxonomic classification and abundance estimation discussed above, the Phylodist file-based method was the most robust method for use with Spades-assembled metagenomes. As with all taxonomic classification methods that rely on query sequence alignment to reference genomes, the taxa identified using this method were considered to be the closest related species to the organisms in the metagenome, and were referred to as OTUs during analyses.

For the analysis of the taxonomic diversity of a lake system, such as Ace Lake, OTUs with relative abundance >1% were considered, due to their higher abundance contribution to the system. After running RefineM for bin refinement, the contigs that did not belong to the OTU were removed from the bin. For example, most of the contigs in *Pseudomonas alcaliphila*, *Pseudomonas pseudoalcaligenes*, and unclassified *Pseudomonas* OTU bins from Ace Lake belonged to *Pseudomonas*_E genus (97%, 90%, and 49%, respectively) (Figure 2.7). Therefore, all contigs that did not belong to this taxon were removed from the bins. Similarly, all contigs in the *C. phaeovibrioides* OTU bin from Ace Lake that did not belong to this taxon, were removed from the bin.

The taxonomies of the refined OTU bins were further verified by assessing their ANI and SSU rRNA gene identity to the reference genomes of their closest related species (Table 2.6). A few examples of the outputs of RefineM, ANI, and SSU rRNA gene identity analyses are shown in Figure 2.7 and Table 2.6. Apart from using ANI and SSU rRNA gene identity for OTU bin taxonomy verification, the OTU bins were also matched against MetaBAT-generated MAGs, to confirm their taxonomy.

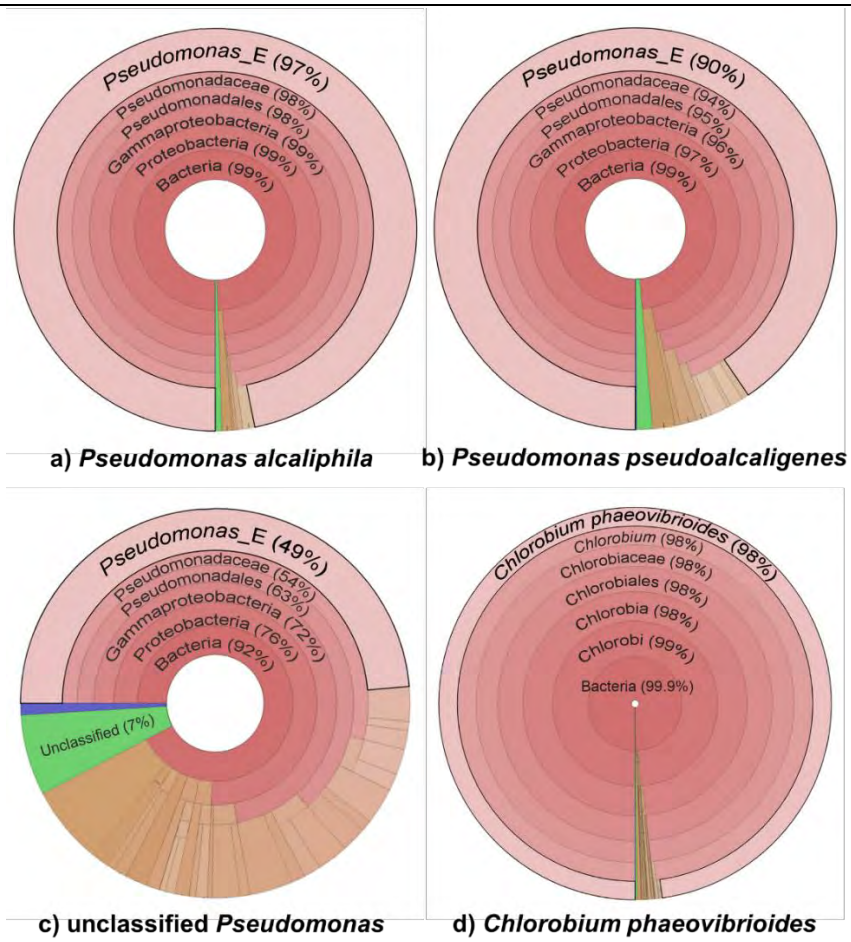


Figure 2.7 RefineM taxonomy verification output of *Pseudomonas* and *Chlorobium* OTU bins from Spades-assembled Ace Lake metagenomes. Krona radial, space-filling (RSF) display is shown for three *Pseudomonas* OTUs, **(a)** *Pseudomonas alcaliphila*, **(b)** *Pseudomonas pseudoalcaligenes*, and **(c)** unclassified *Pseudomonas*, and a *Chlorobium* OTU, **(d)** *Chlorobium phaeovibrioides*, from Ace Lake metagenomes. The bin contigs were re-assigned a taxonomy through the 'taxon_profile' module of RefineM and Krona was used for visualising the output. The percentages alongside the taxa names indicate the number of contigs that were assigned to the taxa relative to the total number of contigs in the OTU bin. For example, RefineM assigned 98% of the contigs in the *C. phaeovibrioides* bin and 97% of the contigs in the *Pseudomonas alcaliphila* bin to *C. phaeovibrioides* and *Pseudomonas*_E genus, respectively, but only 49% of

the contigs in the unclassified *Pseudomonas* bin to the *Pseudomonas*_E genus. Only the highest contributing taxa in an OTU bin have been shown in each RSF display.

Table 2.6 Verification of OTU taxonomy using RefineM, ANI, SSU rRNA identity, and matches to MetaBAT-generated MAGs. [A] The OTUs were renamed based on the outputs of RefineM, ANI, SSU rRNA gene identity, and matches to MetaBAT-generated MAGs. [PA], *Pseudomonas alcaliphila*; [PP], *Pseudomonas pseudoalcaligenes*.

Phylodist-based OTU taxonomy	*Chlorobium phaeovibrioides*	*Pseudomonas alcaliphila*	*Pseudomonas pseudoalcaligenes*	unclassified *Pseudomonas*
RefineM genome summary (% OTU contigs belonging to the indicated taxon)	*Chlorobium phaeovibrioides* (98%)	*Pseudomonas*_E *alcaliphila* (43%)	*Pseudomonas*_E *alcaliphila* (32%)	Unclassified
Reference genome/assembly (Assembly accession ID)	*Chlorobium phaeovibrioides* DSM 265 (NC_009337.1)	*Pseudomonas alcaliphila* JCM 10630 (GCF_900101755.1)	*Pseudomonas pseudoalcaligenes* CECT 5344 (GCF_000297075.2)	*Pseudomonas alcaliphila* JCM 10630 *Pseudomonas pseudoalcaligenes* CECT 5344
ANI (alignment fraction)	85% (85%)	92% (82%)	96% (78%)	[PA]: 91% (68%) [PP]: 94% (58%)
16S/18S SSU rRNA identity	99%	No match.	No match.	No match.
Taxonomy of the MetaBAT bin match	*Chlorobium phaeovibrioides*	*Pseudomonas*_E *alcaliphila*	*Pseudomonas*_E *alcaliphila*	*Pseudomonas*_E *alcaliphila*
OTU name[A]	*Chlorobium*	*Pseudomonas*_E	*Pseudomonas*_E	*Pseudomonas*_E

Based on the outputs from RefineM, ANI, and SSU rRNA gene identity analyses, and the matches to the MetaBAT-generated MAGs, the OTUs were renamed, merged, or split, if required. For example, the taxonomy of the contigs classified as *C. phaeovibrioides* according to the Phylodist file-based method was verified using RefineM, which also assigned 98% of these contigs to *C. phaeovibrioides*. Furthermore, the refined *C. phaeovibrioides* OTU bin had best matches to the *C. phaeovibrioides* MetaBAT MAG and was 99% identical to the *16S rRNA* gene of *C. phaeovibrioides* DSM 265 reference genome. However, the ANI of the OTU to the reference genome

was only 85%, across only 85% alignment fraction. Therefore, the *C. phaeovibrioides* OTU was renamed to genus-level as *Chlorobium.* Similarly, the contigs belonging to the three *Pseudomonas* OTUs had best matches to a *Pseudomonas*_E *alcaliphila*, according to RefineM as well MetaBAT MAG matches. However, based on their ANI to reference genomes, including *Pseudomonas alcaliphila*, and due to the lack of *16S rRNA* genes in the bins, the three OTUs could not be confidently renamed to species-level. Therefore, the three bins were merged and renamed to genus-level as *Pseudomonas*_E (Table 2.6). These methods were used for the analysis of Ace Lake microbial population (discussed in Chapter 3 section 3.2.2).

2.3.3 Functional potential analysis of a system using metagenomes

Various methods for functional potential analysis of a metagenome, including DIAMOND/MEGAN6-based COG analysis (section 2.3.3.1 below), methods using metagenome COG and KEGG files (sections 2.3.3.2 and 2.3.3.4 below), and arCOG analysis (section 2.3.3.3 below), were tested on a randomly selected Megahit-assembled metagenome from Deep Lake surface 0.1 μm-filter fraction from Dec 2013 (Appendix A).

2.3.3.1 DIAMOND/MEGAN6 COG analysis

DIAMOND/MEGAN6 COG output showed the number of proteins assigned to all COG categories, which gave a general idea about the functional distribution of annotated proteins in the metagenome. For example, in the Megahit-assembled Deep Lake surface metagenome (0.1 μm-filter Dec 2013), most of the proteins associated with metabolism were assigned to COG category [E] 'amino acid transport and metabolism' and [C] 'energy production and conversion', which included enzymes for amino acid and glycerol metabolism, respectively (Figure 2.8). This coincides with the requirements of the most abundant haloarchaea in Deep Lake, namely *Hht. litchfieldiae*, that prefers glycerol and other carbohydrates as a carbon source (DeMaere et al, 2013; Williams et al, 2017). The other prominent haloarchaea in Deep Lake, namely DL31 and *Hrr. lacusprofundi*, require amino acids (DeMaere et al, 2013; Williams et al, 2017). In MEGAN6 COG data, the proteins were assigned to multiple COG categories, if they had hits to proteins from more than one COG category. Due to this, the sum of the number of proteins assigned to all COG categories (33,997) and 'No hits' (60,365) was more than the total annotated proteins in the metagenome (93,645) (Figure 2.8).

It is worth noting that many of the proteins were not assigned to any COG categories and were grouped as 'No hits', and it is possible that these proteins are novel, especially considering their origin from an extremely cold lake, Deep Lake. Another limitation of this method was that the MEGAN6 eggNOG mapping file did not include the COG category [X] 'mobilome: prophages, transposons' and, therefore, the proteins that should be assigned to category [X] would be instead classified under category [L] 'replication, recombination and repair' or would be poorly characterised (Galperin et al, 2015). Despite these limitations, the DIAMOND/MEGAN6 COG data can be used for a quick, initial functional potential analysis of a large number of metagenomes, especially in the comparative analysis mode on MEGAN6 GUI (as shown in Figure 2.3), which allows for simultaneous comparison of data from multiple metagenomes.

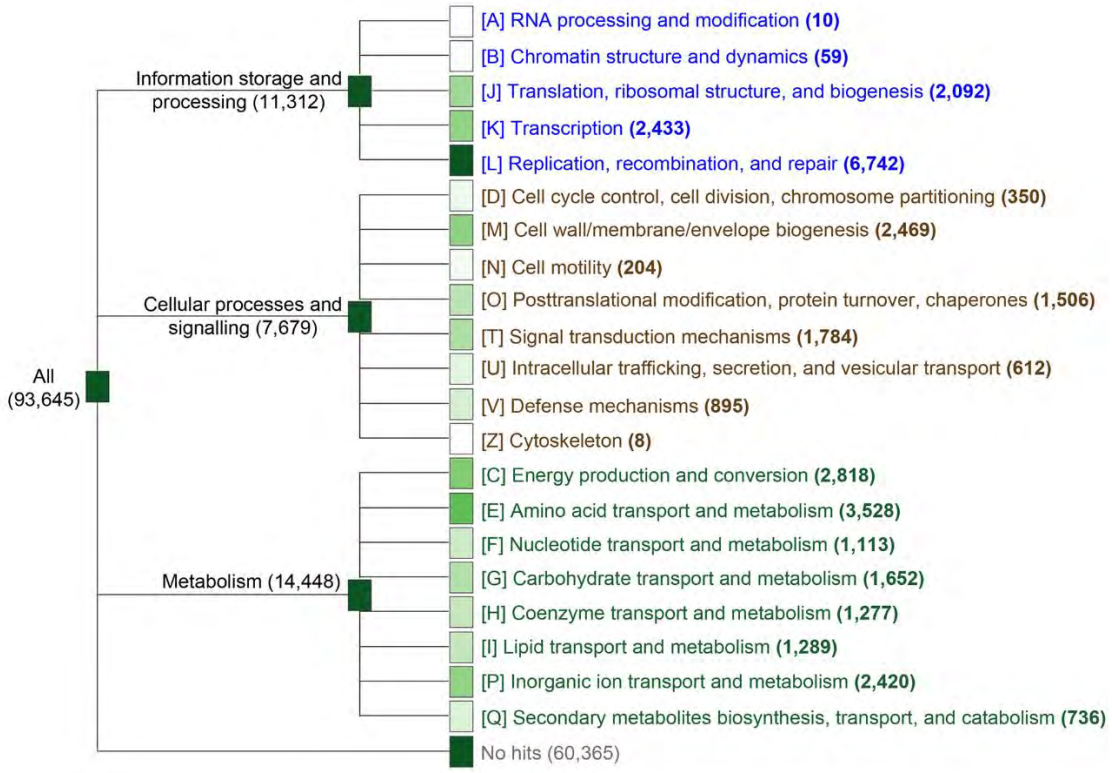


Figure 2.8 MEGAN6 COG data-based functional potential analysis of a Megahit-assembled Deep Lake metagenome. The dendrogram shows the COG categorization of the annotated proteins in a metagenome from the Deep Lake surface (collected on a 0.1 µm-filter in Dec 2013), using DIAMOND/MEGAN6 module of Cavlab pipeline v3.1a. Tree node gradient (green gradient) indicates the number of proteins assigned to the node, which are also mentioned in brackets alongside the node names. Only COG categories with at least one protein assigned to them are shown here.

2.3.3.2 IMG COG annotation data-based analysis

The metagenome COG file-based analysis tested on the Megahit-assembled Deep Lake surface metagenome (0.1 μm-filter Dec 2013) yielded slightly different results to the DIAMOND/MEGAN6 COG data, due to the major differences in the calculation of the output (Figure 2.9a). DIAMOND/MEGAN6 COG output indicates the number of proteins assigned to a COG category (Figure 2.8), whereas the metagenome COG file output represents the abundance of a COG category calculated by summing the read depths of contigs corresponding to predicted genes that were assigned to the COG category (Figure 2.9a). Another difference between the two methods is that the metagenome COG file analysis does not allow for multiple COG category assignments, i.e., each protein was assigned to a single COG category. Moreover, the COG category [X] was a part of the metagenome COG file-based analysis (Figure 2.9a). Regardless of these differences between the two methods, the overall distribution of the annotated proteins/predicted genes in the Deep Lake metagenome estimated using the two methods was similar (Figures 2.8 and 2.9a). For example, most annotated proteins/predicted genes belonged to category [L] in the outputs from both methods (Figures 2.8 and 2.9a). Also, among the metabolism-related COG categories, most annotated proteins/predicted genes belonged to categories [C] and [E] (Figures 2.8 and 2.9a). However, the output of the COG file-based analysis was considered better than the DIAMOND/MEGAN6 COG data because: (i) it reported the abundance of the COG categories, and not just the number of proteins assigned to a COG category, and (ii) it included the COG category [X], which is an indicator of viral content in a metagenome. Nevertheless, the DIAMOND/MEGAN6 method for COG analysis was retained in the Cavlab pipeline v4.1, because its output can be used to look at individual proteins that were assigned to a COG category and its protein taxonomy is required for arCOG analysis (described below in section 2.3.3.3). The metagenome COG file-based analysis was performed on Ace Lake time-series metagenomes for an in-depth analysis of the system (discussed in Chapter 3 section 3.3.7).

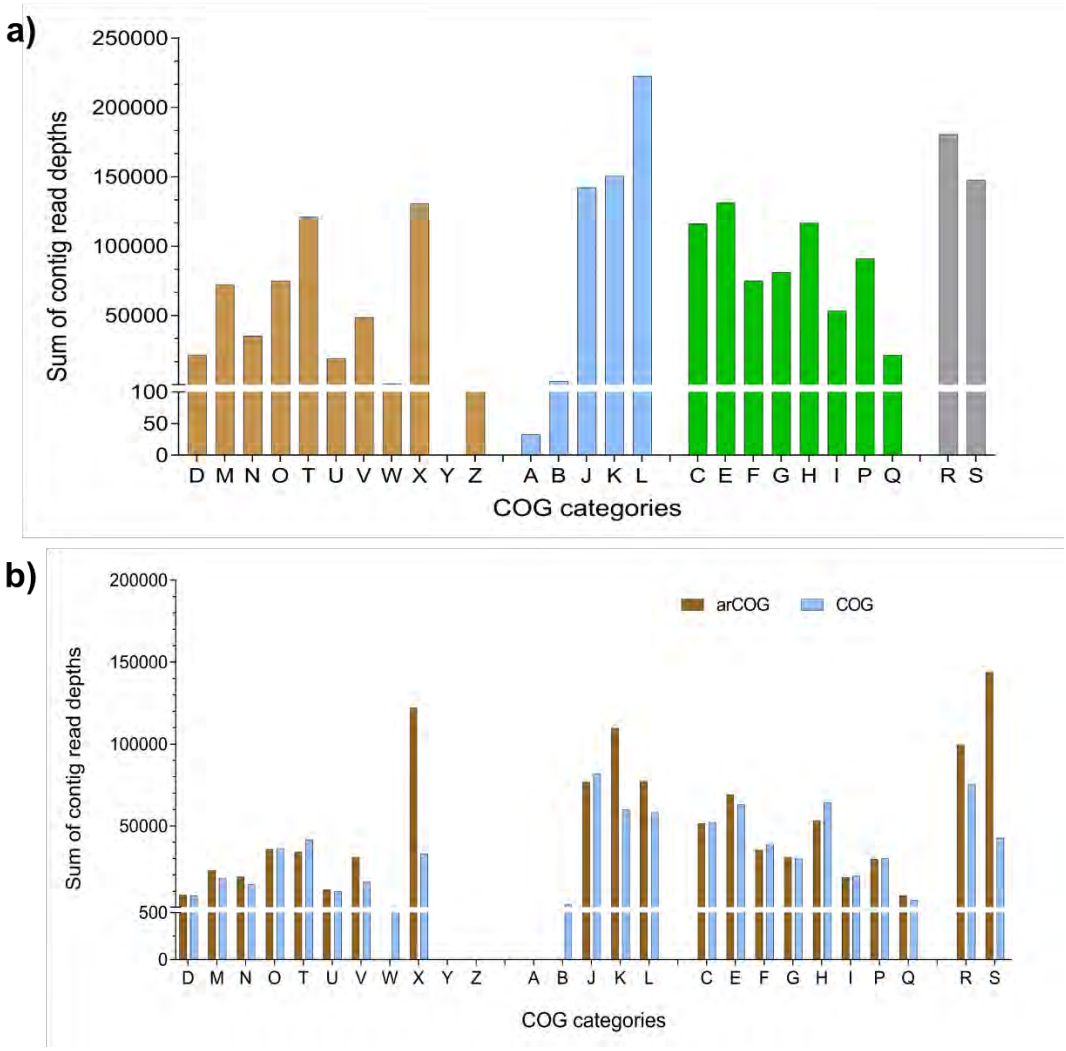


Figure 2.9 COG and arCOG functional potential analyses of a Megahit-assembled Deep Lake metagenome. The bar charts depict COG category abundances, measured by summing the read depths of contigs corresponding to predicted genes/annotated proteins assigned to a COG category. The metagenome was from a surface sample collected from Deep Lake on a 0.1 µm-filter in Dec 2013 (Appendix A). **a) COG analysis:** The graph shows the COG categorisation of all predicted genes in the metagenome, determined using the data in the IMG COG annotation file and the NCBI COG database with the COG analysis component of Cavlab pipeline v3.3. The bar colours represent COG categories associated with cellular processes and signalling (brown), information storage and processing (blue), metabolism (green), and poorly characterized genes (grey). The y-axis was split to show an expanded view of values below 100. **b) arCOG and COG comparative analysis:** The metagenome proteins classified as Archaea using the DIAMOND/MEGAN6 taxonomic classification method and the NCBI arCOG database were used for generating the arCOG data (brown). The predicted genes in the IMG COG annotation file that corresponded to the archaeal proteins and the NCBI COG database were used to produce the COG output (blue) (Appendix D). The COG categories on the x-axis

are grouped based on their association with cellular processes and signalling (D, M, N, O, T, U, V, W, X, Y, Z), information storage and processing (A, B, J, K, L), metabolism (C, E, F, G, H, I, P, Q), and poorly characterized genes (R, S). The y-axis was split to show an expanded view of values below 500. Cellular processes and signalling — D, cell cycle control, cell division, chromosome partitioning; M, cell wall/membrane/envelope biogenesis; N, cell motility; O, post-translational modification, protein turnover, and chaperones; T, signal transduction mechanisms; U, intracellular trafficking, secretion, and vesicular transport; V, defence mechanisms; W, extracellular structures; X, mobilome: prophages, transposons; Y, nuclear structure; Z, cytoskeleton. Information storage and processing — A, RNA processing and modification; B, chromatin structure and dynamics; J, translation, ribosomal structure and biogenesis; K, transcription; L, replication, recombination and repair. Metabolism — C, energy production and conversion; E, amino acid transport and metabolism; F, nucleotide transport and metabolism; G, carbohydrate transport and metabolism; H, coenzyme transport and metabolism; I, lipid transport and metabolism; P, inorganic ion transport and metabolism; Q, secondary metabolites biosynthesis, transport, and catabolism. Poorly characterized — R, general function prediction only; S, function unknown.

2.3.3.3 arCOG analysis

Nearly all Antarctic metagenomes from hypersaline systems in the Vestfold Hills and the Rauer Islands showed high abundance of haloarchaea (Figure 2.5), which makes it important to use methods that specifically consider the archaeal population of these hypersaline systems. The COG numbers generally used for COG categorisation are mostly associated with bacteria; the updated NCBI COG database was prepared using proteins from 628 bacterial and 83 archaeal genomes (https://ftp.ncbi.nih.gov/pub/COG/COG2014/static/lists/homeCOGs.html), and no longer includes eukaryotic COGs (Galperin et al, 2015). However, with the availability of data on archaea-specific COGs (arCOGs) on NCBI (ftp://ftp.ncbi.nih.gov/pub/wolf/COGs/arCOG), prepared using proteins from 168 archaeal genomes, including 27 *Halobacteria* genomes, a more accurate functional distribution of archaea-rich systems can be estimated. To use arCOG for the functional analysis of archaea-rich metagenomes, only proteins classified as Archaea by the DIAMOND/MEGAN6 taxonomic classification method were considered. The arCOG number-based analysis was tested on the archaeal proteins from a Megahit-assembled Deep Lake surface metagenome (0.1 μm-filter Dec 2013). For comparison between the arCOG and COG number assignments of the archaeal proteins, the predicted genes in

the metagenome COG file that corresponded to the archaeal proteins were also assigned COG categories. It was observed that the arCOG number-based assignment of the archaeal proteins to some of the COG categories was better than their COG number-based assignment (Figure 2.9b). For example, the abundance of COG category [X] was much higher in the output from arCOG number-based analysis, which indicated that more archaeal proteins were assigned to this COG category in arCOG analysis than in COG analysis. This difference might also be because not all predicted genes associated with the archaeal proteins had a COG number assignment in the metagenome COG file. The arCOG analysis outputs also included a file containing the archaeal protein IDs and their arCOG number assignments, which can be used for studying individual proteins assigned to a COG category. For example, the archaeal proteins assigned to category [X] can be identified using the arCOG analysis output and their taxonomic assignments can be assessed using the DIAMOND/MEGAN6 output, which was used for extracting the archaeal proteins (section 2.2.3.3). Therefore, the arCOG analysis was found to be useful for assessing the functional distribution of systems that mainly harbour archaea.

2.3.3.4 KEGG analysis

KEGG analysis of the Megahit-assembled Deep Lake surface metagenome (0.1 μm-filter Dec 2013) was performed using the data in the metagenome KEGG file, and the pathway/enzyme abundances were calculated from specific KO numbers (Appendix F). In the Deep Lake surface metagenome, the abundance of genes associated with aerobic respiration and glycerol metabolism was high (Figure 2.10), which coincided with the high abundance of aerobic haloarchaea in the system that are known to utilise glycerol as a major carbon source (DeMaere et al, 2013; Williams et al, 2017). Defence-associated CRISPR-Cas core, CRISPR type 1I, and CRISPR type 2IIB genes also showed high abundance in the Deep Lake surface metagenome (Figure 2.10), which probably corresponded to the most abundant haloarchaea in the lake that have been shown to possess genes for CRISPR type 1I system (type I-B, I-D, or both; Tschitschko et al, 2015). Contrarily, genes associated with processes or microbes that usually occur in anoxic environment, such as Wood-Ljungdahl pathway, methanogenesis, dissimilatory sulfate reduction and oxidation, and reaction core complex of GSB, were not observed in Deep Lake (Figure 2.10), which corroborates the oxic conditions of the lake system that mixes completely at least once a year.

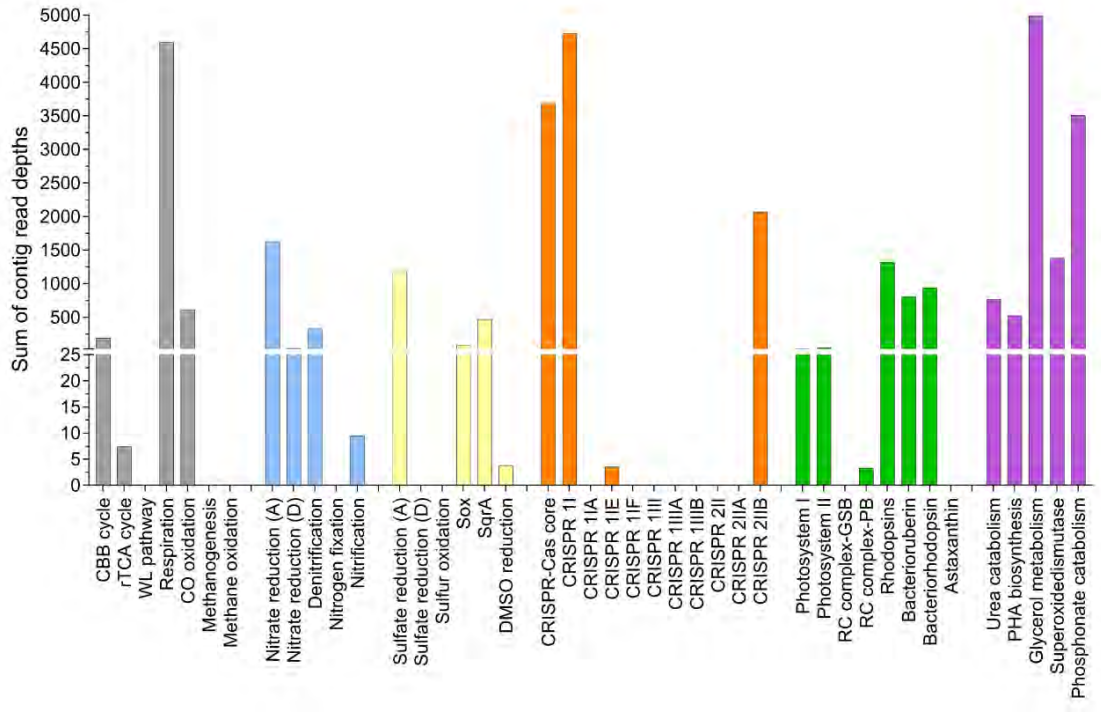


Figure 2.10 KEGG functional potential analysis of a Megahit-assembled Deep Lake metagenome. The bar chart depicts abundances of specific pathways/enzymes, calculated by summing the read depths of contigs corresponding to predicted genes associated with specific KO numbers (Appendix F), using the KEGG analysis module of Cavlab pipeline v3.3 (see Cavlab pipeline v1.2 in Appendix B for code reference). The metagenome was from a surface sample collected from Deep Lake on a 0.1 μm-filter in Dec 2013. The bar colours represent pathways/enzymes associated with carbon cycle (grey), nitrogen cycle (blue), sulfur cycle (yellow), defence mechanisms (orange), light-based energy production (green), and other processes (purple). The y-axis was split to show an expanded view of values below 25. (A), assimilatory; Cas, CRISPR-associated; CBB cycle, Calvin–Benson–Bassham cycle; CO oxidation, carbon monoxide oxidation; CRISPR, clustered regularly interspaced short palindromic repeats; (D), dissimilatory; DMSO, dimethyl sulfoxide; PHA, polyhydroxyalkanoate; RC complex-GSB, reaction centre complex-green sulfur bacteria; RC complex-PB, reaction centre complex-purple bacteria; rTCA cycle, reverse tricarboxylic acid cycle; Sox, sulfur oxidation; SqrA, sulfide quinone reductase A; WL pathway, Wood-Ljungdahl pathway.

While the COG analysis can give a broad overview of the functional distribution of a system or metagenome, the KEGG analysis allows for a more detailed investigation of individual pathways and the organisms that might contribute to it, giving a clearer picture of the biogeochemistry of the system. This KEGG analysis was used for an in-

depth assessment of Ace Lake time-series metagenomes (discussed in Chapter 3 section 3.3.7.1).

2.3.4 Metagenome statistical analyses

PRIMER v7 analysis of Megahit-assembled metagenomes from hypersaline lakes in the Vestfold Hills (Deep Lake and Club Lake) and the Rauer Islands (Rauer Lake 1, 3, 6, and 13) showed that the taxonomic composition of the lakes from the two Antarctic zones was quite different (Figure 2.11a). The Vestfold Hill hypersaline lakes showed very high relative abundance of Archaea (peak relative abundance: 93%) and comparatively low relative abundance of Viruses, Bacteria, and Eukarya (peak relative abundances: 10%, 3%, and 0.5%, respectively) (Figure 2.11a). On the other hand, Rauer Island hypersaline lakes showed very high relative abundance of either Bacteria only (as seen in Rauer 1 Lake: 66%) or both Bacteria and Archaea (as seen in Rauer Island lakes 3, 6, and 13 — peak relative abundance: 66% and 52%, respectively), along with Eukarya and Viruses (peak relative abundances in all Rauer Island lakes: 10% and 24%, respectively) (Figure 2.11a). This difference in the taxonomic composition was also marked by the separate clustering of the Vestfold Hill metagenomes from the Rauer Island lake metagenomes (Figure 2.11b).

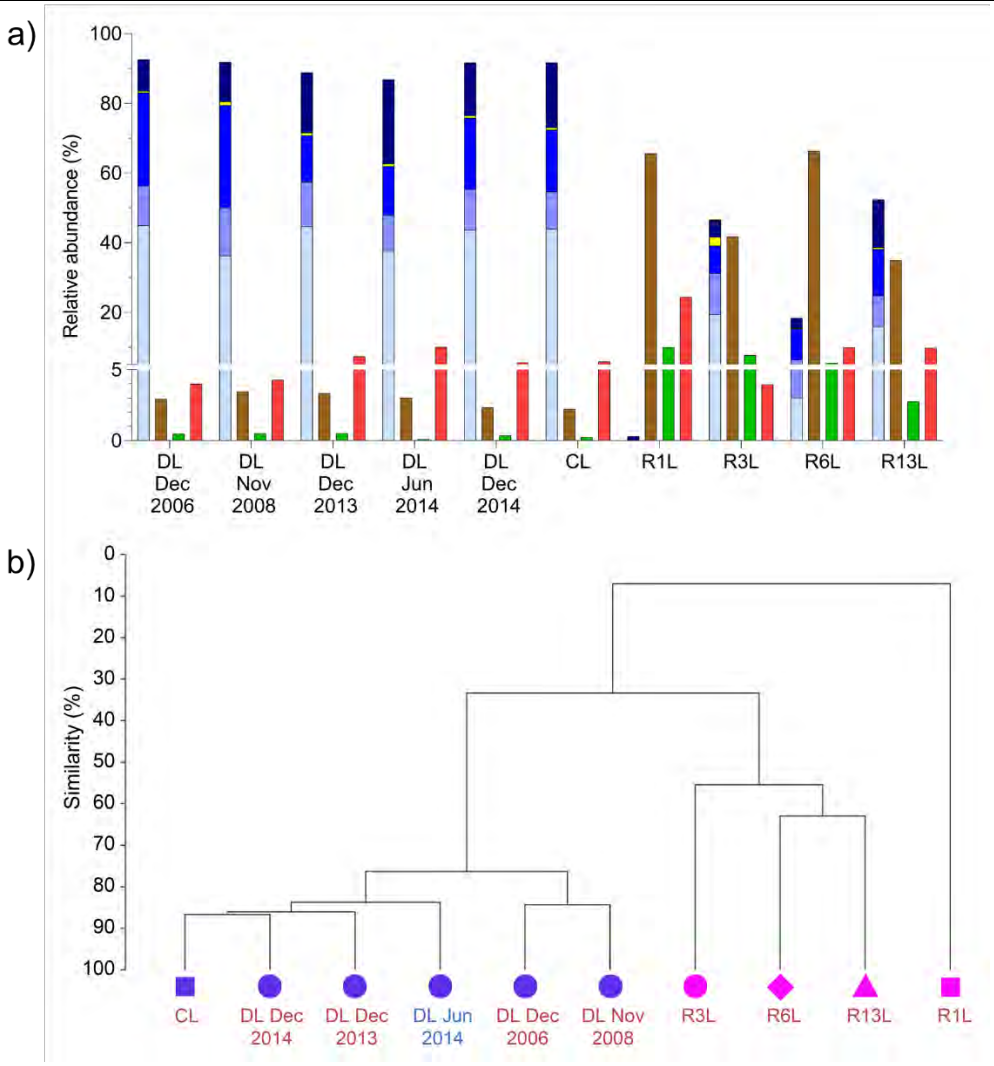


Figure 2.11 PRIMER v7 analysis of Megahit-assembled metagenomes from Antarctic hypersaline lake systems — taxonomic diversity and sample clustering. (a) The bar chart shows relative abundances of Archaea (blue-shaded stacked bar), Bacteria (■), Eukarya (■), and Viruses (■) in metagenomes from the surface of Deep Lake (DL), Club Lake (CL), Rauer 1 Lake (R1L), Rauer 3 Lake (R3L), Rauer 6 Lake (R6L), and Rauer 13 Lake (R13L). Archaea relative abundances are shown using a stacked bar, which includes relative abundances of *Halohasta litchfieldiae* (■), *Halorubrum lacusprofundi* (■), halophilic archaeon DL31 (■), *Halobacterium* sp. DL1 (■), and Other Archaea (■). **(b)** The dendrogram shows clustering of samples from six hypersaline Antarctic lakes, two from the Vestfold Hills — Deep Lake (DL; ●) and Club Lake (CL; ■), and four from the Rauer Islands — Rauer 1 Lake (R1L; ■), Rauer 3 Lake (R3L; ●), Rauer 6 Lake (R6L; ◆), and Rauer 13 Lake (R13L; ▲). The samples from all lakes were collected in Summer (red colour font), except Jun 2014 sample from Deep Lake, which was from Winter (blue colour font). The y-axis indicates the percentage Bray-Curtis similarity between the different samples from the six hypersaline lakes.

The alpha diversity and species evenness of the hypersaline lakes from the Vestfold Hills were lower than that of the Rauer Island hypersaline lakes, because of the high abundance of the three dominant haloarchaea, *Hht. litchfieldiae*, *Hrr. lacusprofundi*, and DL31, in Deep Lake and Club Lake (Figures 2.11a and 2.12a). Contrarily, the species richness of Vestfold Hill hypersaline lakes was generally higher than that of Rauer Island hypersaline lakes, except Rauer 1 Lake (Figure 2.12b). The total OTUs identified in Deep Lake merged metagenome from Nov 2008 was very low, because it included data from only 3 and 0.8 µm-filter metagenomes, as 0.1 µm-filter metagenome from that time period was not available at the time of this analysis.

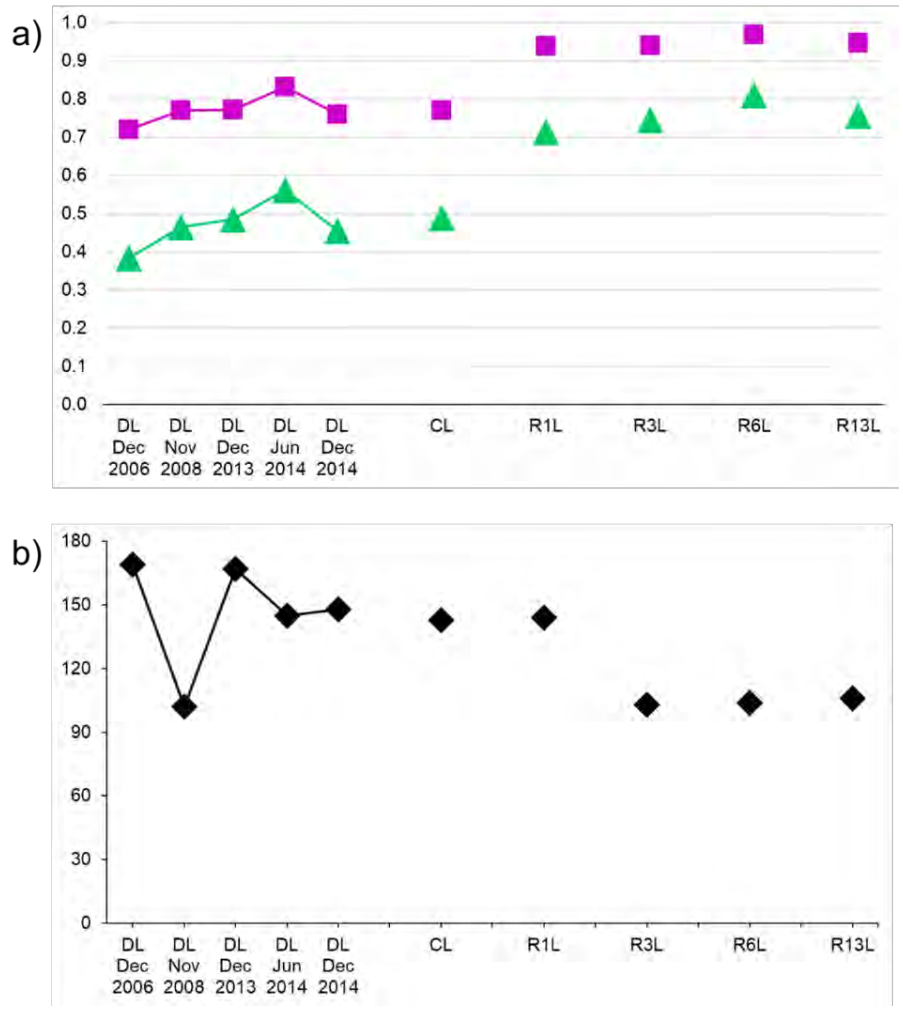


Figure 2.12 PRIMER v7 analysis of Megahit-assembled metagenomes from Antarctic hypersaline lake systems — species diversity analysis. The line graphs show **(a)** species diversity (■) and species evenness (▲) as well as **(b)** species richness (◆) in the time-series metagenomes from Deep Lake (DL) surface and in metagenomes from the surface of Club Lake (CL), Rauer 1 Lake (R1L), Rauer 3 Lake (R3L), Rauer 6 Lake (R6L), and Rauer 13 Lake (R13L). The y-axis in **(a)** represents Simson's index of diversity and Pielou's index of species

evenness; both diversity measures range between 0 and 1. The y-axis in **(b)** represents the total number of OTUs identified in a metagenome, which was used as a measure of species richness.

A total of 467 distinct OTUs were identified in all hypersaline metagenomes studied, which showed that PRIMER v7 was capable of performing multivariate analyses on at least hundreds of variables from multiple samples at a time. The software was also very reliable, such that its output was largely unaffected by the grouping of species-level OTUs to higher taxa levels. For example, the clusters in Figure 2.11b were based on the relative abundances of a total of 467 OTUs identified in the 10 merged metagenomes — 5 from Deep Lake, 1 from Club Lake, and 4 from Rauer Island lakes. Similar clustering pattern was observed when the OTU abundances were grouped as shown in Figure 2.11a, to *Hht. litchfieldiae*, *Hrr. lacusprofundi*, DL31, DL1, Other Archaea, Bacteria, Eukarya, and Viruses, except that the samples from Rauer 3 Lake and Rauer 6 Lake switched places.

Apart from computing multivariate statistical analyses, PRIMER v7 software can also be used for creating a variety of plots, such as line, bar, and scatter plots, heat maps (with or without cluster overlay), dendrograms, PCA plots, and dbRDA (distance-based redundancy analysis) plots. Therefore, the software was used for an in-depth analysis of Ace Lake time-series metagenomes, which also included analysis of the relationship between relative OTU abundances and environmental factors (discussed in Chapter 3 section 3.2.4.2).

2.3.5 Genomic analyses

The key bacteria in Ace Lake, namely *Chlorobium* and *Synechococcus*, were studied to assess their phylogenetic relationships to known species of the genera. Methods and software, such as BLAST+ and IGV, Mauve, and MEGA were used on the OTUs or MAGs of the two bacteria to achieve this. The analyses of these two microbes are described in Chapters 4 and 5.

2.3.6 Development of a metagenome analysis pipeline

The preliminary Cavlab pipeline v1.2 (Appendix B) was upgraded over the years, as more methods were explored. With the goal for improved metagenomic analyses, major additions/changes were made to the pipeline — Cavlab pipeline v4.1 being the latest version (Figure 2.13; Table 2.7; Appendix C).

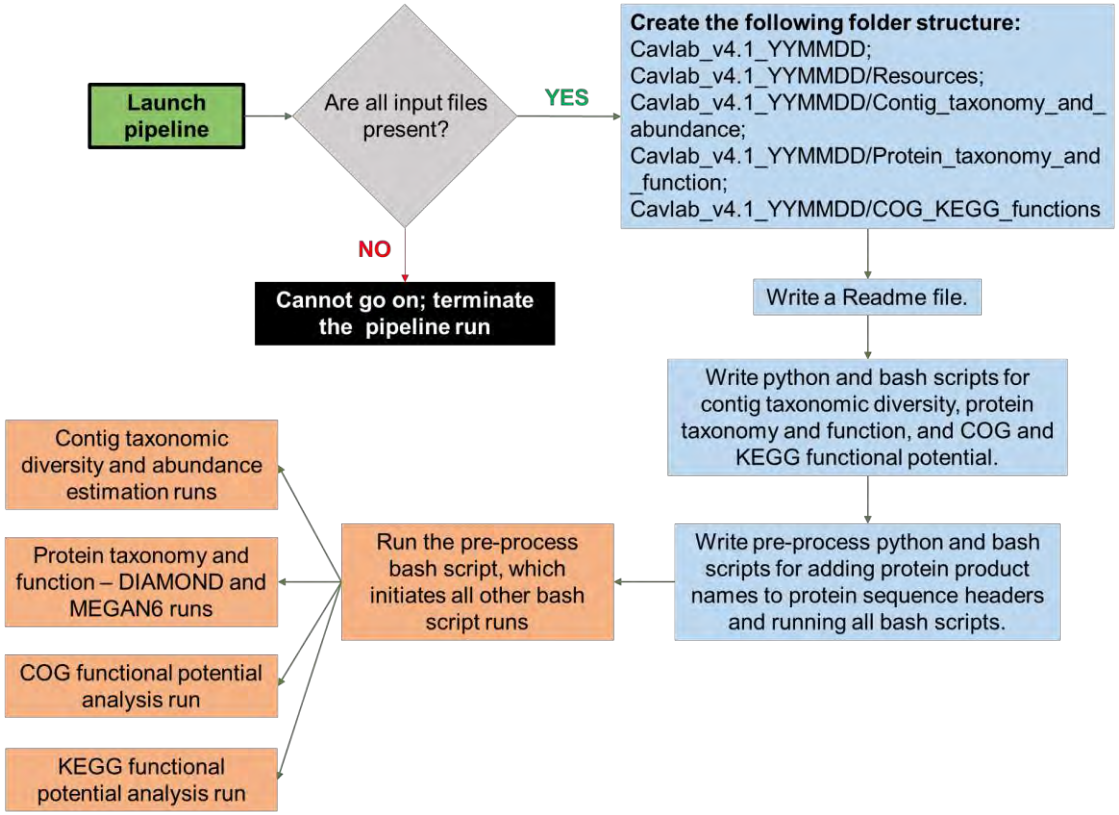


Figure 2.13 Cavlab pipeline v4.1 schematic. The image outlines the workings of the latest version of Cavlab pipeline v4.1, along with the methods and software employed for various analyses in it (Appendix C). It also creates a folder structure for storing the output, with Cavlab_v4.1_YYMMDD as the head folder. The analyses scripts and output log files are stored in Cavlab_v4.1_YYMMDD/Resources subfolder; contig taxonomic classification and abundance estimation output is stored in Cavlab_v4.1_YYMMDD/Contig_taxonomy_and_abundance subfolder; protein taxonomy and function analysis output is stored in Cavlab_v4.1_YYMMDD/Protein_taxonomy_and_function subfolder; and outputs of functional potential analyses are stored in Cavlab_v4.1_YYMMDD/COG_KEGG_functions subfolder.

All versions of the Cavlab pipeline were coded to exploit the folder structure of the IMG annotation output data, so that once launched, the pipeline would select the correct files for various metagenome analyses, with minimal user input. For example, JGI IMG provides filtered sequencing reads in a folder named ‘QC_Filtered_Raw_Data’, therefore, the pipeline would detect and select the correct read sequence file in this folder and perform various read-based metagenome analyses on it, without the user having to prompt the exact file location and name. The pipeline starts with a search for specific input files for the analyses, such as the filtered read FASTQ file, contig and protein sequence FASTA files, Phylodist file, metagenome COG and KEGG files, and

contig coverage file. Once the presence of these resource files is verified, the script creates folders and subfolders for storing pipeline outputs, followed by preparation of python and bash scripts for individual analyses, which have been discussed in section 2.2.

Some of the major changes to the original pipeline include: (i) a shift from read-based (using PhyloSift) to contig-based (using Phylodist file) taxonomic diversity analysis, the latter of which was found to be reliable and was used for in-depth analysis of Ace Lake metagenomes (Chapter 3); (ii) the COG analysis runtime was reduced from >100 h to <10 mins; (iii) the COG database was updated, which included the new COG category [X] 'mobilome: prophages, transposons', and the COG abundance calculations were improved (section 2.2.3.2); (iv) additional KO numbers and databases associated with a variety of metabolic pathways/enzymes were added to the pipeline (Tables 2.2 and 2.7; Appendix F) and the pathway/enzyme abundance calculations were improved (section 2.2.3.2); (v) the pipeline output folder structure was also modified, such that the output of each analysis was stored in separate, specific folders; (vi) the code for the search of input resource files, such as protein and contig sequence files, was also improved by incorporating a variety of file designations used by JGI IMG, including the latest file designations.

Apart from these, various methods were added to the pipeline for improved metagenome analysis, but were then removed either because their output did not corroborate previous findings that were used as references or a better analysis was available. For example, MetaPhlA2 was added to Cavlab pipeline in v3.0 for read-based taxonomic diversity analysis and abundance estimation, but was removed in v3.1 because its clade-specific database proved to be insufficient for taxonomic classification of Antarctic metagenome reads (Table 2.3). Similarly, a CRISPR script that was added to the pipeline in v2.0 was removed in v3.0, when a more in-depth virus analysis of Antarctic metagenomes was made available on IMG-VR (Páez-Espino et al, 2017). Also, the LAST/MEGAN-LR method was added to the pipeline in v3.0 for contig-based taxonomic diversity analysis, after it was successfully tested on various Megahit-assembled metagenomes (section 2.3.1.3), but was removed in v4 because it did not work well with the new Spades-assembled metagenomes (Figure 2.6).

Table 2.7 Cavlab pipeline v1.2 *vs* v4.1 — comparison of methods/software, input files (I) and UNSW Katana computer cluster resources (K). Katana resources: Memory, random-

access memory (RAM) allotted for processing job on server; Wall time, maximum time allotted for running job on server; Nodes, server's computer node on which the job will run; processors, number of central processing unit (CPU) cores of the computer node used for running jobs parallelly. * KEGG database files were specifically created for KO numbers that represented enzymes catalysing redox reactions (e.g., sulfide oxidation and sulfate reduction) or homologous enzymes (e.g., ammonia/methane monooxygenase) (section 2.2.3.2). The database files included protein sequences of the enzymes and their previously observed functional roles.

Metagenome analysis	Cavlab pipeline v1.2 (Appendix B)	Cavlab pipeline v4.1 (Appendix C)
Pre-process Includes steps for detection and verification of input files and creating file paths. Creates a Readme file with details of pipeline methods and software used.	Python v3.5.2 Output folder structure: 'Cav_LaunchDate' as head folder for all outputs 'Cav_LaunchDate/resources' for method scripts and log files 'Cav_LaunchDate/metabat' for initial steps of MetaBAT Adding contig read depth to corresponding protein sequence headers for protein taxonomy runs with DIAMOND/MEGAN. **K:** Memory = 8 GB; Wall time = 12 hr; Nodes: processors = 1:1 **I:** Protein sequence file; Contig coverage file; Scaffold to contig mapping file.	Python v3.8.2 Output folder structure: 'Cavlab_v4.1_YYMMDD' as the head folder 'Cavlab_v4.1_YYMMDD/Resources' for method scripts and log files 'Cavlab_v4.1_YYMMDD/Contig_taxonomy_and_abundance' 'Cavlab_v4.1_YYMMDD/Protein_taxonomy_and_function' 'Cavlab_v4.1_YYMMDD/COG_KEGG_functions' Adding annotated product names to protein sequence headers for protein taxonomy and function runs with DIAMOND/MEGAN. **K:** Memory = 8 GB; Wall time = 12 hr; Nodes: processors = 1:1 **I:** Protein sequence file; IMG protein function annotation file.
Taxonomic classification	Uses PhyloSift v1.0.1, Perl v5.20.1, HMMER v3.1b2, RAxML v8.1.17, FastTree v2.1.7, Pplacer v1.1.alpha16.	Uses Python v3.8.2 Performs contig-based taxonomic diversity analysis and abundance estimation.

	Performs read-based taxonomic diversity analysis. **K:** Memory = 24 GB; Wall time = 200 hr; Nodes: processors = 1:1 **I:** Filtered reads file	**K:** Memory = 96 GB; Wall time = 12 hr; Nodes: processors = 1:1 **I:** IMG protein taxonomy annotation file; IMG protein function annotation file; Contig coverage file; Scaffold to contig mapping file.
Protein taxonomy and functional potential analysis Use DIAMOND and MEGAN for protein alignment and classification, respectively	DIAMOND v0.8.4, MEGAN v6.4.5, and Java v8u45 Input protein sequence file has read depths of corresponding contigs added to the protein header names. **K:** Memory = 63 GB; Wall time = 48 hr; Nodes: processors = 1:8 **I:** Protein sequence file	DIAMOND v0.9.31, MEGAN v6.15.1, Input protein sequence file has annotated product names added to the protein header names. **K:** Memory = 120 GB; Wall time = 48 hr; Nodes: processors = 1:16 **I:** Protein sequence file
COG analysis Uses Python script on metagenome COG file	Python v3.5.2 Uses NCBI COG database, 2003 version, which does not include COG category [X] (mobilome: prophages, transposons). **K:** Memory = 12 GB; Wall time = 48 hr; Nodes: processors = 1:1 **I:** Metagenome COG file; Contig coverage file; Scaffold to contig mapping file.	Python v3.8.2 Uses NCBI COG database, 2014 update version, which includes COG category [X] **K:** Memory = 64 GB; Wall time = 12 hr; Nodes: processors = 1:1 **I:** IMG COG annotation file; Contig coverage file; Scaffold to contig mapping file.
KEGG analysis Uses Python script on metagenome KEGG file (see **Appendix F** for a	Python v3.5.2 Analyses 118 KO numbers and 44 pathways/enzymes. Uses 4 KEGG database file*.	Python v3.8.2 Analyses 427 KO numbers and 173 pathways/enzymes. Uses 8 KEGG database files*.

list of KO numbers used for analysis)	**K:** Runs performed with COG analysis. **I:** Metagenome KEGG file; Contig coverage file; and Scaffold to contig mapping file.	**K:** Memory = 8 GB; Wall time = 12 hr; Nodes: processors = 1:1 **I:** IMG KEGG annotation file; Contig coverage file; Scaffold to contig mapping file.
MetaBAT data preparation	Python v3.5.2 **K:** Memory = 31 GB; Wall time = 12 hr; Nodes: processors = 1:4 **I:** Contig sequence file; Filtered reads file	The initial steps for MetaBAT were removed.

Other than the Cavlab pipeline for metagenome analyses, a python-based pipeline was written for the analysis of arCOGs (Appendix D). This script was not a part of Cavlab pipeline, because it relied on the output of DIAMOND/MEGAN6 component of the Cavlab pipeline, which needed to be handled manually for the preparation of an input file for arCOG pipeline (section 2.2.3.3).

2.4 Conclusion

A major challenge of working with metagenomes is the size of the dataset, but a major advantage of having metagenomes is the amount of information made available. It is a powerful tool for understanding microbial life as it is in its natural habitat. The advances in HTS have allowed for parallel sequencing of multiple DNA samples, making metagenomic studies possible. As sequencing methods continue to improve, so do the methods for analysis of the sequencing data. With the availability of many methods for various kinds of metagenomic analyses, some of which are described in this chapter, it was tricky to choose the right set of methods for the analysis of the Antarctic metagenomes. While some methods or software appeared promising, considering their algorithm or approach, the only way to assess their worth was to test them on real datasets. For example, MetaPhlAn2 is a taxonomic classification method specifically developed for the analysis of metagenomes (Segata et al, 2012), but because it relies on

clade-specific markers, which were developed from the genomes of well-characterised microbes, it could not be effectively used for an initial analysis of Antarctic metagenomes (Table 2.3).

Apart from the advances in metagenome sequencing, the methods for their assembly and annotation are also improving, which can impact the methods used for metagenomic analysis. For example, the LAST/MEGAN-LR method selected for contig taxonomic classification of Megahit-assembled metagenomes did not work on Spades-assembled metagenomes, which were more recent (Figure 2.6). However, the Spades-assembled metagenomes had better contig statistics than the Megahit-assembled metagenome contigs (Table 2.5). The Spades-assembled contigs were much longer and some even represented complete phage (discussed in Chapter 3 section 3.2.6.4).

Considering the various ways in which a metagenome can be analysed, it can be useful to have a pipeline that performs an initial set of analyses, such as taxonomic diversity and functional potential analyses, that are detailed enough to be comprehensive. The Cavlab pipeline was developed keeping this in mind and was improved over-time to include some new methods/software that might perform better than existing pipeline methods. However, for an in-depth analysis of a metagenome or an environment, additional methods need to be applied. For example, apart from assessing the taxonomic composition of a system, it is important to study the key species in the system, for which various genomic analyses need to be performed, some of which were discussed in this chapter.

Based on all the methods tested, the following methods/software were found to be useful for the analysis of Antarctic metagenomes:

a) Phylodist file-based method for contig taxonomic classification and OTU abundance estimation (part of Cavlab pipeline v4.1; Appendix C).
b) Metagenome COG and KEGG file-based methods (part of Cavlab pipeline v4.1; Appendix C) and arCOG analysis (arCOG pipeline v1.2; Appendix D) for functional potential analyses.
c) RefineM, ANI, and SSU rRNA gene identity for OTU bin refinement and taxonomy verification.
d) PRIMER v7 for multivariate statistical analyses.
e) BLAST+/IGV, Mauve, and MEGA for genomic analyses of OTUs and MAGs.

Cavlab pipeline v4 and the methods for OTU bin refinement, taxonomy verification, genomic analysis, and statistical analysis discussed in this chapter were used for the in-depth study of time-series Spades-assembled metagenomes from Ace Lake (discussed in Chapters 3, 4 and 5). Cavlab pipeline v4.1 can be used with all Antarctic metagenomes available on the Katana scratch node, which are also available online on IMG website (https://img.jgi.doe.gov/cgi-bin/m/main.cgi). The pipeline was modified, and recently tested (on 19 June, 2020), to accept the latest JGI IMG file designations, and should run without errors.

3. Seasonal variation in Ace Lake biodiversity and the functional potential of its microbial community

3.1 Introduction

Ace lake is a marine-derived, meromictic lake in the Vestfold Hills, with an Upper oxic zone and a Lower anoxic zone separated by an oxycline/halocline (Burton, 1980). The lake is covered by thick ice for ~11 months of the year, which melts in summer forming a layer of fresh water on the lake surface (Hand and Burton, 1981). With approaching winter, the ice cover reforms, which causes the salt in the water to be removed into the surrounding upper oxic zone waters just below the ice (Gibson and Burton, 1996; Rankin et al, 1999). This salt exclusion drives the water mixing in the oxic zone of Ace Lake, with the anoxic zone remaining stagnant, and is responsible for lake stratification. The physical properties as well as the biology and function of Ace Lake have been investigated extensively for decades (Hand, 1980; Hand and Burton, 1981; Burch, 1988; Burke and Burton, 1988; Gibson and Burton, 1996; Rankin et al, 1997; Rankin, 1998; Bell and Laybourn-Parry, 1999; Rankin et al, 1999; Laybourn-Parry et al, 2005; Madan et al, 2005; Powell et al, 2005; Ng et al, 2010; Lauro et al, 2011; Laybourn-Parry and Bell, 2014). Only two of these studies have employed metagenomic data for the analysis of Ace Lake (Lauro et al, 2010) and its most abundant microbe (Ng et al, 2010).

A few studies have reported seasonal analysis of Ace Lake, mainly focusing on phytoplankton, bacteria, viruses or the physical structure and chemical composition of the lake (Hand and Burton, 1981; Burch, 1988; Gibson and Burton, 1996; Bell and Laybourn-Parry, 1999; Rankin et al, 1999; Laybourn-Parry et al, 2005; Madan et al, 2005). These studies used inverted or epifluorescence microscopy cell counts to determine microbial biomass and abundances (Bell and Laybourn-Parry, 1999; Laybourn-Parry et al, 2005; Madan et al, 2005). The level of photosynthetically active radiation (PAR) experienced at the surface of Ace Lake in different seasons and its attenuation with depth indicated that the presence/absence of snow and ice can affect light availability in the lake water column (Hand and Burton, 1981; Burch, 1988). The chemical composition of Ace Lake has been shown to vary with depth and season (Burch, 1988; Gibson and Burton, 1996; Bell and Laybourn-Parry, 1999; Rankin et al,

1999). The abundances of phytoplankton identified in the Upper oxic zone of Ace Lake being high in summer and low in winter (Burch, 1988; Bell and Laybourn-Parry, 1999; Laybourn-Parry et al, 2005). Moreover, these phytoplankton showed niche adaptation to different depths of the Upper zone (Burch, 1988). In Ace Lake Upper zone, autotrophic bacteria are prevalent in summer, whereas heterotrophic bacteria are more abundant in winter than summer (Bell and Laybourn-Parry, 1999; Laybourn-Parry et al, 2005). Overall bacterial biomass in the Upper zone of Ace Lake decreases in winter through to spring and then recovers in summer (Madan et al, 2005). Contrarily, virus-like particle counts, probably indicating viral abundance, show no seasonal variation in Ace Lake (Madan et al, 2005). Here, a time-series of metagenomes sampled from Ace Lake over a period of 10 years between 2006 and 2015, including summer and winter samples, were analysed using an in-house pipeline developed for analysis of IMG-annotated metagenomes (Chapter 2).

3.1.1 Ace Lake metagenomes

Ace Lake sample collection, DNA extraction, and metagenome sequencing, assembling, and annotation are described in Chapter 2 section 2.1.1. Water samples were collected from the Ace Lake in 2006, 2008, 2013, 2014, and 2015, including summer and winter samples, from the surface and six depths of the lake. The sampling depths included three depths from the upper oxic zone of Ace Lake (referred to as Upper 1, Upper 2, Upper 3), one from the oxycline/halocline (referred to as Interface), and three from the lower anoxic zone (referred to as Lower 1, Lower 2, Lower 3). However, winter samples were not collected from the Lower zone of Ace Lake, as sampling in Antarctic winter was logistically challenging. Additionally, the metagenomes used for the analysis of Ace Lake were assembled using the metaSPAdes assembler, as opposed to the Megahit assembler (Chapter 2 section 2.3.1.4). All metagenomes were annotated by JGI's IMG system (Appendix A: Table A1).

3.1.2 Aims

The main aim was to analyse the time-series metagenomes from Ace Lake, to assess seasonal variation in the biodiversity of the lake and examine the functional potential of its microbial population. The Ace Lake metagenomes were assessed using the in-house Cavlab pipeline v4 (Appendix C). Additionally, the most prominent OTU bins (≥1% relative abundance) identified in the Ace Lake metagenomes were refined and analysed

using various software discussed in Chapter 2. The MetaBAT-generated MAGs were used to verify OTU taxonomy, among other methods (Chapter 2).

A specific aim was to analyse the viruses identified in the Ace Lake metagenomes from different lake depths (upper, interface, lower) and seasons (summer, winter, spring). The association between changes in virus and potential host (especially phototrophic host) abundances were also explored, to determine the effects of viral predation *vs* light availability on host abundance. For this purpose, catalogues of Antarctic viral contigs (referred to as Antarctic virus catalogue hereafter), nucleocytoplasmic large DNA virus contigs (NCLDV; referred to as Antarctic NCLDV catalogue hereafter), virophage contigs (referred to as Antarctic virophage catalogue hereafter), and an IMG/VR spacer database (referred to as spacer database hereafter) generated from the Antarctic datasets, including the Ace Lake metagenomes, were used. A list of viral contigs representing complete, circular phage genomes (referred to as complete phage catalogue hereafter) in metagenomes from Antarctic lakes, including Ace Lake, were also used.

3.2 Methods

3.2.1 Taxonomic classification, abundance calculation, and functional potential analyses using Cavlab pipeline v4

The taxonomic diversity and functional potential of the Ace Lake metagenomes (Appendix A: Table A1) were analysed using the in-house Cavlab pipeline v4 (Appendix C). In the pipeline, the data in the Phylodist file were used to prepare a contig taxonomy file containing contig IDs and their length, read depth, and predicted taxonomy (Chapter 2 section 2.2.2.5). The contig taxonomy file only included contigs that contained predicted proteins with corresponding taxonomic assignments in the Phylodist file. Furthermore, the data in the contig taxonomy file were used to calculate the OTU abundances by summing coverages of the contigs (contig length × contig read depth) assigned to the OTU in a metagenome. All metagenome contigs that could not be assigned a taxonomy were referred to as unassigned contigs and the sum of their coverages was referred to as unassigned contig abundance in a metagenome. The total metagenome abundance was calculated by summing the coverages of all contigs in a metagenome, including all contigs assigned to an OTU as well as all unassigned contigs. PRIMER v7 software was used to calculate the relative abundances as

percentages, by dividing the OTU or unassigned abundances with total metagenome abundance (Chapter 2 section 2.2.7). All relative abundances mentioned in the chapter were calculated relative to the total metagenome abundances, unless otherwise specified, therefore, relative abundances from a metagenome were comparable. The relative OTU abundances were calculated by:

$$\text{Relative OTU abundance (\%)} = \frac{\sum_{\text{OTU}}(\text{ContigLength} \times \text{ContigReadDepth})}{\sum_{\text{Metag}}(\text{ContigLength} \times \text{ContigReadDepth})} * 100$$

Formula (1), where the numerator denotes the absolute abundance of an OTU in a metagenome, calculated by summing the coverages of contigs assigned to the OTU. The denominator represents the total abundance in a metagenome (Metag) calculated by summing the coverages of all contigs in the metagenome.

For comparative analysis of OTU abundances from the 120 metagenomes, the OTU abundance files were merged using a MetaPhlAn2 python script (Chapter 2 section 2.2.2.2). Among the OTUs identified in the Ace Lake metagenomes, OTUs with relative abundance ≥1% in at least one metagenome (referred to as abundant OTUs hereafter) were considered for further studies. Additionally, peak relative abundance of an OTU referred to its highest relative abundance in metagenomes from a depth, season, filter fraction, or all metagenomes, depending on how it is described in the chapter.

Functional potential analysis of Ace Lake was performed using the data in the metagenome COG and KEGG files from all Ace Lake metagenomes. The data in the metagenome COG files, i.e., protein IDs and their COG number annotations, were compared against a COG conversion database to generate COG category abundances by summing the read depths of contigs corresponding to predicted genes assigned to the COG category (Chapter 2 section 2.2.3.2). The data in the metagenome KEGG files, i.e., protein IDs and their KO number annotations, were used to generate abundances of specific KO numbers by summing the read depths of contigs corresponding to predicted genes assigned to the KO numbers (Chapter 2 section 2.2.3.2). These KO number abundances were then summed/averaged to calculate the abundances of specific pathways/enzymes (Chapter 2 section 2.2.3.2). The COG category and specific pathway/enzyme abundances were normalised prior to functional potential analysis, using the formula.

$$\frac{\text{COG or Pathway/enzyme abundance}}{\sum_{\text{Protein}}(\text{Contig read depth})} \times \left(\frac{\sum_{all}\{\sum_{\text{Protein}}(\text{Contig read depth})\}}{120}\right)$$

Formula (2), where, the left-side numerator denotes the abundance of a COG category or a pathway/enzyme. The left-side denominator represents the total protein abundance in a metagenome and was calculated by summing the read depths of contigs corresponding to all proteins in a metagenome. The right-side denotes the average of total protein abundances from the 120 Ace Lake metagenomes.

The abundant OTUs that probably contributed toward specific pathways were also identified during KEGG analysis. For this, a list of KO numbers and their corresponding pathway/enzyme was prepared and parsed using a python script, to extract the protein IDs associated with the specific KO numbers. The protein IDs were further used to select the corresponding contig IDs, and the contig taxonomies were deduced from the contig taxonomy output of Cavlab pipeline v4 runs on the metagenomes.

3.2.2 OTU bin refinement, taxonomy verification, and preparation of high-quality OTU bins

The abundant OTUs identified in the Ace Lake metagenomes were studied by preparing their OTU bins, which were composed of contig sequences that were assigned to the OTUs. The OTU bins were refined using RefineM v0.0.24 as described in Chapter 2 section 2.2.4.1. After bin refinement, the OTU bins with sufficient number of genes for functional potential analysis were selected for the preparation of high-quality bins. The contigs of these selected OTUs were further matched against MetaBAT-generated MAGs to verify their taxonomic composition. Note that the RefineM output as well as the MetaBAT MAGs used GTDB for taxonomic classification of bacteria and archaea. The taxonomies of the refined OTU bins were also verified by assessing their ANI using pyani and SSU rRNA gene identity to their closest related reference genomes from NCBI (Chapter 2 section 2.2.4.2). The high-quality OTU bins were prepared based on their RefineM output, matches to MetaBAT MAGs, ANI, and SSU rRNA gene identity. Some of the refined OTUs were merged to higher taxa levels due to similar taxonomic composition (Chapter 2 section 2.3.2). Some of the refined OTUs were merged to a higher taxa level and then split to lower taxa levels because the OTUs had matches to similar taxa. For example, five verrucomicrobial species-level OTUs, namely *Coraliomargarita akajimensis*, *Chthoniobacter flavus*, *Haloferula* sp. BvORR071, *Prosthecobacter debontii*, and *Rubritalea squalenifaciens*, were first merged together as *Verrucomicrobia* and then split to five genus-level OTUs, namely *Verrucomicrobia* SW10, *Verrucomicrobia* UBA4506, *Verrucomicrobia* BACL24,

Verrucomicrobia Arctic95D-9, and *Haloferula*. Additionally, some OTUs such as a *Parcubacteria* were simply split into lower taxa level OTUs to which they had matches; the *Parcubacteria* bin was split into six individual OTUs. A higher taxa level name, such as phylum, order, or family, was added before the alphanumeric genus names of refined OTUs for context. As a result, 45 high-quality bacterial and archaeal OTU bins along with a eukaryal and five algal virus bins were prepared from the Ace Lake metagenomes. The relative abundances of the high-quality OTUs were recalculated in metagenomes from specific depths in which their abundance was originally high. All low abundance OTUs (relative abundance <1%) as well as all low-quality OTUs were combined to higher taxa levels as 'other' bacteria, archaea, eukarya, and viruses using PRIMER v7.

3.2.3 Ace Lake metadata collection from various seasons and lake depths

The physical characteristics of Ace Lake, such as lake depth, salinity, temperature, dissolved oxygen concentration, and ice cover thickness, were measured during sample collection in summer, winter and spring (Appendix I). However, lake temperature and dissolved oxygen concentration measures could not be gathered for all time periods of sample collection. For the statistical analysis of the Ace Lake metagenomes, additional environmental factor measures, such as maximum and minimum air temperature, sunlight hours, and maximum wind velocity, were procured from the Australian Antarctic Data Centre (AADC) for Davis Station in Antarctica (Appendix I). Sunlight hours referred to the number of hours in a day the sun shines brightly, with sunlight being brighter than a specified threshold and without being obstructed by a cloud cover; it was calculated as bright sunshine hours using a Campbell-Stokes recorder. The daylength, i.e., the number of hours in a day the sun is above the horizon, was also used as an environmental factor for the analysis of Ace Lake metagenomes; the data were gathered for Davis Station from a web service (https://www.timeanddate.com).

3.2.4 Statistical analyses

3.2.4.1 Assessing alpha diversity and OTUs contributing to seasonal variation

For the statistical analysis of OTU abundances in 120 Ace Lake metagenomes, PRIMER v7 software was used. In PRIMER v7, the genus to domain level classifications of each OTU were used as indicators for the grouping of taxa variables, whereas sample collection date, lake depth, filter size, and season name were added as

factors for the grouping of metagenome samples. The relative OTU abundances were square root transformed and used to generate a resemblance matrix of sample similarities using the Bray-Curtis similarity measure. The transformed data were also used for SIMPER (similarity percentage) analysis, to identify OTUs with highest contribution to similarity between metagenome samples from a season and dissimilarity between metagenome samples from different seasons. The alpha diversity of all metagenomes was measured using the Simpson's index of diversity in Primer v7 (Chapter 2 section 2.2.7).

3.2.4.2 Assessing relationship between OTU abundance variation and changes in season

The relationship between the changes in the relative abundances of OTUs in the metagenome samples and the variation in certain predictor variables was explored using a distance-based linear model (distLM) in PRIMER v7. Environmental factors such as lake depth, lake salinity, air temperature, maximum wind velocity, daylength, and sunlight hours were considered for this purpose. Of these factors, monthly average values were calculated for air temperature, maximum wind velocity, daylength, and sunlight hours by calculating the mean of the values from a month, which were used as predictor variables along with lake depth and salinity. All predictor variable values were normalised in Primer v7 before analysis. Lake temperature could not be used as a predictor variable because it was not measured for all sampling periods. The distLM analysis was performed on the resemblance matrix of sample similarities and the predictor variables using a step-wise variable selection procedure with an adjusted R^2 fitness measure. The output was represented on a dbRDA plot.

3.2.4.3 Determining associations between specific OTUs, or virus and host

The correlation between certain OTU abundances, or read depths of viral contigs and host marker genes, was calculated using the Pearson Product Moment Correlation measure and the statistical significance of the correlation was assessed using the Analysis of Variance regression measure in Data Analysis Tools of Microsoft Excel.

3.2.5 Unassigned data analyses

The relative abundances of the unassigned contigs from the Ace Lake metagenomes were calculated using Formula (1) described in section 3.2.1. The genetic composition

of the unassigned contigs with relative abundance ≥1% (referred to as abundant unassigned contigs hereafter) was assessed to help identify their taxonomic affiliation. Furthermore, these abundant unassigned contigs were compared against the Antarctic virus catalogue to identify their corresponding viral cluster or singleton, if any. Of the abundant unassigned contigs, five contigs showed presence of *cas* genes and four contigs contained restriction-modification (R-M) system genes; both groups were investigated (section 3.2.6.5).

The gene annotations of the unassigned contigs of length ≥1 kb were manually analysed using the data in the IMG protein name file. Of the total gene annotations on these unassigned contigs in each metagenome, the number of genes associated with viruses, transposases, transfer RNAs (tRNAs), 16S rRNA, 18S rRNA, and hypothetical proteins were counted to assess the overall genetic composition of the unassigned contigs. The *16S* and *18S rRNA* genes were also aligned against the NCBI-nr nucleotide database using the blastn mode of BLAST+ v2.9.0, and their domain-level taxonomy was deduced by manually assessing the best alignments. The domain-level relative abundances associated with the unassigned contigs were estimated based on the coverages of the contigs containing the *16S* and *18S rRNA* genes, using Formula (1) (section 3.2.1).

The unassigned contigs were also parsed through VirSorter v1.0.3 (Roux et al, 2015), to identify potential viral and prophage contigs. For this, all unassigned contigs from the Ace Lake 2006 metagenomes and all unassigned contigs >10 kb length from the Ace Lake 2008 and 2013-2015 metagenomes were assessed, along with 1-10 kb length unassigned contigs from Ace Lake 2008 and 2013-2015 metagenomes with relative abundance ≥0.1% or read depth ≥200. The unassigned contigs that were confidently predicted to be viruses (VirSorter categories 1 and 2) or prophages (VirSorter categories 4 and 5) were used to calculate the relative abundance of viruses among the unassigned contigs from each metagenome.

3.2.6 Viral analyses

The Antarctic virus catalogue contained 71,689 viral contigs of length >5 kb, identified from 309 Antarctic metagenomes, including the 120 Ace Lake metagenomes, using a previously described method (Páez-Espino et al, 2016). The Antarctic NCLDV catalogue contained 2,296 NCLDV contigs, whereas the Antarctic virophage catalogue

contained 69 virophage contigs, identified using methods reported before (Páez-Espino et al, 2019b; Schulz et al, 2020). All three viral catalogues contained the viral cluster (group of similar viral contigs) or singleton designations of the viral contigs. The spacer database contained a list of genome- and metagenome-associated CRISPR spacers, the host contigs on which they were identified, and the matches of the CRISPR spacers to the viral contigs in the Antarctic virus catalogue; all of these data were generated using a previously described method (Páez-Espino et al, 2019a). The complete phage catalogue contained 516 viral contigs representing complete genomes of circular phage identified from various Antarctic lake systems.

3.2.6.1 Analysis of potential GSB viruses

The contig IDs of the *Chlorobium* OTU bin contigs were compared against the contig IDs in the IMG CRISPR annotation files (hereafter referred to as metagenome CRISPR files), which contained CRISPR spacer and repeat sequences found on the contigs, to identify any CRISPR spacers associated with the Ace Lake *Chlorobium* (Figure 3.1). The *Chlorobium*-associated CRISPR spacer sequences were then aligned against all contigs from the 120 Ace Lake metagenomes using the 'megablast' option of BLAST+ v2.6.0, with e-value $\leq 10^{-3}$ and $\geq$97% alignment identity. The metagenome contigs with matches to the *Chlorobium* spacers were compared against the Antarctic virus catalogue to identify the potential *Chlorobium* virus contigs and their corresponding viral clusters and/or singletons (Appendix H: Table H1). For verification of the hosts of the potential *Chlorobium* viruses, the spacer matches to these viral contigs in the spacer database were assessed and the host contig taxonomy was determined (section 3.2.1).

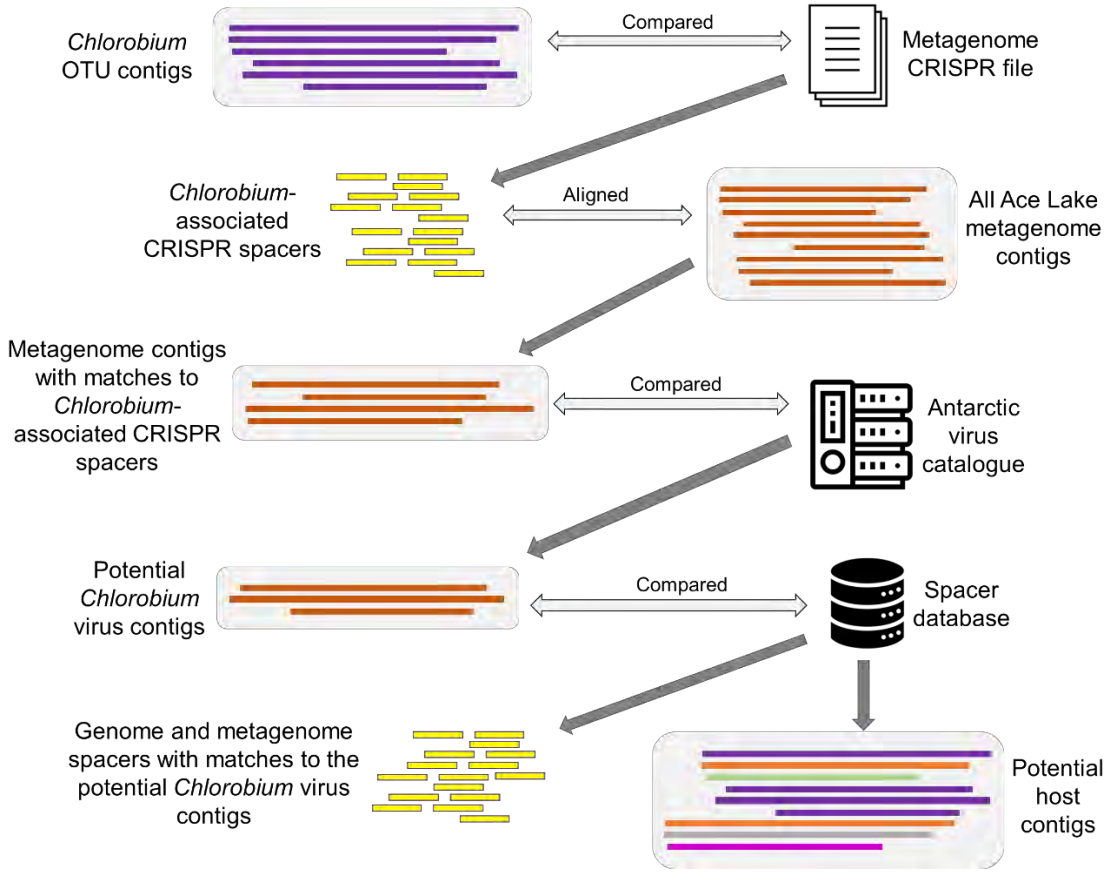


Figure 3.1 Identifying potential *Chlorobium* viruses. The schematic shows the methods used for the identification of potential *Chlorobium* viruses (section 3.2.6.1). Metagenome CRISPR file refers to the IMG CRISPR annotation file. Antarctic virus catalogue and spacer database are described in section 3.2.6. All sequence alignments (Aligned) were performed using BLAST+, whereas contig IDs were used for comparison (Compared) between queries and databases. The icons used in the figure were taken from The Noun Project website (https://thenounproject.com/).

The contigs belonging to the potential *Chlorobium* virus clusters and singletons were also matched against all Ace Lake metagenome contigs using 'blastn' mode of BLAST+ v2.9.0, with e-value $\leq 10^{-4}$ and $\geq$90% alignment identity, to include any additional viral contigs that were not part of the Antarctic virus catalogue, and to assess the similarity between the contigs belonging to the *Chlorobium* virus clusters and singletons (Appendix H: Table H1). The average read depth of specific marker genes from *Chlorobium*, namely *16S rRNA*, recombinase A and *fmoA*, and the read depths of its viral contigs were used to analyse the virus-host abundance correlation (section 3.2.4.3).

3.2.6.2 Analysis of potential *Synechococcus* viruses

A cyanophage sequence (IMG taxon ID: 3300016486; contig: Ga0078900_115654) was assembled from a 0.1 µm-filter metagenome from the Ace Lake Upper 2 zone sampled in Dec 2006. The cyanophage sequence was matched against the Antarctic virus catalogue using the 'blastn' mode of BLAST+ v2.9.0, with e-value $\leq 10^{-3}$ and ≥99% alignment identity, to identify additional cyanophage sequences (Figure 3.2) (Appendix H: Table H1). The cyanophage sequences were also aligned against each other using the 'blastn' mode of BLAST+ v2.9.0 (e-value $\leq 10^{-4}$), to assess their sequence similarity. The spacer database was used to identify the probable host of the cyanophage by examining the matches of the cyanophage sequences to the spacers on the host contigs. The host contig taxonomy was determined using the method described in section 3.2.1. The correlation between the read depth of the cyanophage contigs and the average read depth of the marker genes (*16S rRNA* and recombinase A) from *Synechococcus*, the most abundant cyanobacteria in Ace Lake, was also calculated to explore probable virus-host relationship (section 3.2.4.3).

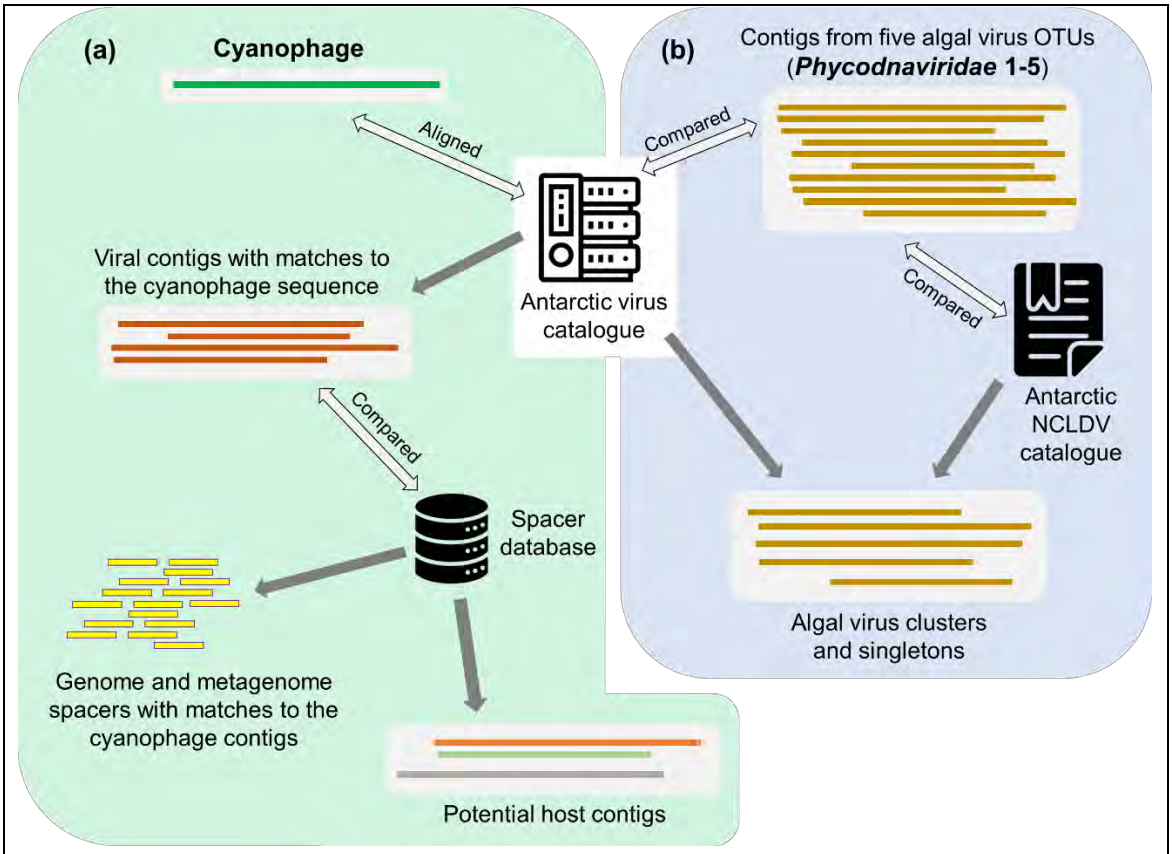


Figure 3.2 Analysis of a cyanophage and some algal viruses. The schematic shows the methods used for the analysis of a cyanophage (green background; section 3.2.6.2) and five abundant algal viruses, namely *Phycodnaviridae* 1, 2, 3, 4, and 5 (blue background; section 3.2.6.3). Antarctic virus catalogue, Antarctic NCLDV catalogue, and spacer database are

described in section 3.2.6. All sequence alignments (Aligned) were performed using BLAST+, whereas contig IDs were used for comparison (Compared) between queries and databases. The icons used in the figure were taken from The Noun Project website (https://thenounproject.com/).

3.2.6.3 Analysis of algal viruses

The metagenome contigs assigned to the five *Phycodnaviridae* OTUs (*Phycodnaviridae* 1–5) were compared against the viral contigs in the Antarctic virus catalogue as well as the Antarctic NCLDV catalogue, to identify all virus/NCLDV clusters and singletons associated with the *Phycodnaviridae* 1–5 OTUs (Figure 3.2). Correlation analysis was performed between the relative abundance of the Ace Lake green alga OTU *Micromonas* and the relative abundances of *Phycodnaviridae* 1–5 OTUs (section 3.2.4.3).

3.2.6.4 Analysis of viral contigs representing complete genomes

The viral contigs in the complete phage catalogue were compared with the viral contigs in the Antarctic virus catalogue to identify the viral cluster or singleton designations of the complete virus genomes, if any. The viral contigs were also compared against the Antarctic NCLDV and virophage catalogues to assess whether any of the complete virus genomes represented NCLDVs or virophages, respectively. Additional data for the viral contigs representing complete genomes, such as contig length, read depth, GC content, gene count, and gene function, were collected from the Antarctic metagenome data available on JGI's IMG/M website (https://img.jgi.doe.gov/cgi-bin/m/main.cgi). Based on the viral cluster or singleton designations, clade assignment, contig lengths, and GC content, the viral contigs in the complete phage catalogue identified from the Ace Lake metagenomes were grouped as distinct viruses (Appendix H: Table H1).

3.2.6.5 Analysis of viruses containing defence genes

The nine abundant unassigned contigs containing defence genes (section 3.2.5) were thoroughly analysed. The contigs were compared against the Antarctic virus catalogue and the complete phage catalogue to identify any viral clusters or singletons they might be associated with and to assess whether any of the contigs represented complete viral genomes, respectively. The contigs were also compared against the output of VirSorter analysis of the unassigned contigs (section 3.2.5). The abundant viral contigs and the contigs belonging to the viral clusters to which they belonged were further aligned

against each other using the 'progressive Mauve' mode of Mauve v2.4.0 using default parameters, to identify additional cluster contigs that might represent complete or nearly complete phage genome (Appendix H: Table H1). The viral cluster contigs were also compared against the spacer database to identify potential hosts.

The viral cluster contigs containing *cas* genes were aligned against the Ace Lake metagenome contigs using the blastn mode of BLAST+ v2.9.0, considering only 100% identity matches across 100% query alignment fraction, to include additional viral contigs that were not a part of the Antarctic virus catalogue. Additionally, CRISPR spacer arrays were identified in the viral contigs containing *cas* genes. Using the spacer database, the potential viral targets of these viral spacers were determined. Moreover, these viral spacer sequences were compared against the spacer sequences of its probable host, in search of any similarities between the two groups of spacers.

3.2.6.6 Analysis of abundant Ace Lake viral clusters

Among the viral contigs in the Antarctic virus catalogue, the contigs identified from the Ace Lake metagenomes were separated out and grouped into their viral cluster and singleton designations for analysis. The read depths of the viral contigs belonging to a viral cluster from a time period (including all depths and filter fractions) were summed. Viral clusters with summed read depths >4000 in at least one time period as well as singletons with read depths >4000 were considered to be abundant (referred to as abundant viral clusters or singletons hereafter) and were further analysed (Appendix H: Table H2). Additionally, abundant viral clusters were assessed to determine the Ace Lake depth from which most of the cluster contigs originated. The probable hosts (bacterial, archaeal, or eukarya) of the viral clusters or singletons were also discerned from the data in the Antarctic virus catalogue, which included some of the viral data (such as viral cluster or singleton designation and its probable host) available on IMG/VR (Páez-Espino et al, 2017) that had matches to the Antarctic viral contigs. In search of additional potential *Chlorobium* viruses, the spacer matches to the abundant viral clusters from the Ace Lake Interface and Lower zone were examined and their potential hosts were deduced using the data in the spacer database. The abundant viral clusters with predicted eukaryal hosts were also compared to the viral clusters with matches to the five *Phycodnaviridae* OTUs.

Read depth-based abundance correlation analysis was also performed between *Chlorobium* and the abundant viral clusters from the Ace Lake Interface and Lower zone as well as between *Synechococcus* and the abundant viral clusters from the Ace Lake Upper zone that had potential bacterial hosts. Relative abundance-based correlation analysis was performed between *Micromonas* and the abundant viral clusters from the Ace Lake Upper zone that had potential eukaryal hosts.

3.3 Results and discussion

3.3.1 Antarctic seasons: defining seasons in polar regions

Antarctica is the southern-most continent on Earth and contains the geographic South Pole. It is the coldest and driest continent with very little annual precipitation; the mean annual precipitation around Davis Station (68.577° S, 77.968° E) in East Antarctica ranged between 0.1 to 0.5 mm over a 10-year period from January 2006 to December 2015 (Davis Station data from AADC). Notably, the light cycle prevalent in the polar regions is very different from that experienced in the other non-polar cold regions, with 24 hours of sunlight available for a few weeks in summer when the sun does not set and no sunlight available for a few weeks in winter when the sun does not rise (Figure 3.3a).

For the in-depth time-series analysis of the Ace Lake in the Vestfold Hills, samples were collected over a 10-year period between 2006 and 2015, in January, February, July, August, October, November, and December. Therefore, it was important to reasonably define the seasons based on the environmental factors that indicate a change in season and might directly or indirectly affect the biodiversity and function of the lake, as opposed to defining seasons using a simple system of months (Table 3.1). The environmental factors studied included daylength (number of hours the sun was above the horizon; Figure 3.3a), sunlight hours (number of hours of bright sunlight was available without being obstructed by a cloud cover; Figure 3.3a), air temperature (Figure 3.3b), maximum wind velocity (Figure 3.3c), and lake ice cover thickness (Table 3.1; Appendix I). Daylength and sunlight hours indicate the availability of light, which can directly impact species diversity and abundance, especially in the upper oxic zone and oxycline of Ace Lake, where the phototrophic algae and bacteria thrive (Rankin et al, 1997; Rankin, 1998; Rankin et al, 1999; Powell et al, 2005; Ng et al,

2010; Lauro et al, 2011; Laybourn-Parry and Bell, 2014). For example, in summer, intense sunlight can cause photoinhibition and reduce the photosynthetic capacity of these microorganisms (Rankin et al, 1999; Powell et al, 2005; Cuvelier et al, 2017). As Ace lake is covered by ice for most of the year, the air temperature and wind likely do not directly affect the organisms living in the lake waters, except probably the lake surface microorganisms. However, they might be useful as indicators of change in season. Additionally, changes in air temperature can alter the thickness of the Ace Lake ice cover, which in turn can affect water mixing and light penetration in the lake as well as lake temperature, thereby impacting the lake biodiversity (Hand and Burton, 1981; Burch, 1988; Gibson and Burton, 1996; Rankin, 1998; Rankin et al, 1999; Laybourn-Parry and Bell, 2014).

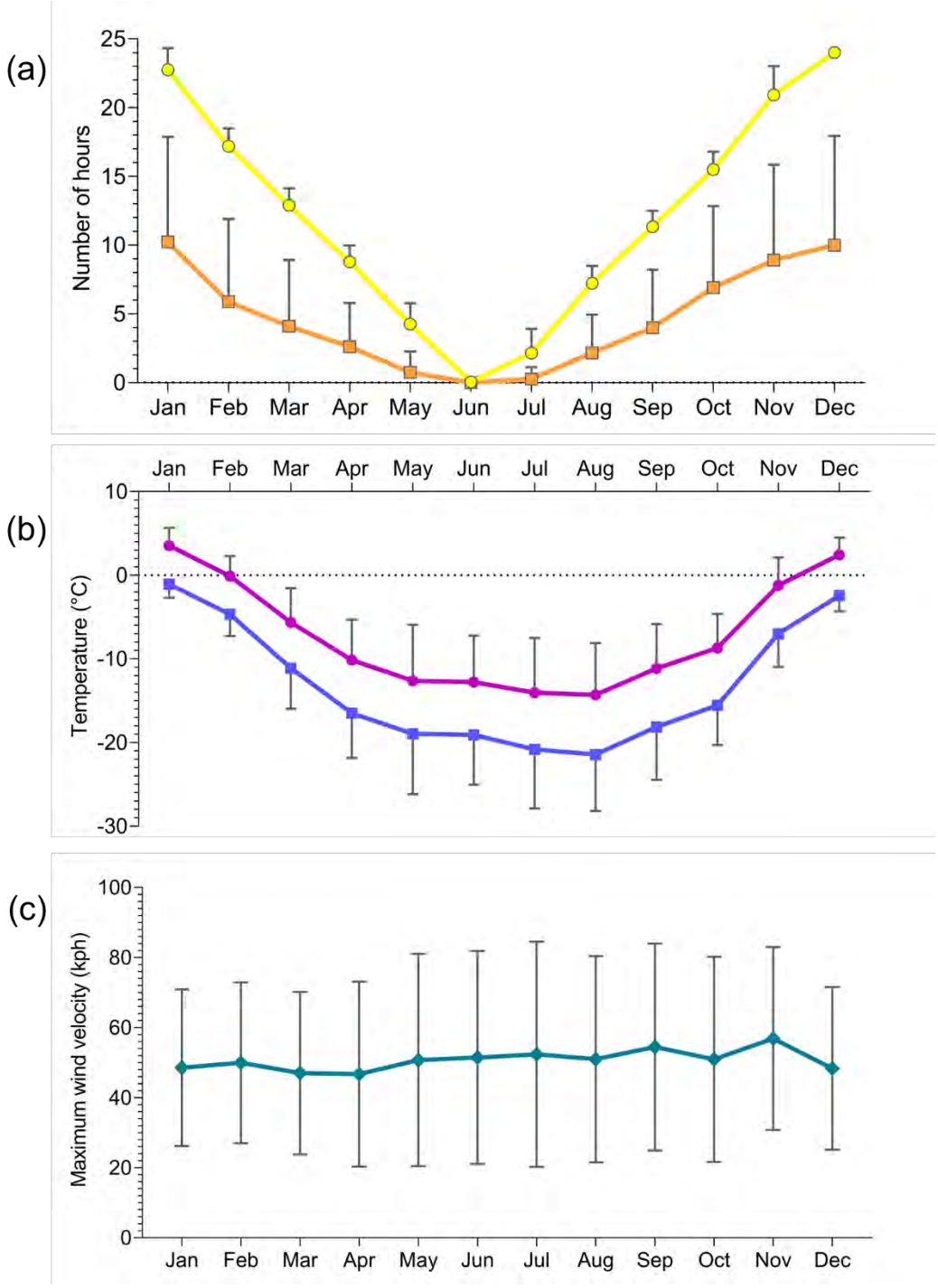


Figure 3.3 Environmental data recorded at Davis Station, East Antarctica. The line graphs show the mean of monthly average values of **(a)** daylength (yellow line) and sunlight hours (orange line); **(b)** maximum (purple line) and minimum (blue line) air temperature; and **(c)** maximum wind velocity (green line) recorded over a 10-year period from Jan 2006 to Dec 2015 (monthly average values described in section 3.2.4.2**).** Daylength data for Davis Station were taken from an online service (https://www.timeanddate.com), whereas the sunlight hours, air temperature, and wind velocity data gathered at Davis Station were taken from AADC. All error bars indicate both positive and negative standard deviations from the mean.

The daylength values measured at Davis Station were consistent over the 10-year period, with 24 h daylight from the end of November to mid-January and no daylight from the beginning of June to early July (Figure 3.3a). However, the number of bright sunshine hours (sunlight hours) varied throughout the year, with the variations being directly proportional to the availability of light (Figure 3.3a). For example, the sunlight hours showed no variations over the years in Jun when no daylight was available, but the variations steadily increased from Jun to Dec with the increase in daylength, fluctuating by as much as 8 h in Dec and Jan. Additionally, air temperature values showed seasonal variation, with lowest temperatures measured in Jul and Aug (-14 to -21 °C) and mostly positive temperatures measured in Dec and Jan (2 to 4 °C) (Figure 3.3b). The air temperature also fluctuated more in the colder months (varying by up to 7 °C) than in the warmer months (varying by ~2 °C). Wind velocity could not be used for defining season, as it showed no discernible pattern of change (Figure 3.3c). As Ace Lake is covered by ice for nearly 11 months a year, ice cover thickness could probably be used to define only the summer months, when the ice cover would be partially or completely melted (Table 3.1). It has also been previously noted that the Ace Lake ice cover is thickest in spring or early summer (Rankin et al, 1999). Among the environmental data collected during Ace Lake sampling, the maximum ice cover thickness was observed in October (~2 m) and November (1.75 m) with no visible ice melting, and so these two months could be defined as spring months based on ice cover thickness.

Eventually, based on light availability and air temperature, December and January were considered summer months, whereas July and August were considered winter months. However, the results of the environmental data analysis could not help with confidently categorising some of the sample collection months, such as October, November, and February (Table 3.1). Consequently, a more general season grouping was used to categorise all the sample collection months: December, January, and February as summer months; July and August as winter months; and October and November as spring months.

Table 3.1 Season description based on environmental data gathered during sample collection and at Davis Station, Antarctica. * The percentage light hours were calculated by dividing the number of hours of sunlight by 24 (number of hours in a day). The monthly average values were calculated as described in section 3.2.4.2. † Wind velocity did not show

any seasonal pattern and was not used for season classification. ‡ The cut-offs for season classification were selected based on some of the previously applied season classifications, which were used as references. For example, the winter cut-off of % light hours was selected as 30% keeping in mind that Jul and Aug were considered winter months by some of the previous seasonal studies on Ace Lake (Burch, 1988; Burke and Burton, 1988; Rankin et al, 1999; Laybourn-Parry and Bell, 2014).

Sample collection time period	Monthly average daylength in hours (% light hours)*	Monthly average sunlight hours (% light hours)*	Monthly average maximum air temperature (°C)	Monthly average minimum air temperature (°C)	Monthly average maximum wind velocity (km/h)†	Ice cover thickness
Dec 2006	24 (100%)	10 (42%)	2	-2	42	-
Nov 2008	21 (88%)	9 (38%)	-2	-7	67	1.75 m
Nov 2013	21 (88%)	13 (54%)	-1	-7	54	Covered by thick ice
Dec 2013	24 (100%)	10 (42%)	2	-3	44	Covered by thick ice, but melting
Feb 2014	17 (71%)	6 (25%)	1	-4	46	Half covered by ice
Jul 2014	2 (8%)	1 (4%)	-17	-24	37	Covered by thick ice
Aug 2014	7 (29%)	3 (13%)	-14	-21	62	>1 m
Oct 2014	15 (63%)	7 (29%)	-7	-13	49	~2 m
Dec 2014	24 (100%)	10 (42%)	3	-2	49	~1.8 m ice, beginning to melt

8 Jan 2015	23 (96%)	9 (38%)	3	-1	47	Covered in poor quality ice
27 Jan 2015	23 (96%)	9 (38%)	3	-1	47	No Ice
Probable season classification based on observed environmental data‡						
% Light hours*	≤30% Low light levels Winter.		31-80% Medium light levels Spring/Autumn.		>80% High light levels Summer.	
Air temperature	< -5 °C Very cold Winter.		-5 to 0 °C Cold Spring/Autumn.		>0 °C Comparatively warmer Summer.	
Ice cover thickness	Thick ice cover Winter.		Maximum ice cover Spring.		Ice cover melting or no ice cover Summer.	

3.3.2 Seasonal changes in Ace Lake environment

Ace Lake is a stratified lake with an upper oxic (Upper) zone separated from a lower anoxic (Lower) zone by an oxycline (Interface), which also coincided with the halocline and thermocline of the lake (Figure 3.4). Ace Lake environmental data collected between 2006 and 2014 showed that the lake salinity increased with lake depth and ranged from 3.6 to 4.2 % at the lowest lake depth. The lake temperature also increased with depth, reaching 3 to 5 °C at the thermocline, but then decreased with lake depth to 2–3 °C (Figure 3.4). The data also showed that the Ace Lake environment varied with changes in season, especially the oxycline, probably due to changes in the ice cover thickness. As summer recedes, the ice cover begins to form, and it thickens with approaching winter. Ice formation in Ace Lake causes salt exclusion into the Upper zone waters, just below the ice, which causes the Upper zone waters to mix via convection (Gibson and Burton, 1996; Rankin et al, 1999). The thickness of the ice cover directly governs the depth to which the water mixes. Therefore, in winter when the ice was thicker, the oxycline was possibly pushed deeper down the lake depth. However, little to no environmental changes were observed in the Lower zone of Ace Lake (Figure 3.4).

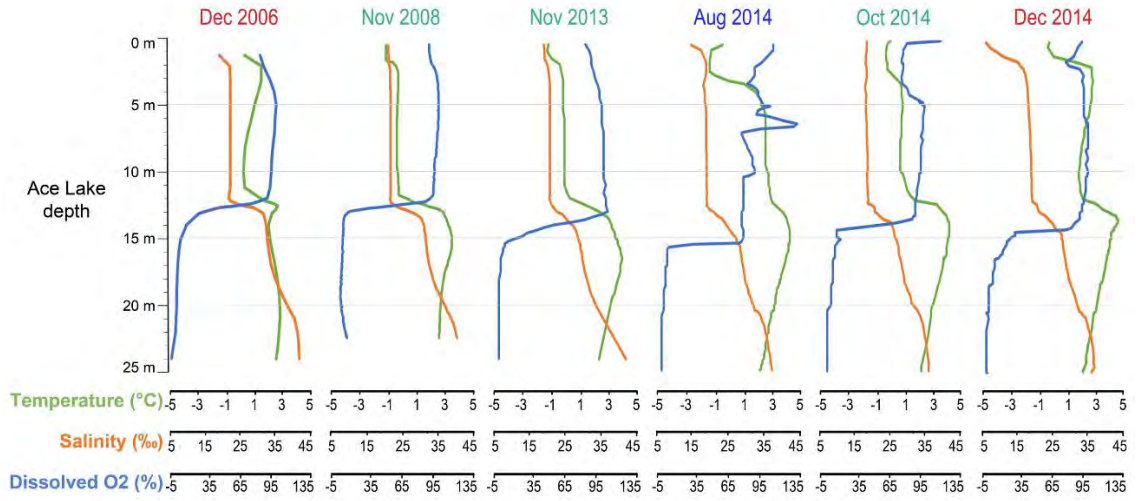


Figure 3.4 Ace Lake temperature, salinity, and dissolved oxygen profiles. The scatter plots show the changes in the Ace Lake environment factors, namely temperature (green line), salinity (orange line), and dissolved oxygen (blue line), from 2006 to 2014. The data were collected from three seasons (top x-axis: summer, red font; winter, blue font; spring, green font). The y-axis indicates increasing lake depth from top to bottom. The x-axis is shown separately for each factor. As different data collection devices were used for calculating dissolved oxygen content (YSI Sonde in 2006 and 2008; TOA WQC in 2013 and 2014), the data were normalized across sampling periods. Dissolved oxygen content data were not gathered during Dec 2013, Feb 2014, Jul 2014, and Jan 2015 sample collections, and are not shown here.

3.3.3 Ace Lake biodiversity

The taxonomy and abundance analysis of ~25 million contigs (20 Gbp) assembled from 120 Ace Lake metagenomes yielded 17,157 OTUs, of which 117 OTUs had relative abundance ≥1%. The abundant OTUs, along with 3.5 billion (0.5 Tbp) metagenome reads, were used for generating a total of 45 high-quality bacterial and archaeal OTU bins as well as a eukaryal bin and five algal virus bins.

3.3.3.1 Eukarya

OTUs belonging to the Eukarya domain were found to be abundant in the Upper zone of Ace Lake. A total of 508 eukaryal OTUs were identified in the 120 Ace Lake metagenomes, of which only four had relative abundance ≥1% in at least one metagenome. Of the four abundant eukaryal OTUs, only two yielded good quality bins after bin refinement using RefineM. The two OTUs were closely related to two *Micromonas*, namely *Micromonas pusilla* and *Micromonas commoda*, members of the

Mamiellaceae family. However, based on their ANI to respective reference genomes and their matches to similar MetaBAT MAGs, the two OTUs were merged to genus-level as *Micromonas* (Appendix G).

The *Micromonas* OTU contributed to most of the Eukarya relative abundance in the Upper zone of Ace Lake in summer and spring (Figure 3.5). Other green algae including *Py. gelidicola* and a *Mantoniella* have been reported in the Ace Lake Upper zone (Rankin et al, 1999; Lauro et al, 2011). However, neither of these algae were detected in the Ace Lake metagenomes studied here. It is possible that this difference is due to the different approaches taken for taxonomic classification of these organisms, especially considering that *Mantoniella*, which is also a member of the *Mamiellaceae* family like *Micromonas*, was identified in the Ace Lake data from 2006.

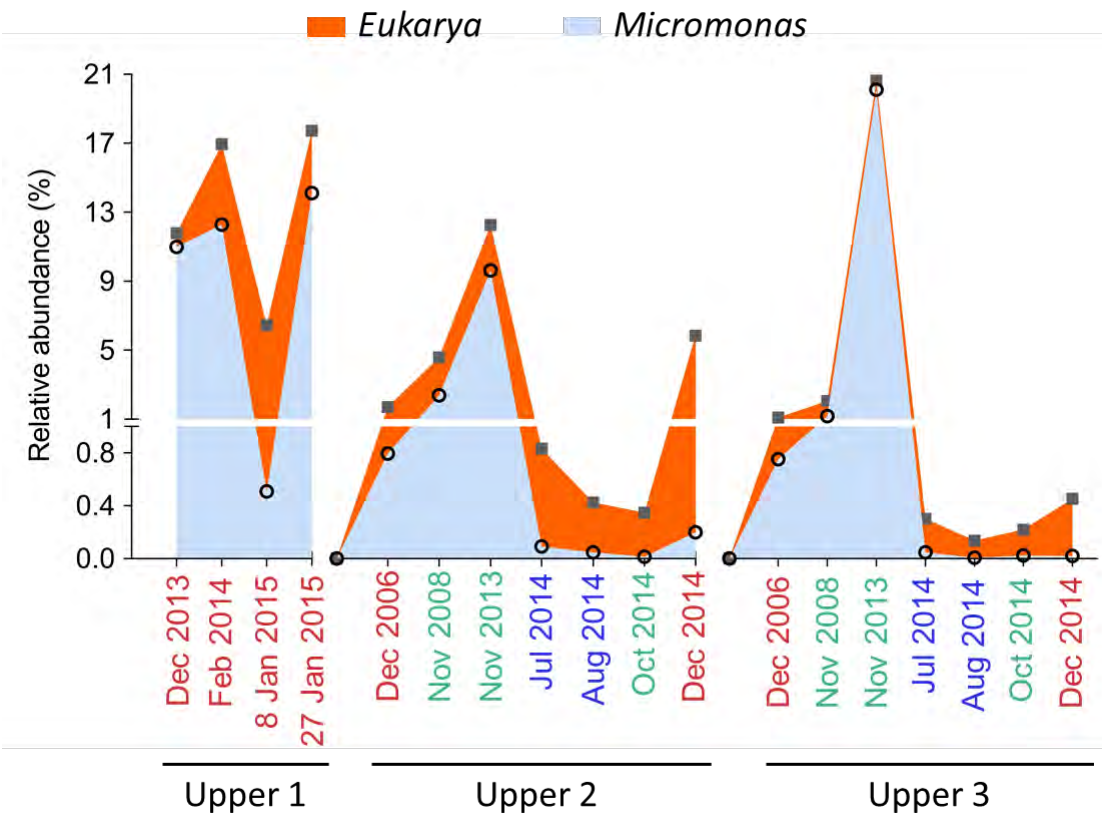


Figure 3.5 Relative abundances of eukaryal OTUs in the Upper zone of Ace Lake. The area graph shows the combined relative abundances of all eukaryal OTUs (orange area cover) as well as *Micromonas* (blue area cover) in the 0.8–3 µm-filter Ace Lake metagenomes. As the figure is an area graph, it indicates the amount of Eukarya relative abundance that is contributed by the *Micromonas* genus. For example, in Upper 3 Nov 2013 metagenome, nearly all of the Eukarya abundance is represented by *Micromonas*. The x-axis shows the Ace Lake depth zones (Upper 1, 2, 3) as well as the sample collection time periods, including data from three seasons

(summer, red font; winter, blue font; spring, green font). The y-axis was split to show an expanded view of relative abundances below 1%. All Ace Lake surface (Upper 1) samples were from summer.

3.3.3.2 Bacteria

In the Ace Lake metagenomes, 84% of the OTUs (14,387 out of 17,157 OTUs) belonged to the Bacteria domain and were dispersed throughout the lake system. Of these bacterial OTUs, only 96 OTUs had relative abundance ≥1% and were used to generate 43 high-quality bacterial OTU bins. The OTUs included members of *Actinobacteria* (2), *Alphaproteobacteria* (4), *Atribacteria* (1), *Bacteroidetes* (11), *Balneolaeota* (1), *Betaproteobacteria* (3), *Chlorobi* (1), *Cloacimonetes* (1), *Cyanobacteria* (1), *Deltaproteobacteria* (5), *Gammaproteobacteria* (6), *Planctomycetes* (1), *Tenericutes* (1), and *Verrucomicrobia* (5).

These OTUs were found to be abundant at particular depths of Ace Lake (Upper, Interface, or Lower zone depths), indicating their niche specificity (Figure 3.6), which has also been previously reported (Rankin et al, 1999; Lauro et al, 2011). For example, some of the *Alphaproteobacteria* (*Loktanella*), *Bacteroidetes* (*Algoriphagus, Leadbetterella*, *Nonlabens*, *Saprospiraceae* sp., *Polaribacter*), and *Betaproteobacteria* OTUs (*Hydrogenophaga*, *Burkholderiaceae* MOLA814) were abundant only in the metagenomes from the Ace Lake surface (Upper 1) (Figure 3.6). Of these, only *Burkholderiaceae* MOLA814 had good matches to a previously known species genome, *Betaproteobacteria* bacterium MOLA814 (isolated from a cold, marine environment — Beaufort Sea, Arctic Ocean), with 100% SSU rRNA gene identity and 98% ANI across 94% alignment fraction (Appendix G). *Leadbetterella* had matches to the *Cytophagales* bacterium TFI 002 genome, but with 91% SSU rRNA gene identity and 71% ANI across only 19% alignment fraction, suggesting that the two organisms were quite different. All other OTUs abundant in Upper 1 also had matches to reference genomes, but with low ANI (<85%) and no SSU rRNA gene matches, probably due to incomplete bins lacking SSU rRNA genes (Appendix G). Therefore, these OTUs could not be assigned a species-level taxonomy.

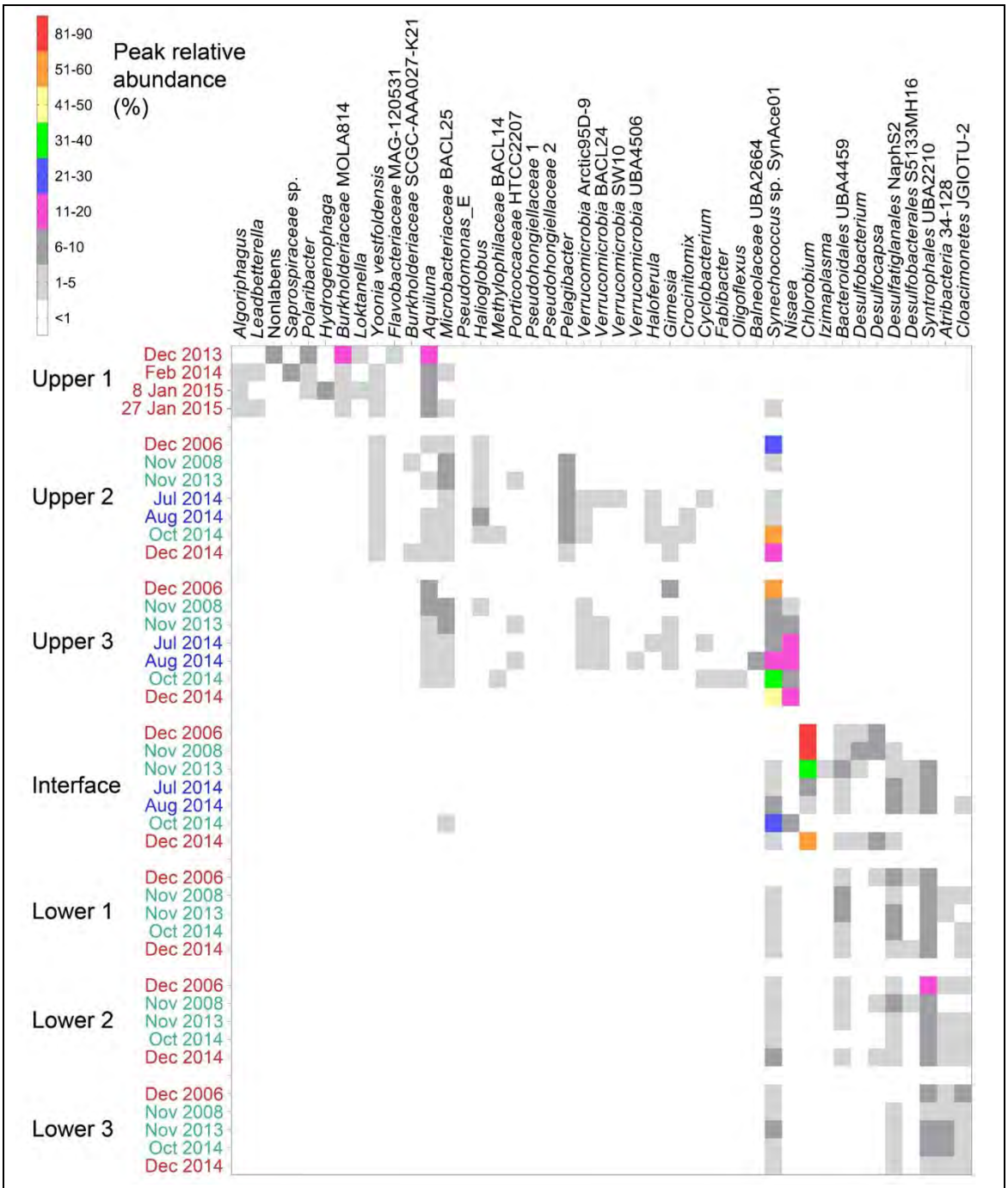


Figure 3.6 Relative abundances of bacterial OTUs throughout the Ace Lake. The heat map shows the peak relative abundance of bacterial OTUs from a depth and time period, i.e., the highest relative abundance of an OTU among the three filter fractions from a depth and time period. The OTUs shown in the figure represent the abundant bacterial OTUs for which high-quality bins were generated and their abundances recalculated in metagenomes from depths where their abundances were originally high (section 3.2.2). The y-axis represents increasing lake depth from top to bottom — Upper 1 to Lower 3, and shows data collected from summer (red font), winter (blue font), and spring (green font). All Ace Lake surface (Upper 1) samples

were from summer. Winter data were not gathered from the Lower zone of Ace Lake due to logistics issues during sample collection.

Apart from the microorganisms identified only in Upper 1, the Upper zone of Ace Lake supported a variety of microbes including members of *Actinobacteria* (*Aquiluna*, *Microbacteriaceae* BACL25), *Alphaproteobacteria* (*Nisaea*, *Pelagibacter*, *Yoonia*), *Bacteroidetes* (*Crocinitomix*, *Cyclobacterium*, *Fabibacter*, *Flavobacteriaceae* MAG-120531, *Oligoflexus*), *Balneolaeota* (*Balneolaceae* UBA2664), *Betaproteobacteria* (*Burkholderiaceae* SCGC-AAA027-K21), *Cyanobacteria* (*Synechococcus*), *Gammaproteobacteria* (*Halioglobus*, *Methylophilaceae* BACL14, *Porticoccaceae* HTCC2207, *Pseudohongiellaceae* 1, *Pseudohongiellaceae* 2, *Pseudomonas*_E), *Planctomycetes* (*Gimesia*), and *Verrucomicrobia* (*Haloferula*, *Verrucomicrobia* Arctic95D-9, *Verrucomicrobia* BACL24, *Verrucomicrobia* SW10, *Verrucomicrobia* UBA4506) (Figure 3.6). Of these, *Synechococcus* was found to be abundant throughout the Ace Lake, whereas *Nisaea* was abundant in Upper 3 as well the Interface of Ace Lake. *Synechococcus* was one of the two most abundant bacteria in Ace Lake and had good matches to the reference genome of *Synechococcus* sp. SynAce01 (also isolated from Ace Lake), with 99.9% *16S rRNA* gene identity and 99% ANI across 97% alignment fraction (Appendix G; discussed in Chapter 4). This cyanobacterium is abundant in Ace Lake, especially at depths just above the oxycline (Rankin et al, 1997; Rankin, 1998; Rankin et al, 1999; Powell et al, 2005; Lauro et al, 2011). Additionally, a *Yoonia* OTU had good matches to the reference genome of *Yoonia vestfoldensis* SKA53 (previously isolated from Ace Lake, Antarctica; Van Trappen et al, 2004), with 99.9% SSU rRNA gene identity and 93% ANI across 89% alignment fraction. Some of the other Upper zone OTUs also had good SSU rRNA gene matches (≥99%) to their reference genomes, suggesting that they could be different strains of the reference species. However, their ANI was usually low (<90%), either because the OTU bins were incomplete or probably because the microbes had distinct genomes compared to the reference species (Appendix G). Therefore, these OTUs could not be assigned a species-level taxonomy.

At the Ace Lake Interface, GSB belonging to the *Chlorobium* genus were found to be abundant (Figure 3.6); it was also the most abundant microorganism in Ace Lake and has been reported before (Rankin et al, 1999; Ng et al, 2010; Lauro et al, 2011). The Ace Lake *Chlorobium* had 99% *16S rRNA* gene identity to *C. phaeovibrioides* DSM

265 reference genome, but only 85% ANI across 85% alignment fraction, suggesting that it was probably a different species to the reference (Appendix G; discussed in Chapter 5).

The Lower zone of Ace Lake also supported a variety of bacterial populations, mostly including members of *Deltaproteobacteria* (*Desulfobacterium*, *Desulfocapsa*, *Desulfatiglanales* NaphS2, *Desulfobacterales* S5133MH16, *Syntrophales* UBA2210), *Tenericutes* (*Izimaplasma*), and *Bacteroidetes* (*Bacteroidales* UBA4459), along with members of candidate phyla such as *Atribacteria* (*Atribacteria* 34-128) and *Cloacimonetes* (*Cloacimonetes* JGIOTU-2) (Figure 3.6). Of these, the candidate phyla microbes were mostly prevalent in the deeper Lower zone depths, especially Lower 2 and 3, whereas the *Deltaproteobacteria*, *Bacteroidetes*, and *Tenericutes* OTUs were found to be more abundant in the Interface and Lower 1 metagenomes (Figure 3.6). All Lower zone OTUs had matches to reference genomes, however they could not be assigned a species-level taxonomy, either because the OTU bins were incomplete and lacked SSU rRNA genes for comparison or their SSU rRNA gene identities and ANI were low (Appendix G).

3.3.3.3 Archaea

The archaea identified in Ace Lake were prevalent in the Lower zone of the lake and were also found at the Interface in the winter months (Jul and Aug 2014), with most of their abundance contributed by members of *Euryarchaeota* (Figure 3.7). A total of 445 archaeal OTUs were identified in the Ace Lake metagenomes, of which only four had relative abundance ≥1% and were used for generating high-quality archaeal OTU bins. However, after refinement with RefineM, only two high-quality archaeal OTU bins were generated, namely *Methanomicrobiaceae* 1 and *Methanothrix*_A, both of which were methanogens and belonged to the *Euryarchaeota* phylum. The two OTUs contributed to some of the Archaea relative abundance in the deeper depths of Ace Lake (Lower 2 and 3), but not in Lower 1. Two other methanogenic archaea have been previously isolated from the Ace Lake anoxic zone, namely *M. burtonii* (Franzmann et al, 1992) and *Mtg. frigidum* (Franzmann et al, 1997), of which the latter was detected in the Ace Lake data, but its relative abundance was very low (<0.06%) in all metagenomes. The two high-quality archaea OTUs could not be assigned a species-level taxonomy based on their SSU rRNA gene comparison and ANI values to reference genomes (Appendix G).

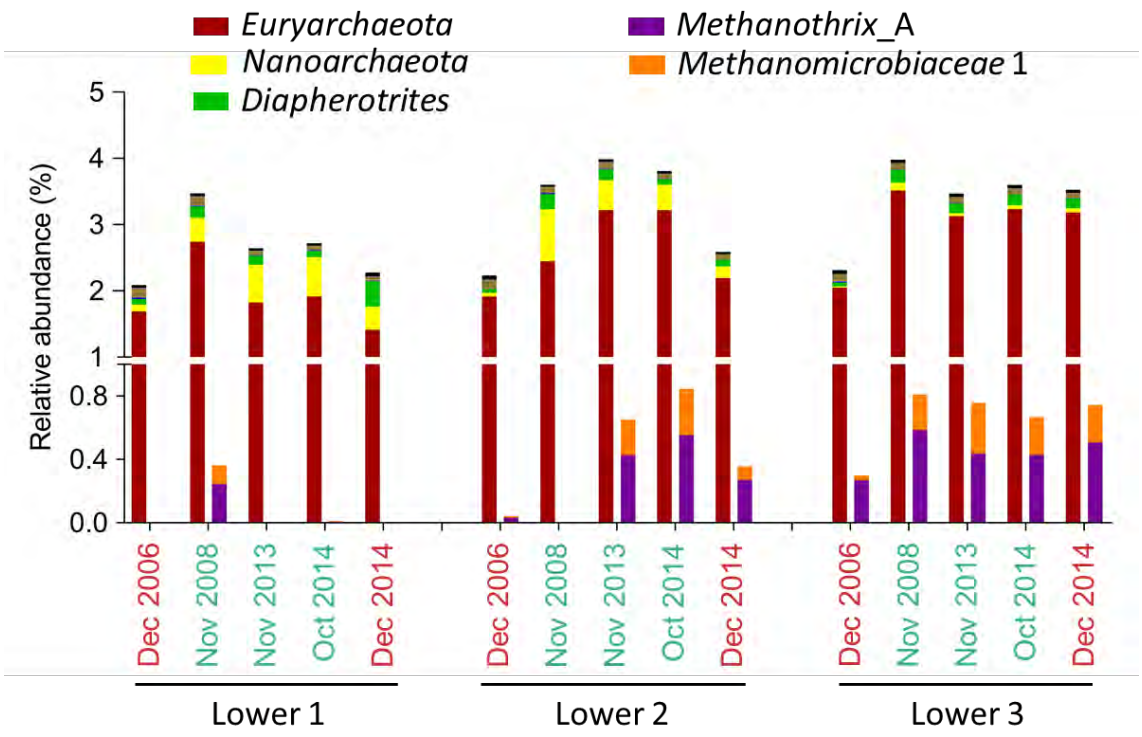


Figure 3.7 Relative abundances of archaeal OTUs in the Lower zone of Ace Lake. The stacked bar chart shows the mean of relative abundances of archaea in metagenomes from a depth and time period, i.e., the mean of relative abundances in the three filter fractions collected from each depth and time period. The means of relative abundances are shown for all archaeal OTUs belonging to *Euryarchaeota* (dark red bar), *Nanoarchaeota* (yellow bar), *Diapherotrites* (green bar), *Aigarchaeota*, *Candidatus* Korarchaeota, *Candidatus* Micrarchaeota, *Crenarchaeota*, *Thaumarchaeota* phyla as well as two abundant archaea OTUs, namely *Methanothrix*_A (purple bar) and *Methanomicrobiaceae* 1 (orange bar), identified in the Lower zone of Ace Lake. The x-axis indicates the Ace Lake depth zones (Lower 1, 2, 3) as well as the sample collection time periods, including three seasons (summer, red font; winter, blue font; spring, green font). The y-axis was split to show an expanded view of relative abundances below 1%. Winter data were not gathered from the Lower zone of Ace Lake due to logistics issues during sample collection.

3.3.4 Seasonal variations in OTU abundances

All OTU relative abundance data were normalised and used for statistical analyses, such as distLM analysis, to assess whether or not any abundance variation occurred (Figure 3.8). The relationship between variations in the OTU relative abundances and change in season was explored using environmental parameters such as air temperature, daylength, and sunlight hours that changed with season. All variations like seasonal variation, inter-annual variation, depth-based variation or biomass size-based variation

(incurred from using samples from different filter fractions) were then identified from the output of the statistical analysis. The overall Ace Lake microbial community showed prominent seasonal variation in their abundances (Figure 3.8). This was supported by the seasonal segregation of the Ace Lake metagenomes (vertical segregation along dbRDA2 in Figure 3.8) with change in season factors, especially air temperature and sunlight hours. The distLM output also showed segregation of metagenomes based on the lake depth from which they were collected (horizontal segregation along dbRDA1 in Figure 3.8). The environmental factors air temperature (P=0.001), daylength (P=0.001), and sunlight hours (P=0.002) were significant explanatory factors of change in season, whereas depth values (P=0.001) and salinity (P=0.001) significantly contributed to change in lake depth. On the other hand, inter-annual changes were not evident, with 2014 (Oct) spring samples being clustered with 2008 (Nov) and 2013 (Nov) spring samples, but separate from 2014 winter (Jul, Aug) and 2014 summer (Dec) samples. Similarly, biomass size-based variations were not observed, and all populations from the three filter fractions were highly similar and completely overlapped in the dbRDA plot (Figure 3.8).

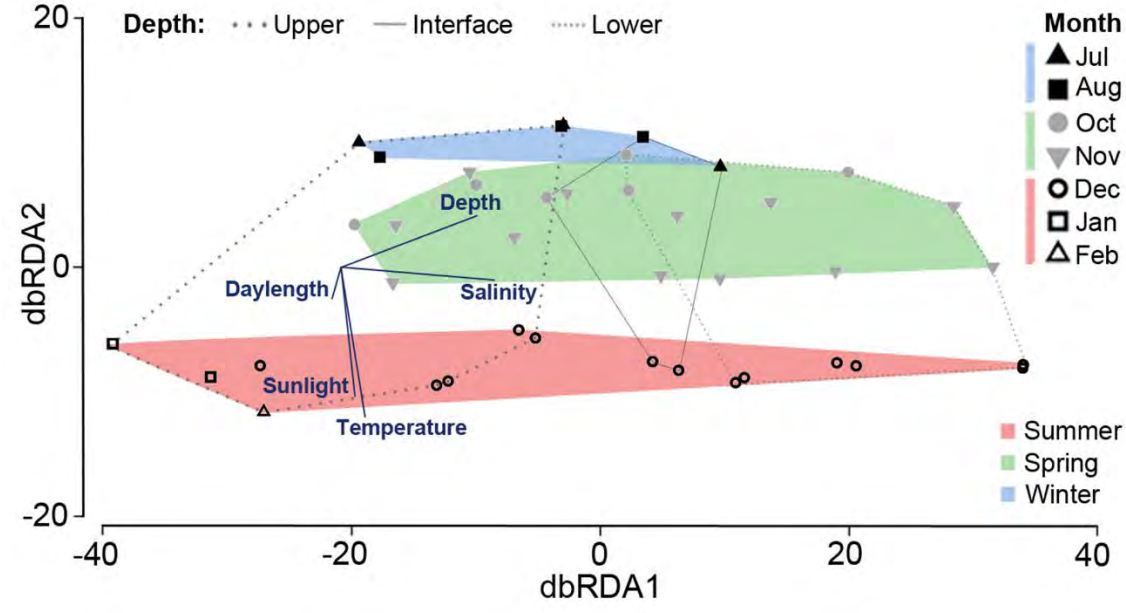


Figure 3.8 Seasonal and depth-related variations in relative abundances of OTUs identified in Ace Lake. The dbRDA plot highlights the relationship between changes in environmental factors, such as lake depth (depth), salinity (salinity), monthly average daylength (daylength), monthly average sunlight hours (sunlight), and monthly average air temperature (temperature), and variations in OTU abundances. The x-axis of the dbRDA plot explains 77% of the fitted and 21% of the total variations, whereas the y-axis of the plot explains 13% of the

fitted and 4% of the total variations. The data points representing the three filter fractions from each depth and time period overlap in the plot, reducing the 120 metagenomes to 40 data points in the plot. The figure includes vector overlays for the environmental factors. The relationship between an environmental factor and variations in OTU relative abundances is indicated by the direction and length of the factor vector, with increased vector length indicating a stronger relationship. Samples from Dec (empty circle), Jan (empty square), and Feb (empty triangle) were grouped as summer (red area cover); Jul (black triangle) and Aug (black square) were grouped as winter (blue area cover); and Oct (grey circle) and Nov (grey triangle) were grouped as spring (green area cover) to highlight seasonal variations. Samples from Upper 1, 2, and 3 were grouped as Upper (thick-dotted line); interface were grouped as Interface (solid line); Lower 1, 2, and 3 were grouped as Lower (thin-dotted line) to highlight depth-related variations.

Apart from this, the changes in the alpha diversity also showed the effects of seasonal changes on the biodiversity of Ace Lake (Figure 3.9). In the Upper 3 zone of Ace Lake, the alpha diversity was low in summer and high in winter and spring in the 0.8–3 µm-filter metagenomes, which coincided with the change in the abundance of *Synechococcus* in these metagenomes. The sudden decrease in the 0.8–3 µm-filter spring metagenome from Upper 2 was also due to the high abundance of *Synechococcus* in that metagenome. The effect of change in season was most obvious in the alpha diversity measured at the Ace Lake Interface (Figure 3.9). Here the diversity was high in winter and Oct 2014 spring when the *Chlorobium* population was very low (peak relative abundances: 6% Jul 2014; 5% Aug 2014; <1% Oct 2014), but low in summer and spring when the *Chlorobium* population dominated the Interface (peak relative abundances: 84% Dec 2006; 81% Nov 2008; 33% Nov 2013; 59% Dec 2014) (Figure 3.6). Contrarily, the Lower zone of Ace Lake seemed mostly unaffected by change in season from summer to spring, with very little variations in alpha diversity (Figure 3.9). The high contribution of *Synechococcus* and *Chlorobium* toward the similarity between metagenomes from a season (summer, winter, spring) and dissimilarity between metagenomes from different seasons (summer *vs* winter, summer *vs* spring, spring *vs* winter) was also indicated by the output of SIMPER analysis (Table 3.2).

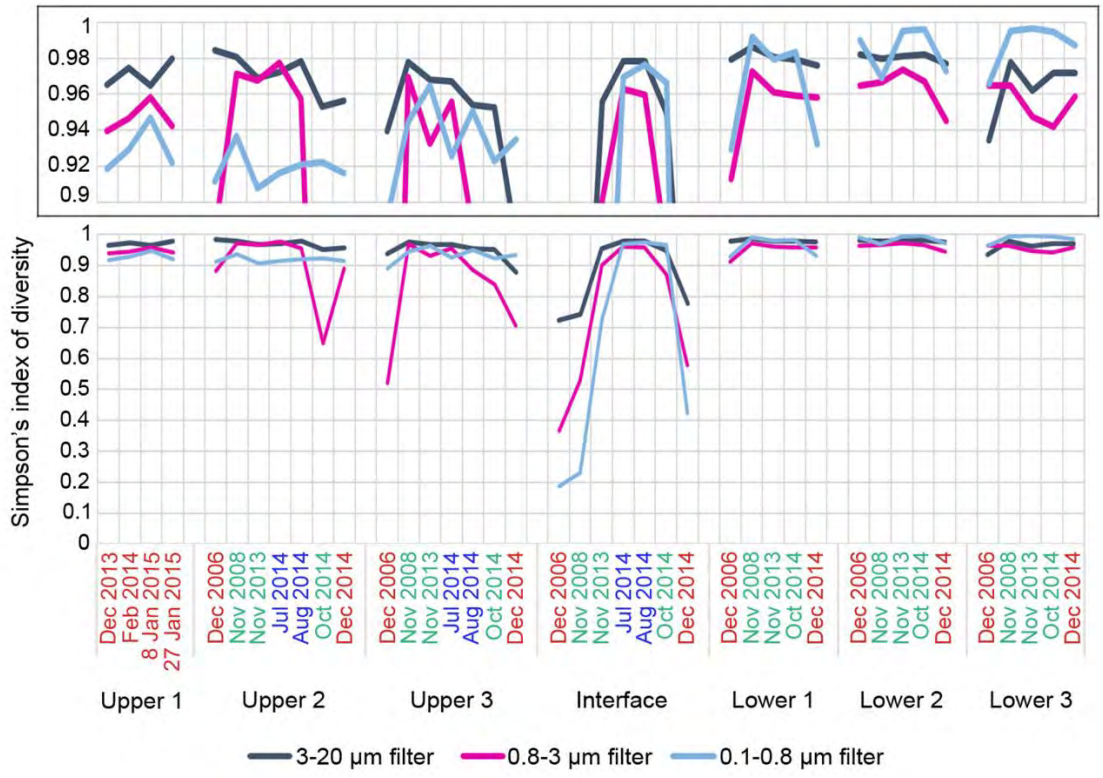


Figure 3.9 Seasonal variation in Ace Lake alpha diversity. The line graph shows the Simpson's index of diversity measures in metagenomes from three filter fractions (3–20 µm, dark blue line; 0.8–3 µm, pink line; 0.1–0.8 µm, blue line), seven Ace Lake depths (x-axis: Upper 1 to Lower 3), and three seasons (x-axis: summer, red font; winter, blue font; spring, green font). The y-axis was split to show an expanded view of diversity measures between 0.9 and 1. All Ace Lake surface (Upper 1) samples were from summer. Winter data were not gathered from the Lower zone of Ace Lake due to logistics issues during sample collection.

Table 3.2 SIMPER analysis showing similarities between samples from a season and dissimilarities between samples from different seasons as well as the top contributing OTUs. The percentages in the cells with yellow background indicate similarities between metagenomes from the same season (summer, winter, or spring). The percentages in the cells with grey background indicate dissimilarities between metagenomes from different seasons (summer *vs* winter, summer *vs* spring, winter *vs* spring). The data are shown for summer (red), winter (blue), and spring (green) samples along both x- and y-axes, and for six Ace Lake depths (Upper 2 to Lower 3) along the y-axis. The two most abundant Ace Lake bacterial taxa (*Chlorobium* and *Synechococcus*) have been highlighted in bold. No winter data are shown for Ace Lake Lower zones, as winter samples were not collected due to logistics issue during sample collection. Upper 1 data are also not shown, as all Upper 1 samples were collected in summer.

Ace Lake depth	Seasons	Summer	Winter	Spring
Upper 2	Summer	**40%** *Phycodnaviridae* 2 ***Synechococcus*** *Phycodnaviridae* 3 *Phycodnaviridae* 4 *Phycodnaviridae* 5		
	Winter	**53%** ***Synechococcus*** *Verrucomicrobia*	**62%** *Phycodnaviridae* 2	
	Spring	**52%** ***Synechococcus***	**40%** ***Synechococcus***	**59%** *Phycodnaviridae* 2
Upper 3	Summer	**42%** ***Synechococcus*** *Phycodnaviridae* 2		
	Winter	**51%** ***Synechococcus***	**61%** *Phycodnaviridae* 2 ***Synechococcus*** *Nisaea* *Microbacteriaceae* BACL25	
	Spring	**50%** ***Synechococcus***	**37%** ***Synechococcus***	**62%** *Phycodnaviridae* 2
Interface	Summer	**43%** ***Chlorobium***		
	Winter	**56%** ***Chlorobium***	**70%** ***Chlorobium***	
	Spring	**55%** ***Chlorobium***	**43%** ***Chlorobium***	**55%** ***Chlorobium***
Lower 1	Summer	**52%** ***Chlorobium***		
	Spring	**41%** ***Chlorobium*** *Desulfatiglanales* NaphS2		**70%** *Syntrophales* *Cloacimonetes* *Desulfatiglanales* NaphS2

				Bacteroidales UBA4459 *Atribacteria* 34-128
Lower 2	Summer	**51%** ***Chlorobium*** *Syntrophales* *Cloacimonetes*		
	Spring	**41%** *Syntrophales* ***Chlorobium***		**69%** *Syntrophales* *Cloacimonetes* *Desulfatiglanales* NaphS2 *Atribacteria* 34-128
Lower 3	Summer	**53%** *Atribacteria* 34-128 *Cloacimonetes* *Syntrophales*		
	Spring	**39%** *Cloacimonetes* *Atribacteria* 34-128 ***Chlorobium***		**71%** *Atribacteria* 34-128

The seasonal variation in the daily maximum incident light recorded at Ace Lake is very prominent, with light levels as high as 1,225 µE m^{-2} S^{-1} measured in summer and only 1.3 µE m^{-2} S^{-1} measured in winter (Burch, 1988). The amount of light, especially PAR, penetrating the Ace Lake depends on a number of factors like the quality of the surface ice cover (thickness and opaqueness), the amount of snow cover, and the microbial growth, all of which are affected by change in season (Hand and Burton, 1981; Burch, 1988; Burke and Burton, 1988; Rankin et al, 1999). Moreover, it has been previously noted that an ice and/or snow cover can block out large proportions of the incident light, allowing only 21% of the total incident light to pass through 1.6 m of ice in the absence of a snow cover and only 7% in the presence of an additional 30 cm snow cover (Burch, 1988). Therefore, in winter when the incident light is very low, the presence of a thick ice cover (>1m in Aug 2014; Table 3.1; Appendix I) would further limit the amount of PAR available to the phototrophs in the Ace Lake, thereby affecting their abundance. As both *Synechococcus* and *Chlorobium* were phototrophs, their low abundance in

winter was probably due insufficient amounts of PAR for energy production. However, *Synechococcus* abundance in the Upper zone recovered much faster than *Chlorobium* abundance at the Interface, in late winter and the following spring (peak relative abundances: 16% *vs* 5% in Aug 2014 and 51% *vs* <1% in Oct 2014, respectively). Also, *Synechococcus* was quite abundant at the Interface in Oct 2014 (peak relative abundance: 25%), in the near absence of *Chlorobium*, and in the Lower zone metagenomes (peak relative abundance: 8%). This could be attributed to its capacity for fermentation in the anoxic waters, allowing it to survive and grow in the dark; genes associated with fermentation were identified in the Ace Lake *Synechococcus*. This fermentative ability has also been reported in *Synechococcus* from the deep, dark waters of the Black Sea (Callieri et al, 2019).

Apart from these two photoautotrophs, some of the other abundant OTUs also displayed seasonal variations (Figure 3.10). *Algoriphagus*, *Flavobacteriaceae* MAG-120531, *Hydrogenophaga*, *Leadbetterella*, *Loktanella*, *Nonlabens*, *Polaribacter*, and *Saprospiraceae* sp. were found only in the metagenomes from the Ace Lake surface (Upper 1), especially near-shore sites, and were prevalent only in summer, as all Upper 1 samples were from summer (Figure 3.6). Other OTUs such as the eukarya *Micromonas* and the algal viruses *Phycodnaviridae* 1 and 3 also showed seasonal variation. *Micromonas* abundance was negligible in winter, which was consistent with its light-dependent survival and growth, and was quite high in spring and summer (peak relative abundances: 20% and 14%, respectively). Also, *Phycodnaviridae* 1 was more abundant in winter (peak relative abundance: 6%) in the 3–20 and 0.8–3 μm-filter metagenomes, whereas *Phycodnaviridae* 3 was more prevalent in summer and spring (peak relative abundances: 3% in both seasons) in 3–20 and 0.1–0.8 μm-filter metagenomes. Other abundant bacterial OTUs in the Ace Lake Upper zone that showed high abundance in summer and spring included *Aquiluna*, *Burkholderiaceae* MOLA814, *Burkholderiaceae* SCGC-AAA027-K21, and *Gimesia*. On the other hand, OTUs like *Balneolaceae* UBA2664, *Crocinitomix*, *Cyclobacterium*, *Halioglobus*, *Porticoccaceae* HTCC2207, *Pseudomonas*_E, and the five *Verrucomicrobia* OTUs were more abundant in winter and sometimes spring (Figure 3.10). Furthermore, *Fabibacter* and *Methylophilaceae* BACL14 were abundant only in spring samples.

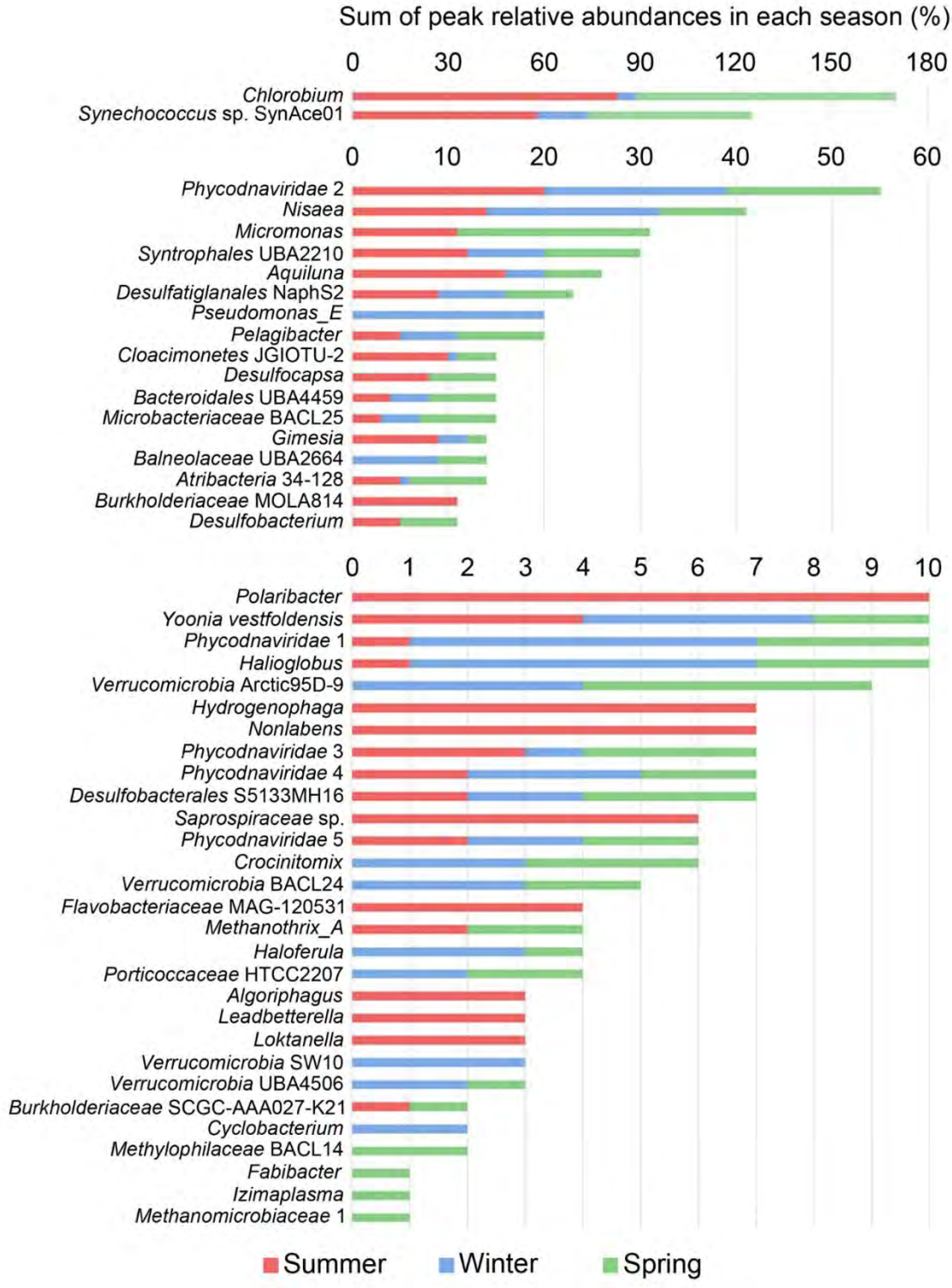


Figure 3.10 Seasonal distribution of peak relative abundances of abundant OTUs in Ace Lake. The stacked bar chart shows the peak relative abundances of the OTUs in metagenomes from summer (red bar), winter (blue bar), and spring (green bar), i.e., the highest relative abundance of an OTU among all metagenomes from summer, winter, and spring. The graph was separated into three parts and the abundance scales were redrawn to show expanded view of total peak abundances ranging from 0–10, 0–60, and 0–180.

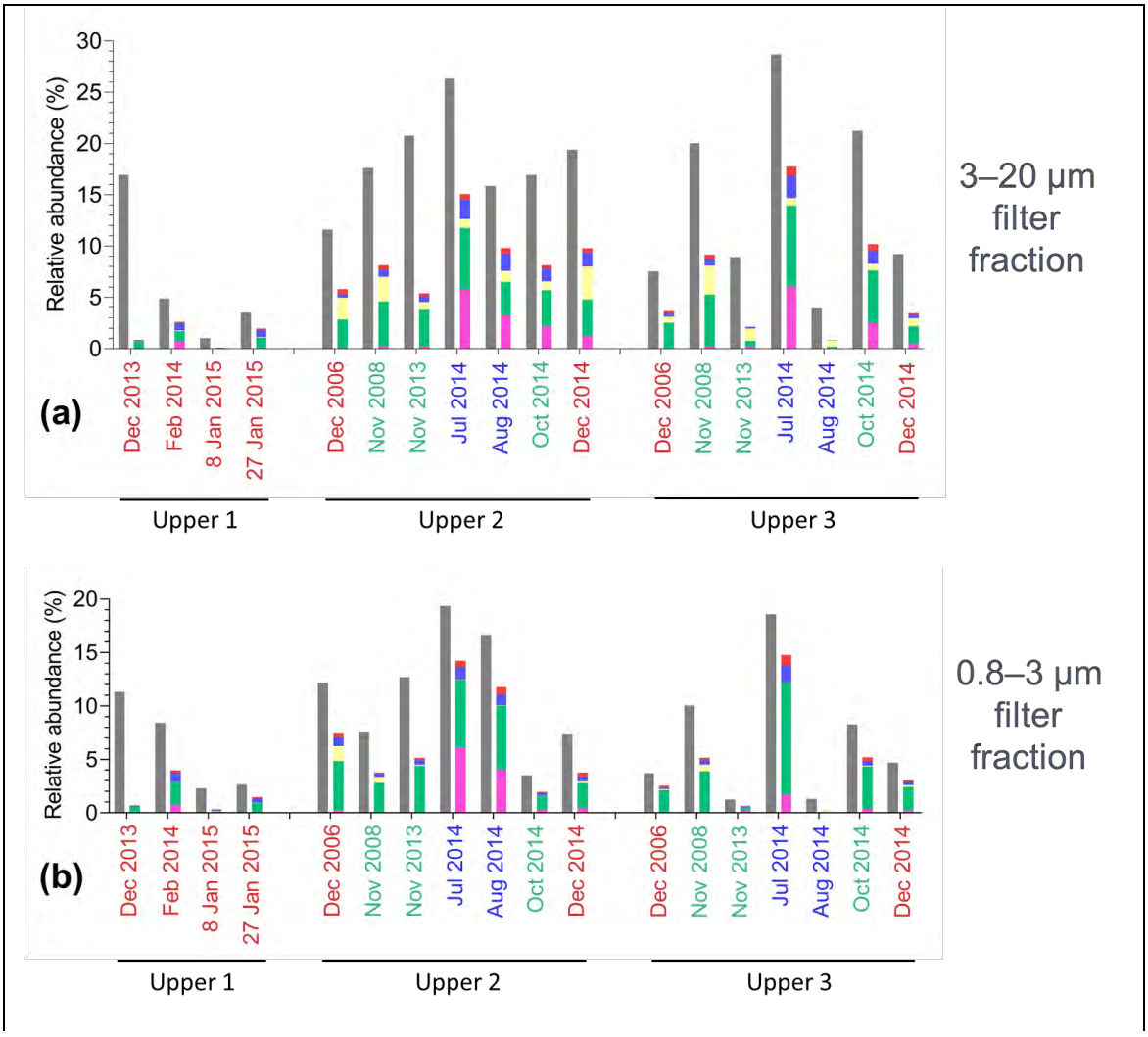
At the Ace Lake Interface, the abundances of the two *Deltaproteobacteria* OTUs *Desulfobacterium* and *Desulfocapsa* also varied with season and were found to be correlated to the *Chlorobium* abundance (Table 3.3). It has been previously shown that the Ace Lake *Chlorobium* is involved in sulfur cycling at the Ace Lake Interface, where it oxidises sulfide to sulfate, which is reduced back to sulfide by SRB (Rankin et al, 1999; Coolen et al, 2006; Ng et al, 2010; Lauro et al, 2011). *Desulfobacterium* are SRB that can reduce sulfate, and was potentially involved with *Chlorobium* in sulfur cycling. *Desulfocapsa* possessed genes for sulfur and thiosulfate disproportionation, using which it could convert sulfur as well as thiosulfate to sulfide and sulfate, also previously reported (Finster et al, 2013), that could be used by *Chlorobium* and *Desulfobacterium*, respectively. Therefore, the significant positive correlation between the abundances of *Chlorobium* and the two *Deltaproteobacteria* might suggest a possible strong co-dependence. The probable causes of the seasonal variations in the abundances of some of these OTUs were found to be related to their nutrient requirements (Panwar et al, 2020).

Table 3.3 Correlation between relative abundances of *Chlorobium* and members of *Deltaproteobacteria* at Ace Lake Interface. The correlation coefficient (*R*) and its significance (*P*-value) were calculated as described in section 3.2.4.3. The correlation between two microbes was not calculated (NC) if either of their abundances were low (<1%) in all Ace Lake Interface metagenomes from a size fraction. Correlation coefficients that were significant at 95% confidence level have been highlighted with a blue background.

Organism 1	Organism 2	3–20 µm filter		0.8–3 µm filter		0.1–0.8 µm filter	
		R	*P*-value	*R*	*P*-value	*R*	*P*-value
Chlorobium	*Desulfobacterium*	0.9	0.002	NC	NC	NC	NC
	Desulfocapsa	0.8	0.026	0.8	0.031	NC	NC
	Desulfobacterales S5133MH16	0.6	0.173	NC	NC	NC	NC
	Desulfatiglanales NaphS2	0.5	0.287	0.6	0.179	NC	NC
	Syntrophales UBA2210	0.7	0.096	0.7	0.103	NC	NC
Desulfobacterium	*Desulfocapsa*	0.9	0.015	NC	NC	NC	NC

3.3.5 Ace Lake viruses

Viral OTUs were prevalent in the Upper zone of Ace Lake and showed high abundance in metagenomes from 0.1–0.8 µm-filter, followed by 3–20 µm-filter and 0.8–3 µm-filter metagenomes. Of the 1,817 viral OTUs identified in the Ace Lake metagenomes, only 13 OTUs, including one uncharacterized viral OTU referred to as 'unclassified Virus' in the metagenome Phylodist files, had relative abundances ≥1% in at least one metagenome. Among these abundant viral OTUs, five viruses belonging to the *Phycodnaviridae* family yielded good quality bins after bin refinement with RefineM. Based on the ANI matches to their reference genomes, the five algal viruses were classified as *Phycodnaviridae* 1, 2, 3, 4, and 5 (Appendix G). The five *Phycodnaviridae* OTUs contributed to most of the viral abundance in the Ace Lake Upper zone, with *Phycodnaviridae* 2 making most of the contributions (Figure 3.11).



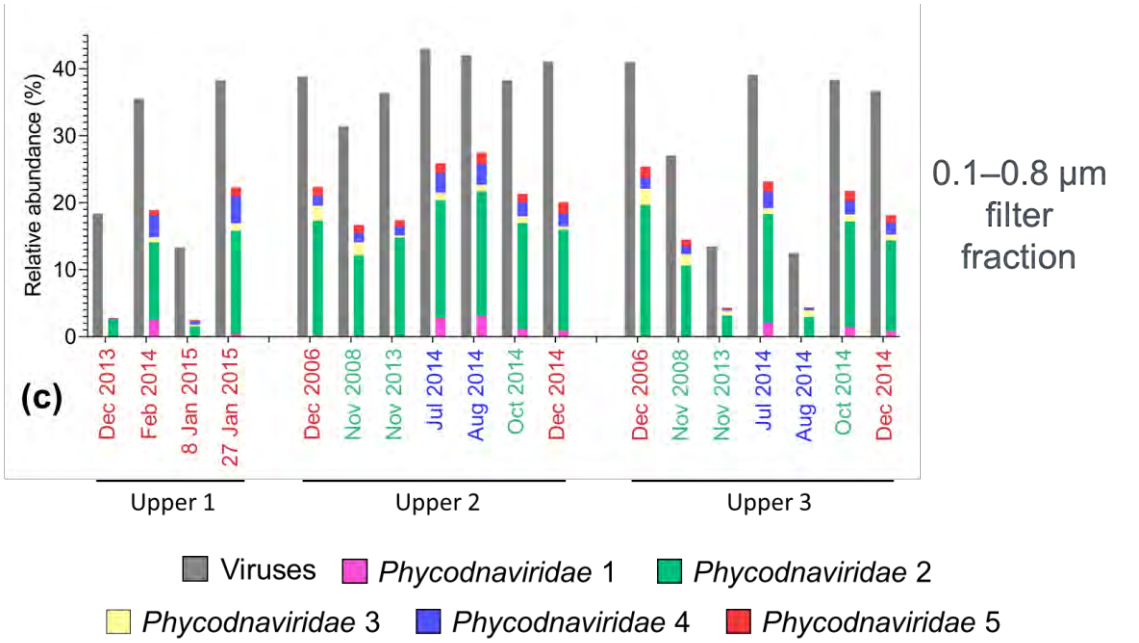


Figure 3.11 Relative abundances of viral OTUs in the Upper zone of Ace Lake. The stacked bar charts show the relative abundances of all Viruses (dark grey bar) as well as *Phycodnaviridae* 1–5 OTUs (1, pink bar; 2, green bar; 3, yellow bar; 4, blue bar; 5, red bar), the five algal viral OTUs found to be abundant in the Upper zone of Ace Lake (x-axes: Upper 1, 2, 3). The relative abundances are shown in **(a)** 3–20 µm-filter, **(b)** 0.8–3 µm-filter, and **(c)** 0.1–0.8 µm-filter metagenomes from three seasons (x-axis: summer, red font; winter, blue font; spring, green font). All Ace Lake surface (Upper 1) samples were from summer.

3.3.5.1 Viral contigs representing complete genomes

Of the 516 viral contigs in the complete phage catalogue, 337 contigs were from the Ace Lake metagenomes. These Ace Lake viral contigs were grouped as 173 distinct viral genomes based on their length, GC content, viral cluster or singleton designation, and clade assignment. The viral genomes belonged to *Caudovirales* (158), *Microviridae* (1), *Retrovirales* (1), and Cress-DNA virus/*Parvovirus* (6), but the clade of seven genomes could not be determined. Additionally, the Ace Lake zones in which these circular phage were prevalent was also determined (Appendix H: Table H1). Among the 173 distinct viruses, 87 were identified only in the Ace Lake Upper zone metagenomes and three were identified in Upper and Interface zone metagenomes. Notably, the viral genomes from *Microviridae*, *Retrovirales*, and Cress-DNA virus/*Parvovirus* were identified only in the Upper zone metagenomes from Ace Lake. These observations were consistent with the overall analysis of the Ace Lake metagenomes, which showed a higher proportion and variety of viruses in the Upper zone (Figure 3.11; also see

'other' viruses in Figure 3.13). These phage contig groups representing complete genomes of viruses need to be studied further to assess their potential hosts using the data in the spacer database (discussed in Chapter 6).

3.3.5.2 Ace Lake 'huge' phage with defence genes

Among the unassigned contigs from the Ace Lake metagenomes, 28 contigs had relative abundances ≥1% (Table 3.4). Five of these abundant unassigned contigs from the Ace Lake Lower zone were associated with cluster 24 (cl_24) in the Antarctic virus catalogue and were predicted most likely to be prophages by VirSorter (category 5). Four of these five cl_24 contigs contained *cas* genes and two of them represented complete virus genomes in the complete phage catalogue (Table 3.4). The cl_24 contained total 56 viral contigs, including the five abundant unassigned contigs. Their sequence alignment against each other using Mauve showed that nine of these contigs from Ace Lake metagenomes, plus three from an Ace Lake MetaBAT MAG, represented complete or nearly complete genome of a 'huge' phage of around 528 kb length (Appendix H: Table H1). A similar 'huge' phage was also reported previously (Al-Shayeb et al, 2020). The cl_24 contigs were aligned against all metagenome contigs to increase the representation of the viral cluster, but this yielded small contigs of length ≤1 kb that were not added to the cl_24 for further analysis.

The nine cl_24 viral contigs from Ace Lake metagenomes representing complete phage genomes were identified from the 3–20 μm-filter and 0.1–0.8 μm-filter metagenomes from the Lower zone and probably represented intracellular and virion forms of the Ace Lake 'huge' phage, respectively. The complete phage genome sequences were present in metagenomes from Nov 2008, Nov 2013, Oct 2014, and Dec 2014, with incomplete sequences also found in the Ace Lake Interface metagenomes from Jul 2014 and Aug 2014, suggesting that the phage thrives in the Lower zone. Host analysis of the 56 cl_24 contigs showed potential association with *Ammonifex degensii* (a *Firmicutes*) and *Methylomicrobium alcaliphilum* (a *Gammaproteobacteria*). The phage contained *cas* genes representing a putative type I-C CRISPR-Cas system and CRISPR arrays containing spacer sequences, but did not have any spacer acquisition genes (*cas1*, *cas2*, or *cas4*); similar to what was observed in a recently reported 'huge' phage (Al-Shayeb et al, 2020). The spacer sequences on the phage contigs (32 distinct spacers) did not match any viruses in the spacer database but were identical to the spacer sequences on their potential hosts. Additionally, the repeat sequences in the CRISPR arrays of the

phage contigs (12 distinct repeats) were identical to those from their potential hosts, suggesting that the Ace Lake 'huge' phage might be using host CRISPR-Cas machinery to acquire spacers that target the viruses infecting its host; similar to previous reports from another 'huge' phage (Al-Shayeb et al, 2020).

Apart from the Ace Lake 'huge' phage identified from the abundant unassigned contigs, four of the 28 contigs from the Ace Lake Upper zone and Interface also contained R-M system genes, of which two contigs were associated with cluster 463 (cl_463) in the Antarctic virus catalogue (Table 3.4). This viral cluster was also identified among the most abundant viral clusters in Ace Lake (section 3.3.5.3). However, host analysis of the cl_463 viral contigs did not yield any useful information; the host contigs with spacer matches to cl_463 were very small, only 186 to 514 bp long. Additionally, four of the abundant unassigned contigs contained phage genes and were potentially viral contigs. Notably, three of these contigs were the same length and their read depth was high (>4000, which was ~0.6-fold of the maximum read depth of *Chlorobium* at the Ace Lake Interface). However, these contigs showed no matches to the Antarctic virus catalogue and were not categorised as viruses or prophages by VirSorter, therefore, they were not investigated any further.

Table 3.4 Ace Lake unassigned contigs with relative abundance ≥1%. The relative abundances of the contigs were calculated by dividing the contig coverage (contig length × read depth) with the total metagenome abundance (using Formula (1) described in section 3.2.1). The yellow-highlighted contigs were from cl_24 and most of these contained *cas* genes. The green-highlighted contigs contained restriction-modification genes (R-M genes) and some of them were from cl_463. * VirSorter was used for identifying potential viral contigs. VirSorter category: 2, most likely a virus; 5, most likely a prophage. †Viral cluster or singleton designations were determined by comparing the contigs against the Antarctic virus catalogue. ‡The gene annotations on the contigs were manually parsed to assess any defence gene assignments. The protein sequences of the defence genes were aligned against the UniProtKB/Swiss-Prot database using the ExPASy BLAST online service (https://web.expasy.org/blast/), to verify the gene assignments. The contigs highlighted in red font contained phage proteins and were potentially viral contigs based on their genetic composition.

Contig ID (Metagenome)	Contig length (bp)	Read depth	Relative abundance	VirSorter category*	Cluster or singleton†	Defence genes, if any‡

Ga0222679_1000001 (Oct 2014_L1_0.1 µm)	528,258	65	2%	5	cl_24	*cas* genes
Ga0222682_1000001 (Oct 2014_L2_0.1 µm)	528,256	30	1%	5	cl_24	*cas* genes
Ga0222637_1000003 (Nov 2013_L1_0.1 µm)	323,923	98	1%	5	cl_24	No
Ga0208904_1000003 (Nov 2008_L2_0.1 µm)	447,854	173	1%	5	cl_24	*cas* genes
Ga0222640_1000001 (Nov 2013_L2_0.1 µm)	528,282	44	1%	5	cl_24	*cas* genes
Ga0222632_1000232 (Nov 2013_U2_3 µm)	36,204	4438	5%	NM	cl_463	R-M genes
Ga0222634_1000174 (Nov 2013_U2_0.1 µm)	38,474	6412	6%	NM	cl_463	R-M genes
Ga0222633_1001720 (Nov 2013_U2_0.8 µm)	10,635	5721	2%	5	NM	R-M genes
Ga0222673_1000495 (Oct 2014_I_0.1 µm)	17,665	2206	2%	NM	NM	R-M genes
Ga0222663_1001055 (Aug 2014_U3_0.8 µm)	6,783	4589	1%	NM	NM	No
Ga0222663_1001225 (Aug 2014_U3_0.8 µm)	6,047	4644	1%	NM	NM	No
Ga0222672_1001351 (Oct 2014_I_0.8 µm)	6,783	5052	1%	NM	NM	No
Ga0222672_1001480 (Oct 2014_I_0.8 µm)	6,298	5181	1%	NM	NM	No
Ga0222633_1002523 (Nov 2013_U2_0.8 µm)	7,513	5822	1%	NM	NM	No
Ga0222663_1000940 (Aug 2014_U3_0.8 µm)	7,513	4970	1%	NM	NM	No
Ga0222633_1001866 (Nov 2013_U2_0.8 µm)	9,906	5450	2%	NM	NM	No
Ga0222633_1002795 (Nov 2013_U2_0.8 µm)	6,825	5634	1%	NM	NM	No
Ga0222646_100168 (Dec 2013_U1_0.1 µm)	21,333	1988	1%	NM	NM	No

Ga0222634_1000880 (Nov 2013_U2_0.1 µm)	11,674	5947	2%	NM	NM	No
Ga0222663_1000601 (Aug 2014_U3_0.8 µm)	11,441	4188	2%	NM	NM	No
Ga0302065_10003 (Dec 2006_U2_3 µm)	23,984	26	2%	NM	NM	No
Ga0222632_1001769 (Nov 2013_U2_3 µm)	7,513	4630	1%	NM	NM	No
Ga0222663_1001043 (Aug 2014_U3_0.8 µm)	6,857	4765	1%	NM	NM	No
Ga0222632_1001005 (Nov 2013_U2_3 µm)	11,881	3963	1%	NM	NM	No
Ga0222633_1001528 (Nov 2013_U2_0.8 µm)	11,966	4740	2%	NM	NM	No
Ga0222663_1001083 (Aug 2014_U3_0.8 µm)	6,625	4570	1%	NM	NM	No
Ga0222672_1000724 (Oct 2014_I_0.8 µm)	11,543	4639	2%	NM	NM	No
Ga0222672_1001360 (Oct 2014_I_0.8 µm)	6,756	5489	1%	2	NM	No

3.3.5.3 The abundant Ace Lake viral clusters

A total of 30,897 viral contigs in the Antarctic virus catalogue were from the Ace Lake metagenomes, including 3,034 from Upper 1; 8,022 from Upper 2; 7,939 from Upper 3; 2,971 from Interface; 2,201 from Lower 1; 2,093 from Lower 2; and 4,637 from Lower 3. This suggested that viruses were prevalent throughout the Ace Lake, albeit at lower abundances in the anoxic zone (see 'other' viruses in Figure 3.13). To analyse the Ace Lake viruses, the 4,856 viral clusters and 4,142 singletons to which these Ace Lake viral contigs belonged were studied separately. A total of 17 abundant viral clusters were further analysed to determine their probable niche in Ace Lake and their potential hosts (Appendix H: Table H2). Most of these abundant viral clusters (15 out of 17) were mainly represented by contigs from the Ace Lake Upper zone metagenomes, which coincided with the observation that the Upper zone of Ace Lake harboured more variety and abundance of viruses (Figure 3.11; also see 'other' viruses in Figure 3.13). The Upper zone viral clusters cl_5, cl_11, cl_159, cl_295, and cl_463 had potential bacterial

hosts, but the clusters showed no correlation to the most abundant bacteria in the Upper zone of Ace Lake, namely *Synechococcus* (described below in section 3.3.5.5). Similarly, the Upper zone viral clusters cl_7, cl_9, cl_20, cl_32, cl_35, and cl_66 with potential eukaryal hosts showed no correlation to *Micromonas*, the most abundant eukarya in the Upper zone of Ace Lake (described in section 3.3.5.4 below). However, the two abundant clusters from the Ace Lake Lower zone, cl_248 and cl_400, were found to be associated with the *Chlorobium* in Ace Lake (described below in section 3.3.5.6).

3.3.5.4 Algal viruses

The five algal viruses, *Phycodnaviridae* 1–5, also represented 261 viral clusters and 109 singletons in the Antarctic virus catalogue, of which 107 viral clusters and 7 singletons were classified as NCLDVs in the Antarctic NCLDV catalogue. Notably, the viral clusters associated with *Phycodnaviridae* 3 completely differed from those associated with *Phycodnaviridae* 1, 2, 4, and 5, which shared most viral clusters. This grouping of the algal viral OTUs was also observed in the output of the OTU bin matches to the MetaBAT MAGs, where *Phycodnaviridae* 1, 2, 4, and 5, but not *Phycodnaviridae* 3, matched the MAG bin62, suggesting that the four algal viruses were very similar (Appendix G). This was also supported by the positive correlation between the relative abundances of *Phycodnaviridae* 1, 2, 4 and 5, whereas *Phycodnaviridae* 3 showed no correlation to the other four algal viruses (Table 3.5). Although the associations between the *Phycodnaviridae* viruses was observable, the five algal viruses showed no correlation to *Micromonas*, the most abundant green alga in the Ace Lake (Table 3.5).

The six abundant viral clusters (cl_7, cl_9, cl_20, cl_32, cl_35, and cl_66; section 3.3.5.3) from the Ace Lake Upper zone with predicted eukaryal hosts matched some of the clusters associated with the *Phycodnaviridae* OTUs. An abundance correlation between the *Micromonas* OTU and the six abundant viral clusters was performed to test potential virus-host association, however, no correlation was observed.

Table 3.5 Algal virus OTUs identified in Ace Lake — their associated viral clusters and singletons and their correlation with potential hosts. The correlation coefficient (*R*) and its significance (*P*-value) were calculated as described in section 3.2.4.3. * The correlation between *Micromonas* and *Phycodnaviridae* 1–5 was not calculated (NC) in the metagenomes from 0.1–0.8 μm-filter because *Micromonas* was not detected in this size fraction. Correlation coefficients that were significant at 99% confidence level have been highlighted with a blue background.

Comparison with the Antarctic virus catalogue and the Antarctic NCLDV catalogue				
Viral OTUs	Number of OTU contigs with matches to virus catalogue (NCLDV catalogue)	Number of viral clusters (NCLDV clusters) to which the OTU contigs belong	Number of viral singletons (NCLDV singletons) to which the OTU contigs belong	Viral clusters (NCLDV clusters) unique to an OTU
Phycodnaviridae 1	377 (231)	20 (15)	5 (1)	15 (8)
Phycodnaviridae 2	1,982 (889)	111 (64)	30 (1)	118 (54)
Phycodnaviridae 3	510 (80)	119 (25)	56 (3)	All (All)
Phycodnaviridae 4	315 (129)	32 (13)	14 (1)	26 (5)
Phycodnaviridae 5	68 (30)	12 (7)	4 (1)	5 (None)

Correlation analyses							
Organism 1	Organism 2	3–20 µm filter		0.8–3 µm filter		0.1–0.8 µm filter*	
		R	*P*-value	*R*	*P*-value	*R*	*P*-value
Micromonas	*Phycodnaviridae* 1	-0.3	0.2	-0.3	0.3	NC	NC
	Phycodnaviridae 2	-0.1	0.6	-0.4	0.1	NC	NC
	Phycodnaviridae 3	0.1	0.9	-0.3	0.3	NC	NC
	Phycodnaviridae 4	-0.2	0.4	-0.2	0.4	NC	NC
	Phycodnaviridae 5	-0.2	0.5	-0.4	0.1	NC	NC
Phycodnaviridae 1	*Phycodnaviridae* 2	0.8	0.0004	0.6	0.01	0.5	0.03
	Phycodnaviridae 3	-0.1	0.8	-0.2	0.5	-0.1	0.7
	Phycodnaviridae 4	0.9	2e-7	0.7	0.002	0.7	0.001
	Phycodnaviridae 5	0.7	0.0005	0.7	0.003	0.5	0.2
Phycodnaviridae 2	*Phycodnaviridae* 3	0.4	0.1	0.2	0.6	0.5	0.04
	Phycodnaviridae 4	0.8	4e-5	0.9	3e-7	0.7	0.001
	Phycodnaviridae 5	0.9	2e-8	0.9	5e-9	0.96	3e-10
Phycodnaviridae 3	*Phycodnaviridae* 4	0.2	0.5	0.1	0.7	0.2	0.6
	Phycodnaviridae 5	0.4	0.1	-0.01	1	0.5	0.04
Phycodnaviridae 4	*Phycodnaviridae* 5	0.8	3e-5	0.9	4e-8	0.7	0.002

3.3.5.5 Ace Lake cyanophage

The genome size of the cyanophage (549 kb) assembled from an Ace Lake 2006 metagenome was large enough for it to be considered a 'huge' phage, although no *cas* genes were identified on it. The cyanophage had good matches to 11 Ace Lake viral contigs, plus nine viral contigs from an Ace Lake MetaBAT MAG, in the Antarctic virus catalogue. The 11 Ace Lake contigs belonged to four viral clusters and 10 singletons, which were unique to Ace Lake and did not contain contigs from other Antarctic metagenomes (Appendix H: Table H1). No correlation was observed between the cyanophage and the most abundant cyanobacteria in Ace Lake, namely *Synechococcus*. However, cyanophages are known to drive the development of marine cyanobacteria (Coleman et al, 2006; Avrani et al, 2011). It is possible that the association between the Ace Lake cyanophage and *Synechococcus* depended on additional factors and was not a linear correlation (discussed in Chapter 4). Moreover, the *Synechococcus* population was mainly present in the 3–20 and 0.8–3 µm-filter metagenomes, unlike the cyanophage that was identified only in the 0.1–0.8 µm-filter metagenomes, probably existing in its virion form. The host analysis of the cyanophage using the data in the spacer database did not yield any good matches to potential host contigs. As the *Synechococcus* OTU did not contain any CRISPR-Cas system genes, consistent with previous reports on marine cyanobacteria (Cai et al, 2013), its potential viruses could not be explored using the spacer database. An abundance correlation between the *Synechococcus* OTU and the five abundant viral clusters from Ace Lake Upper zone with potential bacterial hosts (cl_5, cl_11, cl_159, cl_295, and cl_463; section 3.3.5.3) was performed to assess any virus-host relationship, however, no significant correlation was observed.

3.3.5.6 Potential *Chlorobium* viruses

The Ace Lake Interface supports a high abundance population of GSB belonging to the *Chlorobium* genus, which are known to contain genes associated with the CRISPR-Cas system and R-M enzymes for defence against viruses (Ng et al, 2010; Lauro et al, 2011; Llorens–Marès et al, 2017; Boldyreva et al, 2020). To explore the viruses probably associated with the Ace Lake *Chlorobium*, the data in the metagenome CRISPR files were used to identify 80 unique CRSIPR spacer sequences in the *Chlorobium* OTU contigs from Ace Lake. The spacer sequences matched 3,508 contigs from the 120 Ace Lake metagenomes, which were compared against the Antarctic virus catalogue and

three viral contigs were identified as potential *Chlorobium* viruses. Two of these contigs were from the viral cluster 1024 (cl_1024) and one was a singleton (sg_14554). The association of these three viral contigs with the *Chlorobium* OTU was further verified by assessing their matches to host spacers in the spacer database, which showed that their potential hosts included members of *Chlorobi* (including the Ace Lake *Chlorobium*) and *Gammaproteobacteria* and possibly members of *Deltaproteobacteria*, *Firmicutes*, *Flavobacteriia*, and *Verrucomicrobia*, suggesting that these viruses had a broad range of hosts (Table 3.6). The two abundant viral clusters from the Ace Lake Lower zone (cl_248 and cl_400; section 3.3.5.3) were analysed to identify any association with *Chlorobium*. The cl_248 contained 35 viral contigs (Appendix H: Table H1) and their host analysis showed matches to members of *Chlorobi* (including Ace Lake *Chlorobium*) and *Gammaproteobacteria*, indicating a similar host range as cl_1024 and sg_14554. On the other hand, the cl_400 contained 26 viral contigs (Appendix H: Table H1) and had matches to spacers from *Bacteroidales* UBA4459, its potential host.

The cl_1024 contained total 14 viral contigs, including the two with initial matches to the *Chlorobium* CRISPR spacers (Appendix H: Table H1), and the host analysis of these viral contigs also supported the above taxa as potential hosts of the cl_1024 viral contigs. The cl_1024 viral contigs from 2008 and 2013–2015 Ace Lake metagenomes were highly similar to each other with >98% identity across >80% alignment fraction, whereas the contigs from 2006 were also quite similar to the other contigs (>95%) but across a lower alignment fraction (>60%). This difference was probably observed due to the longer lengths of contigs from 2006 metagenomes. The cl_1024 contigs and sg_14554 were also aligned against the Ace Lake metagenome contigs to increase the representation of these potential *Chlorobium* viruses; 69 metagenome contigs had good matches to the cl_1024 (cl_1024 matches) and 29 contigs had good matches to sg_14554 (sg_14554 matches) (Appendix H: Table H1).

Abundance correlation analyses between the Ace Lake *Chlorobium* and cl_1024 group (cl_1024 + cl_1024 matches), sg_14554 group (sg_14554 + sg_14554 matches), cl_248, and cl_400 were performed to ascertain the nature of virus-host association between the bacteria and the viruses. The data showed significant positive correlation between *Chlorobium* abundance and its potential viruses from cl_1024 group (R= 0.7, P=2e-11) and sg_14554 group (R= 0.97, P=0.02) in the Ace Lake Interface and its surrounding

Upper 3 and Lower 1 zones; cl_248 also showed a positive correlation, but it was not significant (*R*=0.5, *P*=0.7) (Figure 3.12). The cl_400 showed a significant positive correlation to *Chlorobium* abundance (*R*=0.9, *P*=5e-11), although *Chlorobium* was not amongst its potential hosts determined from spacer matches. It is possible that these potential *Chlorobium* viruses (cl_1024, sg_14554, cl_248) grow cooperatively with the *Chlorobium* and were not responsible for its very low abundance in Oct 2014, considering that they were not detected in the Ace Lake metagenomes from this time period. Additionally, these potential *Chlorobium* viral clusters and singleton were identified only in the Ace Lake metagenomes in the Antarctic virus catalogue.

The *Chlorobium* spacer sequences were also thoroughly analysed to assess if there were a seasonal pattern of spacer acquisition. For this, the spacer sequences on the host contigs with matches to the two cl_1024 contigs and sg_14554 were analysed. A total of 89 unique spacers were identified on the host *Chlorobium* contigs from the 120 time-series Ace Lake metagenomes. Although no seasonal pattern was observed, many of the spacers were found on contigs from multiple time periods, highlighting that capacity of *Chlorobium* to defend against viral predation (discussed in Chapter 5).

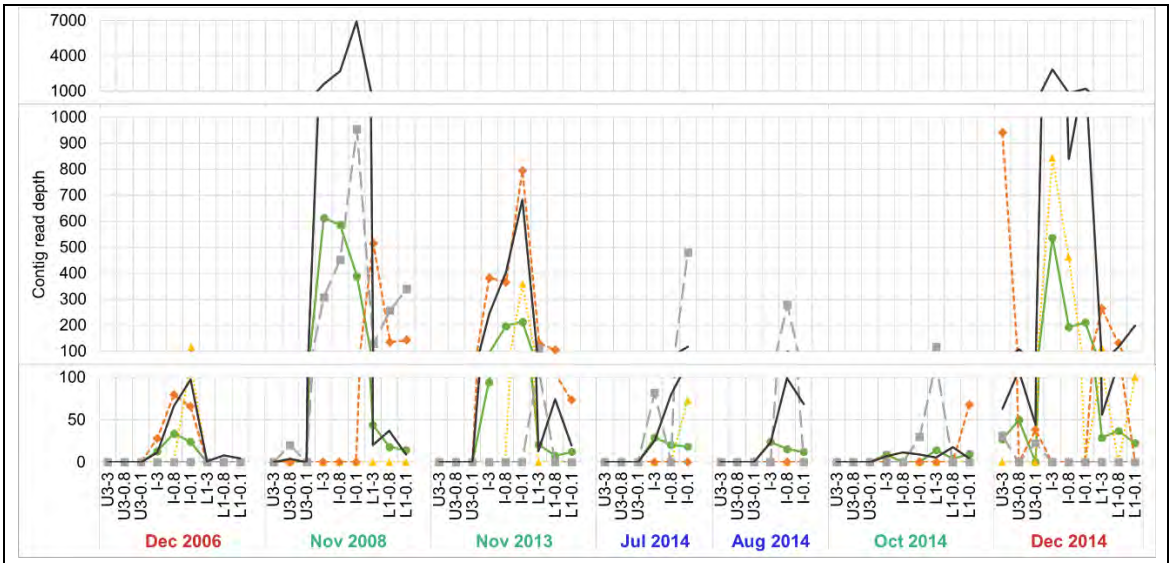


Figure 3.12 Ace Lake *Chlorobium* and its potential viruses. The line graph shows the read depth-based abundance association between *Chlorobium* (black line) and its potential viruses — cl_1024 group (green line), sg_14554 group (yellow line), and cl_248 (orange line) as well as cl_400 (grey line) in the metagenomes from three filter fractions (x-axis: 3–20 µm, 3; 0.8–3 µm, 0.8; 0.1–0.8 µm, 0.1) and three Ace Lake depths (x-axis: Upper 3, U3; Interface, I; Lower 1, L1). The metagenomes were sampled from summer (red font), winter (blue font), and spring

(green font) shown on the x-axis. Winter data were not gathered from the Lower zone of Ace Lake due to logistics issues during sample collection.

Table 3.6 Host analysis of two cluster 1024 (cl_1024) and one singleton (sg_14554) viral contigs. Total 83 spacers matched the three potential *Chlorobium* virus contigs — one singleton (sg_14554) and two cl_1024 contigs from Dec 2006 and Aug 2014 Ace Lake metagenomes (Appendix H: Table H1). Of these, 65 spacers were numbered from S1–S65, whereas the other 18 spacers were either reverse complements of S1–S65 ('RC' after the spacer number) or their sequences were same as S1–S65 but shorter by 1 nucleotide ('-1' after the spacer number; e.g., S1 was 34 nt and S1–1 was 33 nt). Spacers that were both short and reverse complements were numbered as 'RC-1' after the spacer number. The numbers in parentheses represent the number of host contigs that also had *cas* genes flanking the CRISPR spacer arrays. The spacers highlighted in red had 90-99% identity matches to the respective viral contigs, whereas all other spacer matches were 100% identical. The yellow highlighted microbe was the most abundant bacteria, *Chlorobium*, in Ace Lake. * The host taxonomy was determined from the contig taxonomies generated by the Cavlab pipeline v4 runs on the metagenomes (section 3.2.1).

Host phylum/class	Potential host taxonomy*	sg_14554 Dec 2006	cl_1024 Dec 2006	cl_1024 Aug 2014
Chlorobi	*Chlorobaculum tepidum*	S51, S52, S53, S54, S55, S56, S57, S58, S60 **(1)**		S2-2 **(1)**
	Chlorobium phaeobacteroides		S3_RC **(1)**, S4_RC **(1)**	S1 **(1)**, S3_RC **(1)**, S4_RC **(1)**
	Chlorobium phaeovibrioides	S20 **(1)**, S21 **(1)**, S22 **(1)**	S3 **(1)**, S4 **(1)**, S3_RC **(1)**, S4_RC **(1)**	S1 **(1)**, S2 **(3)**, S3 **(1)**, S4 **(1)**, S1-1 **(1)**, S1_RC **(1)**, S2_RC, S3_RC **(1)**, S4_RC **(1)**
	Prosthecochloris sp. CIB 2401	S62 **(1)**		
Gammaproteobacteria	*Acinetobacter* sp. C15	S19		

	Alcanivorax jadensis	S21 **(1)**, S49_RC, S59, S49-1	S11-1	S11-1
	Edwardsiella tarda	S19_RC		
	Halomonas subterranea	S19		
	Klebsiella pneumoniae	S23, S24, S24-1, S25, S26, S27, S28, S29, S30, S31, S32, S33, S34, S35, S35_RC-1, S36, S37, S38, S39, S40, S41, S42, S43, S44, S45	S8, S11-1, S11, S13, S14, S15	S8, S10, S11-1, S11
	Legionella massiliensis	S48 **(1)**, S48_RC **(1)**		
	Legionella pneumophila	S48-2 **(1)**		
	Marinobacter antarcticus	S49, S49_RC	S17, S17_RC	
	Marinobacter sp.	S21 **(1)**, S21_RC, S41, S42, S63, S64, S65	S7 **(1)**, S7_RC **(1)**, S18	S2 **(3)**, S2_RC, S5, S7 **(1)**, S7_RC **(1)**
	Nitrococcus mobilis		S6 **(1)**	S6 **(1)**
	Vibrio cholerae	S50		
Deltaproteobacteria	*Deltaproteobacteria*			S12
	Desulfuromonadaceae	S61_RC		
	Desulfuromonadales	S61 **(1)**		
	Desulfuromonas	S61 **(1)**		
Firmicutes	*Lactobacillus namurensis*	S46, S47	S16	
Flavobacteriia	*Runella zeae*	S31		

Verrucomicrobia	*Verrucomicrobium* sp. 3C	S9	S9

3.3.6 'Other' taxa and unassigned contigs

The low abundance (relative abundance <1%) and low-quality OTUs, together referred to as 'other' taxa — including 'other' archaea, bacteria, eukarya, and viruses, were explored further to determine their overall taxonomic composition. The 'other' bacterial OTUs contributed more toward the total bacterial abundance in the Lower zone of Ace Lake than in the Upper zone (Figure 3.13). Contrarily, 'other' viral OTUs were abundant in the Upper zone of Ace Lake alongside 'other' eukaryal OTUs, rather than in the Lower zone of Ace Lake alongside the 'other' archaeal OTUs. Among the 'other' bacterial OTUs, the overall pattern of depth distribution of the phyla and class OTUs in Ace Lake was similar to the distribution pattern of the high-quality OTUs from those taxa. The members of the phyla *Proteobacteria*, *Bacteroidetes*, *Actinobacteria*, and *Firmicutes* showed the highest combined abundances (sum of relative abundances of OTUs belonging to a phylum), with *Bacteroidetes* and *Actinobacteria* being more prevalent in the Ace Lake Upper zone (peak relative abundances: 14% and 8%, respectively) and *Firmicutes* being more abundant in the Lower zone (peak relative abundance: 6%). Of the *Proteobacteria* OTUs, most belonged to *Deltaproteobacteria* and were abundant in the Ace Lake Lower zone (peak relative abundance: 19%), followed by *Alphaproteobacteria* and *Gammaproteobacteria* that were prevalent in the Ace Lake Upper zone (peak relative abundances: 10% and 8%, respectively).

The 'other' archaeal OTUs mainly belonged to *Euryarchaeota* phylum (peak relative abundance: 3%), especially the *Methanomicrobia* class of this phylum. The 'other' eukarya group mainly comprised of members of *Chlorophyta*, *Streptophyta* as well as some uncharacterised eukaryal OTUs termed as 'unclassified Eukaryota' in the metagenome Phylodist files (peak relative abundances: 1%, 1%, and 2%, respectively). The 'other' viruses group included double-stranded DNA viruses as well as some uncharacterised viral OTUs referred to as 'unclassified Viruses' in the metagenome Phylodist files (peak relative abundances: 11%, and 12%, respectively).

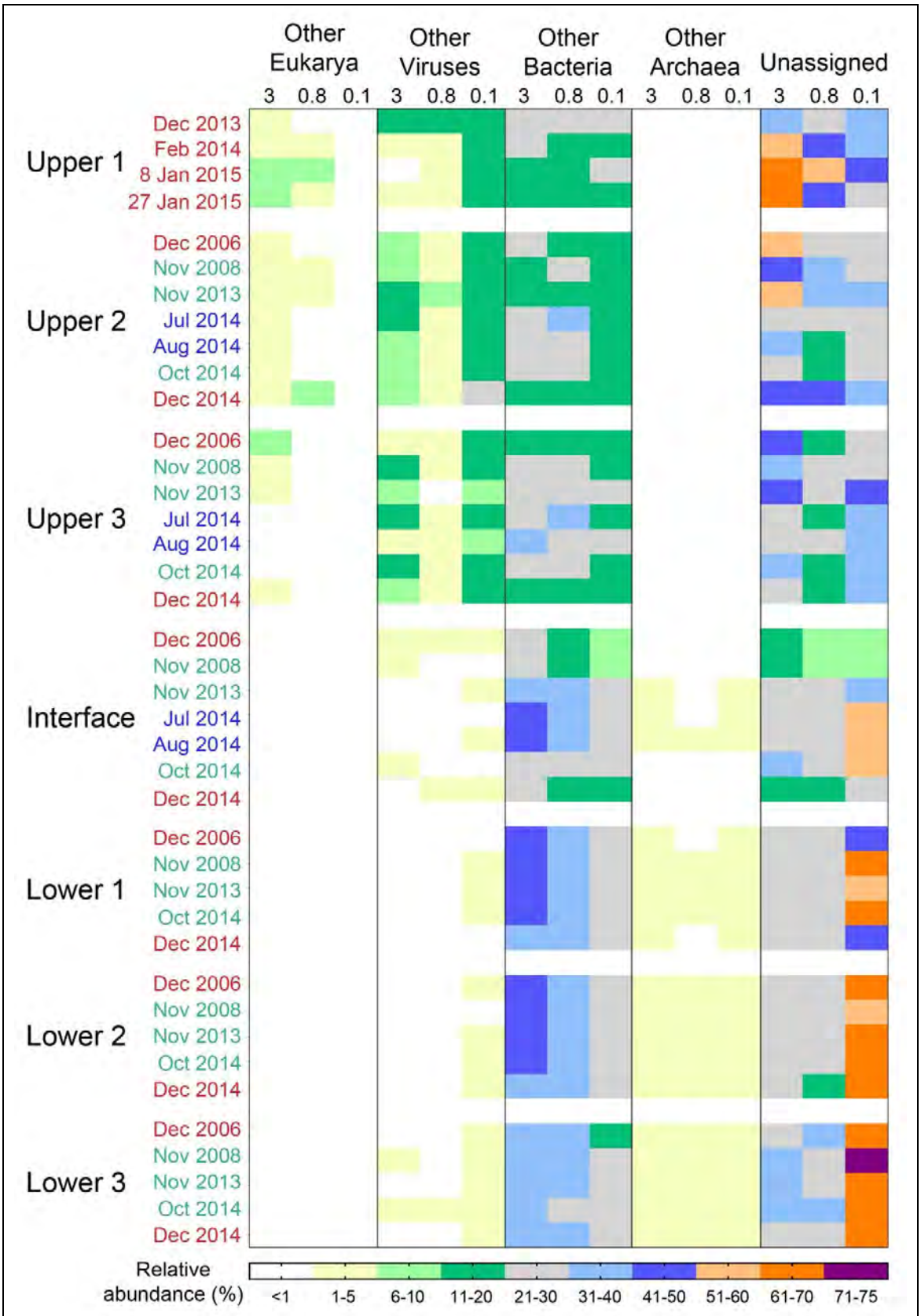


Figure 3.13 Depth and season distribution of unassigned contigs, low abundance OTUs, and OTUs with poor taxonomic assignments. The heat map depicts the relative abundances of low abundance OTUs (relative abundance <1% in all metagenomes) and OTUs with poor

taxonomic assignment grouped under 'Other' Eukarya, Viruses, Bacteria, and Archaea as well as the combined relative abundances of unassigned contigs shown as Unassigned. The relative abundances were calculated in metagenomes from three filter fractions (x-axis: 3–20 μm, 3; 0.8–3 μm, 0.8; 0.1–0.8 μm, 0.1), seven lake depths (y-axis: Upper 1 to Lower 3), and three seasons (y-axis: summer, red font; winter, blue font; spring, green font). The categorical gradient bar indicates relative abundances in percentage. All Ace Lake surface (Upper 1) samples were from summer. Winter data were not gathered from the Lower zone of Ace Lake due to logistics issues during sample collection.

The taxonomic as well as genetic composition of the unassigned contigs were determined by analysing the gene annotations on the contigs, including SSU rRNA genes. The analysis of taxonomy of the SSU rRNA genes contained in the unassigned contigs showed that most of them were associated with 'uncultured' archaea, bacteria, or eukarya, suggesting the presence of novel microbes in Ace Lake that have not yet been characterised (Table 3.7). Additionally, the analysis of the gene annotations on unassigned contigs in Ace Lake showed that they mostly comprised of 'hypothetical' genes, which might code for novel proteins. The gene annotations on unassigned contigs also included mobile elements (transposases, 1%), tRNAs (1%), and viral genes (1–2%), latter of which indicated that the unassigned contigs also included viral contigs. The presence of viruses and prophages among the unassigned data was also supported by the output of VirSorter analysis, which showed that the relative abundances of viruses were higher in the Upper zone of Ace Lake (peak relative abundance: 5%), whereas the relative abundances of prophages were higher in the depths around the Ace Lake Interface (peak relative abundances: 2%) (Table 3.7). The unassigned and 'other' OTUs data needs to be studied further, to ascertain the composition of the Ace Lake 'dark matter' (discussed in Chapter 6)

Table 3.7 Unassigned contigs in Ace Lake metagenomes — their genetic and taxonomic composition. * The percentages were calculated relative to the total gene annotations in the unassigned contigs of length ≥1 kb in a metagenome. Values from metagenomes from each depth were averaged. † The percentage values indicate the number of *16S* and *18S rRNA* genes identified on unassigned contigs that had matches to uncultured Archaea or Bacteria and uncultured Eukarya, respectively. The SSU rRNA genes included partial gene sequences, but the matches to those partial sequences were not considered. Uncultured Archaea, Bacteria, and Eukarya included *16S rRNA* gene matches to uncultured archaeon, bacterium, and eukaryote, respectively, as well as other uncultured microbes with known superphylum, phylum, clade,

and/or class taxonomy. For example, Uncultured Archaea included microbes referred to as uncultured archaeon, uncultured euryarchaeote, and uncultured DPANN archaeon in the NCBI database. Uncultured Bacteria included microbes referred to as uncultured bacterium, uncultured *Bacteroidetes*, uncultured actinobacterium, uncultured marine bacterium, uncultured *Sphingobacteria* bacterium, uncultured gamma proteobacterium, uncultured *Alphaproteobacteria* bacterium, uncultured Arctic sea ice bacterium, uncultured *Flavobacteriia* bacterium, uncultured delta proteobacterium, uncultured planctomycete, uncultured *Lentisphaerae* bacterium, uncultured *Chloroflexi* bacterium, uncultured proteobacterium, uncultured *Firmicutes* bacterium, uncultured *Spirochaetes* bacterium, uncultured Parcubacteria group bacterium, uncultured Microgenomates group, uncultured *Epsilonproteobacteria* bacterium, and uncultured *Candidatus* Atribacteria bacterium in the NCBI database. Uncultured Eukarya included microbes referred to as uncultured eukaryote, uncultured marine eukaryote, uncultured stramenopile, uncultured alveolate, uncultured fungus, uncultured heterolobosean, uncultured labyrinthulid, uncultured marine alveolate, uncultured marine picoeukaryote, and uncultured ciliate in the NCBI database. ‡ The relative abundances of Viruses and Prophages were calculated from unassigned contigs of length ≥1 kb that VirSorter confidently predicted as viruses (VirSorter categories 1 and 2) and prophages (VirSorter categories 4 and 5), respectively, using Formula (1) described in section 3.2.1. The peak relative abundances were the highest relative abundances of Viruses and Prophages in all metagenomes from each depth (Upper 1, 2, 3, Interface, Lower 1, 2, 3).

Genetic composition								
Depth		Upper 1	Upper 2	Upper 3	Interface	Lower 1	Lower 2	Lower 3
Average number of genes at a depth*	Potential viral genes	2%	1%	2%	1%	1%	1%	2%
	Hypothetical genes	71%	65%	62%	59%	57%	59%	71%
	tRNA genes	2%	1%	2%	1%	1%	1%	1%
	Transposase genes	1%	1%	1%	1%	1%	1%	1%
Taxonomic composition								
Depth		Upper 1	Upper 2	Upper 3	Interface	Lower 1	Lower 2	Lower 3
Number of SSU rRNA genes on	Uncultured Archaea	0	0	0	9%	13%	11%	9%

unassigned contigs with matches to†	Uncultured Bacteria	34%	39%	42%	52%	57%	55%	62%
	Uncultured Eukarya	33%	27%	26%	59%	0	0	0
Peak relative abundances (%) of unassigned contigs with an affiliation to‡	Viruses	5%	4%	3%	5%	3%	2%	1%
	Prophages	0.2%	0.5%	2%	1%	2%	1%	0.4%

3.3.7 Overall functional potential of Ace Lake

Nearly 40 million protein-coding genes from the Ace Lake metagenomes were parsed and analysed to understand the functional potential of the lake microbial community. The Ace Lake Upper zone supported aerobes, most of which were capable of phototrophy, whereas the Interface and Lower zones sustained obligate anaerobes, including a highly abundant photoautotroph (*Chlorobium*) at the Interface. A COG analysis of the Ace Lake metagenomes exhibiting a broad distribution of the gene functions identified in the metagenomes showed that the functional potential associated with the oxic and anoxic zones of the Ace Lake differed (Figure 3.14). Most of the genes were associated with amino acid metabolism (E), translation (J), cell membrane biogenesis (M), and replication, recombination and repair (L), and their abundance was higher in the Lower zone than in the Upper zone (Figure 3.14). This could be attributed to the presence of more biomass in the Lower zone of Ace Lake, as shown by the turbidity values measured at different lake depths in different time periods (Table 3.8). Genes associated with energy production (C), signal transduction (T), transcription (K), carbohydrate metabolism (G), lipid metabolism (I), and post-translational modification (O) were also prevalent in the Lower zone (Figure 3.14). At the Ace Lake Interface, genes associated with coenzyme metabolism (H) and inorganic ion metabolism (P) were abundant. Notably, genes associated with defence mechanisms (V) were more abundant in the anoxic zone (Interface and Lower) than in the oxic zone (Upper) of Ace Lake (Figure 3.14). This was also observed in the KEGG analysis of CRISPR-Cas spacer acquisition genes that were more abundant in metagenomes from Interface and Lower zone than from the Upper zone of Ace Lake (Figure 3.15). However, mobilome-associated genes, which could be from viruses or mobile elements, were identified from

all lake depths. The presence of viruses throughout Ace Lake was also supported by the data in the Antarctic virus catalogue that contained viral contigs from all lake depths (section 3.3.5.3). Therefore, it could be speculated that the microbes in the Ace Lake anoxic zone are well-equipped with defence genes such as CRISPR-Cas system genes (Figure 3.15) and can actively defend against viral predation. This would reduce the probability of host infection and could explain the overall low abundance of viruses in the Lower zone of Ace Lake (Figure 3.13). The reduction in or absence of viral predation could also explain the high abundance of bacteria in the anoxic waters of Ace Lake than in the Upper zone (Figure 3.13), where a large variety of viruses have been identified (Figure 3.13).

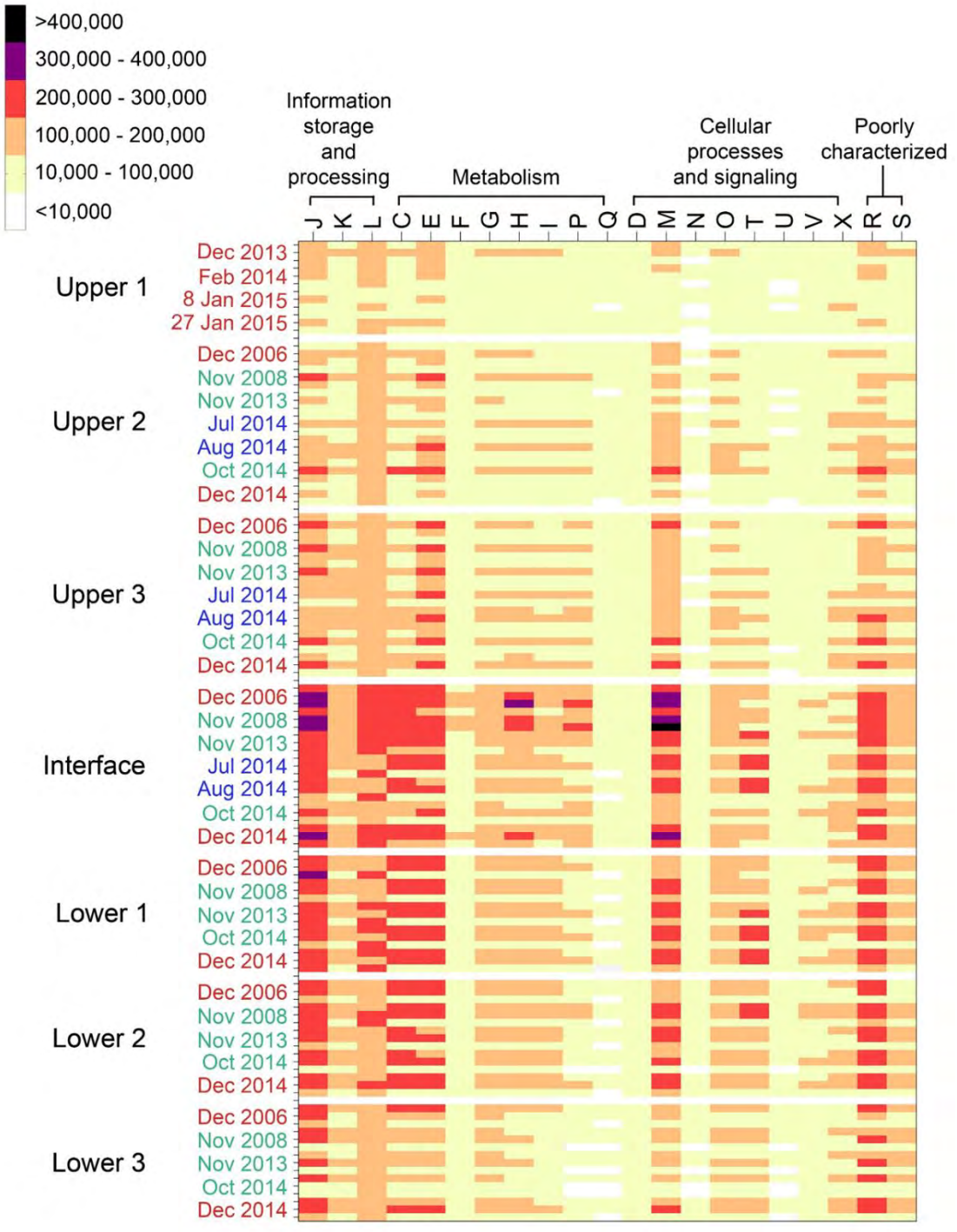


Figure 3.14 COG category classification of proteins identified in Ace Lake metagenomes. The heat map shows the normalized abundance of COG categories (x-axis) in metagenomes from seven lake depths (y-axis: Upper 1 to Lower 3) and three seasons (y-axis: summer, red font; winter, blue font; spring, green font). The COG category abundances in metagenomes from the three filter fractions from each time period and lake depth are also shown (y-axis: top, 3–20 μm; centre, 0.8–3 μm; bottom, 0.1–0.8 μm). The categorical gradient bar indicates the ranges of normalized abundances of the COG categories. COG categories A, B, W, Y, and Z are not shown because their abundance values were very low (<10,000) in all metagenomes. COG

categories: Information storage and processing — A, RNA processing and modification; B, chromatin structure and dynamics; J, translation, ribosomal structure and biogenesis; K, transcription; L, replication, recombination and repair. Metabolism — C, energy production and conversion; E, amino acid transport and metabolism; F, nucleotide transport and metabolism; G, carbohydrate transport and metabolism; H, coenzyme transport and metabolism; I, lipid transport and metabolism; P, inorganic ion transport and metabolism; Q, secondary metabolites biosynthesis, transport, and catabolism. Cellular processes and signalling — D, cell cycle control, cell division, chromosome partitioning; M, cell wall/membrane/envelope biogenesis; N, cell motility; O, post-translational modification, protein turnover, and chaperones; T, signal transduction mechanisms; U, intracellular trafficking, secretion, and vesicular transport; V, defence mechanisms; W, extracellular structures; X, mobilome: prophages, transposons; Y, nuclear structure; Z, cytoskeleton. Poorly characterized — R, general function prediction only; S, function unknown.

Table 3.8 Turbidity values measured at different depths of Ace Lake in different time periods. The turbidity measurements were taken using a YSI Sonde device in 2006 and 2008 but using a TOA WQC device in 2013 and 2014. The values were measured in Nephelometric Turbidity Units (NTU). The negative turbidity values indicate low-level turbidity. Measurements were taken in summer (red), winter (blue), and spring (green) shown on the top x-axis. The highest value in each time period is shown in red-coloured, bold numbers.

Depth	Dec 2006	Nov 2008	Nov 2013	Aug 2014	Oct 2014	Dec 2014
Upper 2	-0.1	-0.1	0.4	12	4	1
Upper 3	0.1	0.5	0	2	1	7
Interface	**139**	**87**	**44**	**26**	9	10
Lower 1	10	8	10	-	5	**53**
Lower 2	6	5	7	-	6	11
Lower 3	26	21	27	-	**24**	29

3.3.7.1 KEGG analysis

The overall functional potential of Ace Lake was analysed from the abundances of the genes associated with specific pathways and enzymes, including nutrient transporters (Figure 3.15). As KEGG analysis reflected functional potential, the abundances of the genes were directly related to the abundances of the contributing microbes. For example, sulfide oxidation was abundant at the Ace Lake Interface, except in metagenomes from winter and Oct 2014 (Figure 3.15), which coincided with the

abundance pattern of *Chlorobium* (Figure 3.6), a GSB containing genes for sulfide oxidation.

Among the high-quality OTUs analysed from Ace Lake, many contained genes for photoheterotrophy (*Aquiluna*, *Balneolaceae* UBA2664, *Burkholderiaceae* MOLA814, *Crocinitomix*, *Cyclobacterium*, *Fabibacter*, *Flavobacteriaceae* MAG-120531, *Haloferula*, *Hydrogenophaga*, *Loktanella*, *Leadbetterella*, *Methylophilaceae* BACL14, *Microbacteriaceae* BACL25, *Nisaea*, *Nonlabens*, *Pelagibacter*, *Polaribacter*, *Porticoccaceae* HTCC2207, *Pseudohongiellaceae* 2, *Saprospiraceae* sp., *Yoonia*, *Verrucomicrobia* BACL24, *Verrucomicrobia* UBA4506), with some also capable of photoautotrophy (*Chlorobium* and *Synechococcus*) and photomixotrophy (*Synechococcus*). *Chlorobium* contains the genes for anoxygenic photoautotrophy through rTCA and is known to use special light-harvesting antennae known as chlorosomes for absorbing light in low-light environments (Buchanan and Arnon, 1990; Eisen et al, 2002). KEGG analysis showed a high abundance of genes associated with rTCA cycle and a GSB type I reaction centre core complex at the Ace Lake Interface (Figure 3.15), which coincided with the high abundance of *Chlorobium* in this zone (Figure 3.6). Apart from this, the genes associated with Calvin cycle were found in all lake depths and were contributed by the most abundant cyanobacteria *Synechococcus*, which was found to be abundant throughout the Ace Lake (Figure 3.6). Genes associated with the Wood-Ljungdahl pathway were also observed in the Ace Lake anoxic zone metagenomes (Figure 3.15), suggesting that anaerobic autotrophy was prevalent in the system. These genes were contributed by some of the obligate anaerobes (*Desulfatiglanales* NaphS2, *Desulfobacterales* S5133MH16, *Desulfobacterium*, *Desulfocapsa*, *Methanomicrobiaceae* 1, *Methanothrix*_A) thriving in the Ace Lake Interface and Lower zones (Figure 3.6). Methanogenesis genes were also present in the Lower zones of Ace Lake (Figure 3.15) and were contributed by *Methanomicrobiaceae* 1 and *Methanothrix*_A, the two abundant methanogenic archaea identified in the lake (Figure 3.7). This was consistent with the previous findings suggesting that methanogenesis occurred in the anoxic zone of Ace Lake (Franzmann et al, 1991). Additionally, the abundance of genes associated with a ribose ABC transporter was high in the Upper zone (Upper 2, 3) of Ace Lake as well as at the Interface in Oct 2014 (Figure 3.15), when the abundance of *Chlorobium* was very low (Figure 3.6). This abundance was contributed by multiple OTUs (*Aquiluna*,

Burkholderiaceae MOLA814, *Gimesia*, *Loktanella*, *Microbacteriaceae* BACL25, *Nisaea*, *Pelagibacter*, *Pseudohongiellaceae* 2, *Verrucomicrobia* SW10, and *Yoonia*) that contained ribose ABC transporter genes and were abundant in the Upper zone of Ace Lake; some of them were also present at the Interface in Oct 2014 (*Aquiluna*, *Gimesia*, *Microbacteriaceae* BACL25, *Nisaea*) (Figure 3.6). Some of these OTUs (*Gimesia*, *Pelagibacter*) had the capacity to utilize ribose as a carbon source. The data showed that a variety of energy production and carbon fixation and utilization pathways (and enzymes) were employed by the diverse microbial population of Ace Lake.

Among the pathways associated with nitrogen cycling, ammonia assimilation was most prominent in the KEGG analysis (Figure 3.15), and all abundant OTUs showed capacity to utilize ammonia as a nitrogen source. Additionally, the nitrogenase gene (catalyses nitrogen to ammonia reduction) was present in the Ace Lake *Chlorobium* and *Desulfocapsa*, which explained the high abundance of this gene in the Interface and Lower zones. The Ace Lake Upper zone is known to contain very low concentration of ammonia, which increases with lake depth and is highest at the Interface (Rankin et al, 1999). Therefore, it is possible that *Chlorobium* maintains ammonia levels in the Ace Lake by fixing nitrogen when the ammonia levels drop after being assimilated by all microbes in the lake system; also suggested previously (Ng et al, 2010; Lauro et al, 2011). The genes associated with the branched-chain amino acid (BCAA) ABC transporter were abundant throughout the lake, especially in the anoxic zone of Ace Lake (Figure 3.15), indicating its importance as a carbon and nitrogen source in the lake system. BCAA ABC transporter genes were detected in *Bacteroidales* UBA4459, *Cloacimonetes* JGIOTU-2, *Desulfatiglanales* NaphS2, and *Nisaea*, of which only *Nisaea* was mainly from the Upper zone of Ace Lake.

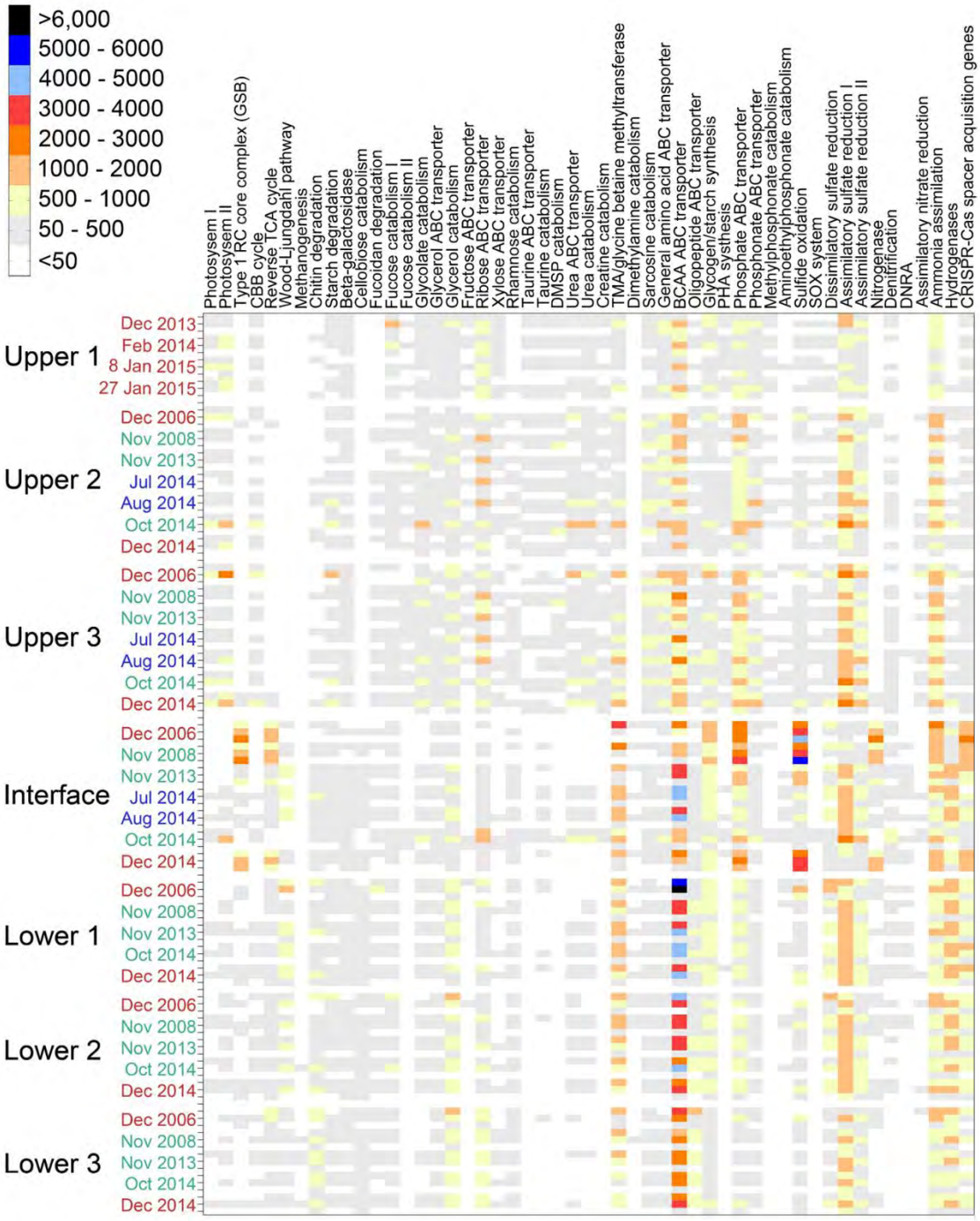


Figure 3.15 Enzymes/pathways involved in energy conservation and metabolism in Ace Lake. The heat map shows the normalized abundance of specific enzymes/pathways (x-axis) in metagenomes from seven lake depths (y-axis: Upper 1 to Lower 3) and three seasons (y-axis: summer, red font; winter, blue font; spring, green font). The enzyme/pathway abundances in metagenomes from the three filter fractions from each time period and depth are also shown (y axis: top, 3–20 µm; centre, 0.8–3 µm; bottom, 0.1–0.8 µm). The categorical gradient bar indicates the ranges of normalized abundances of the enzyme/pathway. BCAA ABC transporter, branched-chain amino acid ATP-binding cassette transporter; Cas, CRISPR-associated; CBB

cycle, Calvin–Benson–Bassham cycle; CRISPR, clustered regularly interspaced short palindromic repeats; DMSP, dimethylsulfoniopropionate; DNRA, dissimilatory nitrate reduction to ammonium; PHA, polyhydroxyalkanoate; reverse TCA cycle, reverse tricarboxylic acid cycle; SOX system, sulfur-oxidizing system; TMA, trimethylamine.

In the Ace Lake Interface, the sulfur cycling between *Chlorobium* and the SRB has been shown before, where *Chlorobium* oxidises sulfide to sulfate and the SRB reduce the sulfate back to sulfide (Rankin et al, 1999; Coolen et al, 2006; Ng et al, 2010; Lauro et al, 2011). In accordance with this, the sulfide oxidation and dissimilatory sulfate reduction pathways were found to be abundant in the Interface and Lower zones of Ace Lake (Figure 3.15), where *Chlorobium* and the SRB (*Desulfatiglanales* NaphS2, *Desulfobacterales* S5133MH16, *Desulfobacterium*) prevailed (Figure 3.6). Apart from the SRB, *Desulfocapsa*, also a member of *Deltaproteobacteria*, was abundant in the anoxic zone of Ace Lake (Figure 3.6). This microbe has the capacity for sulfur and thiosulfate disproportionation (Finster et al, 2013); thereby, producing sulfide that could be used by *Chlorobium* and sulfate that could be used by the SRB (Figure 3.16). Additionally, genes associated with sulfate assimilation (assimilatory sulfate reduction) were abundant throughout the lake, suggesting that many OTUs had the capacity to utilise sulfate as a sulfur source.

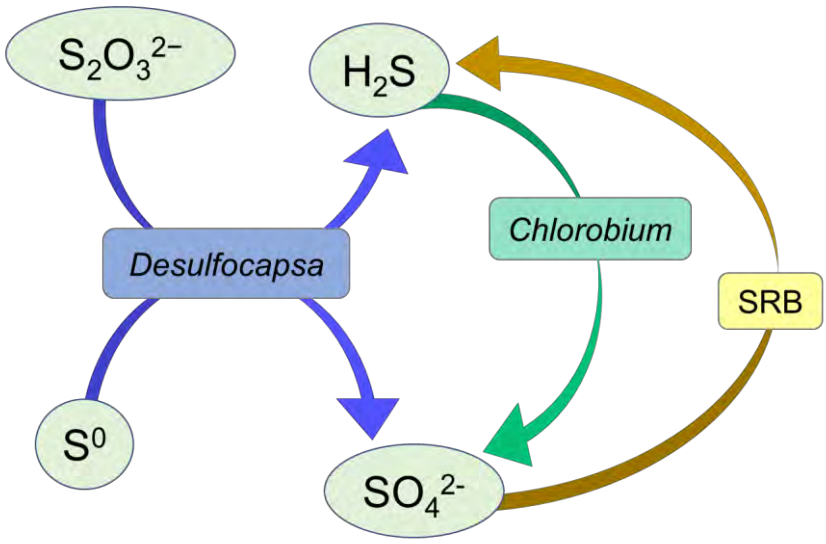


Figure 3.16 Sulfur cycling between *Chlorobium* and some *Deltaproteobacteria* at the Ace Lake oxycline. The figure shows the interaction between the sulfide oxidising GSB *Chlorobium* and the members of *Deltaproteobacteria*, namely *Desulfocapsa* and the three SRB — *Desulfatiglanales* NaphS2, *Desulfobacterales* S5133MH16, *Desulfobacterium*. Of these, *Desulfocapsa* is involved in sulfur (S^0) and thiosulfate ($S_2O_3^{2-}$) disproportionation into sulfate

(SO_4^{2-}) that can be used by the SRB and sulfide (H_2S) that can be used by *Chlorobium*, at the Ace Lake Interface.

A number of hydrogenases were also identified in the abundant OTUs, especially in the Lower zone, which was supported by the high abundance of hydrogenase genes in the anoxic zone of Ace Lake (Figure 3.15). The importance of hydrogen cycling in the Ace Lake Lower zone has been reported in a publication suggesting that the anoxic zone microbes have the capacity to utilise hydrogen as a source of energy and probably exude hydrogen during anaerobic respiration or fermentation. (Panwar et al, 2020).

3.4 Conclusion

The analysis of the Ace Lake data showed that its microbial community was very diverse, with the Upper zone mainly harbouring phototrophs including a picoeukaryote (*Micromonas*) and a highly abundant cyanobacteria (*Synechococcus*) (section 3.3.3). A high abundance of an anoxygenic, photoautotrophic GSB (*Chlorobium*) also existed at the oxycline (Interface) of Ace Lake, which was consistent with previous findings (Rankin et al, 1999; Ng et al, 2010; Lauro et al, 2011). On the other hand, the Lower zone mainly supported obligate anaerobes including many members of *Deltaproteobacteria* (*Desulfatiglanales* NaphS2, *Desulfobacterales* S5133MH16, *Desulfobacterium*, *Desulfocapsa*, *Syntrophales* UBA2210), some bacterial candidate phyla (*Atribacteria* 34-128, *Cloacimonetes* JGIOTU-2), and methanogenic archaea (*Methanomicrobiaceae* 1, *Methanothrix*_A) (section 3.3.3). This niche segregation of the microbes in Ace Lake probably allows them to coexist within the lake environment; and has been reported before (Rankin et al, 1999; Lauro et al, 2011). The analysis of the 120 time-series metagenomes from Ace Lake showed that the changes in season could severely impact the abundances of these microbes, especially the phototrophs in the Upper and Interface zones that rely on light for primary production (section 3.3.4). Contrarily, the abundances of most Lower zone microbes did not vary drastically with seasonal changes, as they relied on chemolithoautotrophy for energy production.

A variety of viruses were also detected throughout the Ace Lake, including the complete genome of an abundant 'huge' phage (~528 kb; cl_24) containing some *cas* genes and CRISPR spacers, which it might use to target other viruses infecting its potential host (section 3.3.5.2). Five algal viruses (*Phycodnaviridae* 1-5), a cyanophage, and some

potential GSB viruses (cl_1024, cl_248, sg_14554 as well as cl_400) were also identified from the Ace Lake metagenomes. Of these, only the GSB viruses showed any association with their potential hosts (*Chlorobium*). These GSB viruses might be prophages considering that they were mainly detected in the 3–20 and 0.8–3 μm-filter metagenomes and their abundances positively correlated with their host abundance (section 3.3.5.6). The algal viruses showed no correlation to the abundant alga *Micromonas* in the Upper zone of Ace Lake, suggesting that either there was no virus-host relationship or that the relationship was more than a simple, linear correlation (section 3.3.5.4). Similarly, the cyanophage did not correlate with the most abundant marine cyanobacteria *Synechococcus* in Ace Lake (section 3.3.5.5), but their interaction might be more complex, especially considering that cyanophages have the capacity to direct the evolutionary growth of marine cyanobacteria (Coleman et al, 2006; Avrani et al, 2011). Overall, the variations in the relative abundances of these abundant microbes (*Chlorobium*, *Synechococcus*, *Micromonas*) probably did not result from viral predation and lysis and was rather due to seasonal changes and low light availability in winter.

Ace Lake biodiversity has been thoroughly investigated by many research groups (Hand and Burton, 1981; Burch, 1988; Burke and Burton, 1988; Franzmann et al, 1992; Gibson and Burton, 1996; Franzmann et al, 1997; Rankin et al, 1997; Rankin, 1998; Bell and Laybourn-Parry, 1999; Rankin et al, 1999; Laybourn-Parry et al, 2005; Madan et al, 2005; Powell et al, 2005; Ng et al, 2010; Lauro et al, 2011; Laybourn-Parry and Bell, 2014; Panwar et al, 2020), yet there are some aspects of the lake system that need to be studied further. Among the ~25 million contigs assembled from the 120 Ace Lake metagenomes, many of the contigs (~11 million) could not be assigned a taxonomy, most of these being <1 kb in length (~10 million), and were studied separately as 'unassigned contigs' (section 3.3.6). The analysis showed that the unassigned contigs of length ≥1 kb comprised of uncultured microbes that could potentially be novel organisms. Most of the genes annotated on these contigs coded for 'hypothetical' proteins and could potentially be novel genes (Table 3.7). Therefore, further studies are required to analyse this 'dark matter'. A variety of viruses were detected throughout the Ace Lake, with some potentially associated with the most abundant taxa in the lake (section 3.3.5; Appendix H: Tables H1 and H2). Many viral contigs representing complete genomes were also identified (section 3.3.5.1; Appendix H: Table H1), which need to be analysed to assess their potential hosts. High-quality bins of the green alga

Micromonas and the *Phycodnaviridae* 1-5 algal viruses are also required to study their functional potential and probable interactions.

The two most abundant microbes in Ace Lake, *Synechococcus* and *Chlorobium*, have been analysed and discussed in Chapters 4 and 5, respectively. Additionally, the *Chlorobium* OTUs detected in two other meromictic systems in the Vestfold Hills, namely Ellis Fjord and Taynaya Bay, have been compared with the Ace Lake *Chlorobium*, to assess potential endemism, and are discussed in Chapter 5.

4. Ace Lake *Synechococcus* — genomic variation and potential for defence against viruses

4.1 Introduction

Synechococcus are picocyanobacteria belonging to the *Synechococcaceae* family of *Synechococcales* order. *Synechococcus* was among the most abundant microorganisms in Ace Lake (Chapter 3 section 3.3.3). *Synechococcus* has been previously reported to be abundant in the oxic zone (Upper zone) of Ace Lake, showing highest abundance in the depths just above the oxycline of the lake (Rankin et al, 1997; Rankin, 1998; Rankin et al, 1999; Powell et al, 2005; Lauro et al, 2011). Two *Synechococcus* genomes have been assembled from a *Synechococcus* species isolated from Ace Lake by Powell et al (2005) — a complete genome of *Synechococcus* sp. SynAce01 (hereafter referred to as SynAce01) sequenced and assembled by Tang et al (2019) and a draft genome of *Synechococcus* sp. Ace-Pa sequenced and assembled by JGI as part of a project led by Cavicchioli R. The *16S rRNA* gene-based phylogeny and functional potential, especially metabolic functions, of this *Synechococcus* have been previously analysed (Rankin, 1998; Powell et al, 2005; Lauro et al, 2011; Tang et al, 2019).

More than 20 *Synechococcus* clades, each representing an ecotype, have been identified in various marine habitats (Ahlgren and Rocap, 2006; Ahlgren and Rocap, 2012; Sohm et al, 2016). In the Black Sea, four *Synechococcus* phylotypes representing different strains of *Synechococcus*, two from epipelagic zone and two from mesopelagic zone, were reported (Cesare et al, 2020). In the Sargasso Sea, seven *Synechococcus* phylotypes were identified, of which two were *Synechococcus* ecotypes with varying capacity for light and nitrogen utilization (Ahlgren and Rocap, 2006). The marine *Synechococcus* ecotypes are shaped by temperature and availability of macronutrients and iron, based on the data gathered from the surface waters of the Atlantic and Pacific Oceans (Sohm et al, 2016). *Synechococcus* phylotypes from seven *Synechococcus* clades were also identified in two cyclonic eddies in South China Sea, and their abundances and distribution in the eddies were defined by temperature, salinity, chlorophyll a concentration, and availability of macronutrients and light (Jing and Liu, 2012).

Synechococcus was observed in all depths of Ace Lake, although it was more abundant in the Upper oxic zone than at the Interface or Lower anoxic zone (Figure 3.6). Its abundance varied with season, being high in summer and low in early winter but recovering by late winter (Figure 3.6). Its ability to survive in the anoxic depths of Ace Lake and grow in the dark winter could be attributed to its fermentative capacity (Chapter 3 section 3.3.4). As the Ace Lake environment in summer/winter and oxic/anoxic zones is different (Chapter 3 section 3.3.2, Figure 3.4), it is likely that the *Synechococcus* identified in metagenomes from the Upper oxic *vs* Lower anoxic zone and summer *vs* winter represent different phylotypes or ecotypes. The term phylotype is rank-neutral and often used in place of OTUs in microbiology (Moreira and López-García, 2011). Similar to phylotypes, ecotypes have no taxonomic rank, and refer to organisms that belong to the same species but have different genetic composition, which allows them to adapt to specific environments. These genetic differences, however, are not sufficient to categorise ecotypes as sub-species, since the differences in their genetic makeup is due to the specific environment they are found in (Mayr, 1999). In this chapter, genomic variations in Ace Lake *Synechococcus* that might represent distinct phylotypes or ecotypes of this cyanobacterium were investigated. Here, *Synechococcus* phylotypes refer to Ace Lake *Synechococcus* that showed subtle differences in their genetic composition but the genetic differences did not appear to affect their metabolism. *Synechococcus* ecotypes refer to Ace Lake *Synechococcus* that showed genetic differences related to their metabolic capacity, and might indicate niche adaptation. *Synechococcus* phylotypes and ecotypes have also been referred to as *Synechococcus* subpopulations in this chapter.

The metagenome-based study of Ace Lake viruses revealed a cyanophage potentially associated with *Synechococcus* (Chapter 3 section 3.3.5). Since *Synechococcus* abundance is high in Ace Lake for most of the year (Figure 3.6), it is likely that it has some defence mechanisms that help in evading viruses or disrupting viral attacks. In general, the prokaryotic defence systems have been broadly classified as: (i) host resistance-based, (ii) host immunity-based and (iii) host cell dormancy and apoptosis-based (Koonin et al, 2017). In host resistance-based defence against viruses, the host cells generate variant cell surface receptors, which can affect virus attachment leading to viral evasion; previously reported in marine cyanobacteria and Antarctic haloarchaea (Avrani et al, 2011; Tschitschko et al, 2015; Tschitschko et al, 2018). The host

immunity-based defence against viruses involves defence systems such as CRISPR-Cas system, restriction-modification (R-M) system, bacteriophage exclusion (BREX) system and a defence island system associated with restriction–modification (DISARM) that identify invading viruses and neutralize them (Barrangou et al, 2007; Goldfarb et al, 2015; Koonin et al, 2017; Ofir et al, 2018). The host cell dormancy and apoptosis-based defence against viruses involves toxin-antitoxin (T-A) system, particularly the T-A systems involved in abortive infection (ABI) mechanism (Gerdes et al, 2005; Koonin et al, 2017). In hosts with T-A systems, a stable toxin component (protein) is produced along with an unstable antitoxin component (RNA or protein), which can either inactivate the toxin or downregulate its expression (Koonin et al, 2017). In the absence of the antitoxin component, which can happen during viral infection, the toxin can cause cell dormancy or cell death, which prevents the virus from spreading to the rest of the host cell colony (Koonin et al, 2017). Of these prokaryotic defence systems, the CRISPR-Cas defence system has not been identified in marine cyanobacteria such as *Synechococcus* and *Prochlorococcus* (Cai et al, 2013). This was also true for Ace Lake *Synechococcus* that did not appear to have any CRISPR-Cas genes (Chapter 3 section 3.3.5.5). Therefore, the genomic composition of Ace Lake *Synechococcus* was further investigated to identify other bacterial defence systems (described below in section 4.2.4).

4.1.1 Aims

The overall aim was to investigate any genomic variation in the *Synechococcus* present in metagenomes from different seasons (summer *vs* winter *vs* spring) and lake depths (upper oxic *vs* interface *vs* lower anoxic), to identify its potential phylotypes or ecotypes in Ace Lake. For this purpose, the *Synechococcus* MAGs were compared with each other and SynAce01 genome in a preliminary analysis. This was followed by FR (fragment recruitment) of metagenomic reads from different time periods and Ace Lake depths to the SynAce01 genome to further verify potential *Synechococcus* phylotypes and ecotypes. As *Synechococcus* associated viruses have been identified in the Ace Lake metagenomes (Chapter 3 section 3.3.5), a specific aim was to examine the defence genes annotated in the MAGs to assess the possibility of defence against viruses.

4.2 Methods

4.2.1 Preliminary analysis of genomic variation within Ace Lake *Synechococcus* population using MAGs

Ace Lake metagenomes were sequenced, assembled, and annotated as described in Chapter 2 section 2.1.1. From each Spades-assembled metagenome, high- and medium-quality MAGs were generated by JGI's IMG system, using MetaBAT and CheckM. The *Synechococcus* MAGs represented the draft genomes of the Ace Lake *Synechococcus* from different time periods and lake depths. According to the IMG taxonomic classification of the *Synechococcus* MAGs, the closest related species to these MAGs was *Synechococcus* sp. SynAce01 indicating that the *Synechococcus* MAGs and SynAce01 are probably the same *Synechococcus* species. The MAG data included nucleotide sequences of the contigs as well as protein and nucleotide sequences of the open reading frames annotated on those contigs. The *Synechococcus* MAGs (Appendix A: Table A2) were downloaded from the Ace Lake time-series metagenomes (Appendix A: Table A1) available on JGI's IMG/M website (https://img.jgi.doe.gov/cgi-bin/m/main.cgi).

Synechococcus MAGs with ≥99% genome completeness, i.e., MAGs that contained most of the lineage marker genes associated with the species to which the MAGs were assigned (Parks et al, 2015), were used for the preliminary analysis of *Synechococcus* genomic variation. A more detailed analysis of *Synechococcus* genomic variation was performed using FR (described below in section 4.2.2). The *Synechococcus* MAG contigs were aligned to the contigs of the *Synechococcus* MAG generated from Dec 2014_Upper 3_0.8 µm-filter metagenome. This MAG was selected because it had the highest total base pair count among the Ace Lake *Synechococcus* MAGs and ≥99% genome completeness. The *Synechococcus* MAG contigs were also aligned to SynAce01 genome, but not to Ace-Pa. Both SynAce01 and Ace-Pa were sequenced from the same *Synechococcus* isolate, but SynAce01 represented the complete genome of the species, whereas Ace-Pa represented a draft genome. For these alignments, the blastn module of BLAST+ v2.9.0 was used. The BAM and BAI alignment and index files were generated from the contig alignments using Samtools v1.10. The alignments were analysed using IGV to assess the types of variation, indels or single nucleotide polymorphisms (SNPs), in the MAG sequences (method described in Chapter 2 section 2.2.5.2). The genes annotated in the variable sequence regions of the MAGs were analysed. The ANIs of the MAGs against SynAce01 genome and against each other

were calculated using pyani (method described in Chapter 2 section 2.2.4.2). The average amino acid identity (AAI) of the MAGs was calculated using the AAI-profiler online service, which compared the input protein sequences with the proteins of species in the UniProt database (http://ekhidna2.biocenter.helsinki.fi/AAI/; Medlar et al, 2018).

4.2.2 FR analysis of genomic variation within Ace Lake *Synechococcus* population

The FR analysis of Ace Lake metagenomic reads to SynAce01 genome was used for a more in-depth study of the genomic variation in Ace Lake *Synechococcus* populations from different seasons and lake depths. This involved aligning metagenome filtered reads from different seasons and time periods to Ace Lake *Synechococcus* genome SynAce01. The number of read bases from a metagenome that mapped to each base of SynAce01 were referred to as SynAce01 base coverages in a metagenome. The average of all SynAce01 base coverages from a metagenome was referred to as SynAce01 mean read depth in the metagenome. The base coverages of SynAce01 in each metagenome were used to identify genomic regions with coverages lower (low coverage regions, LCRs) or higher than SynAce01 mean read depth in the metagenome. LCRs indicated genomic regions present only in a fraction of the *Synechococcus* population from a season or lake depth, thereby suggesting presence of different phylotypes. Depending on the genes present in the LCRs, the *Synechococcus* phylotype containing the LCR might have a unique metabolic capacity that helps in niche adaptation and might represent an ecotype. High coverage regions would indicate overrepresented genomic regions, usually observed in regions containing mobile elements or multicopy genes in certain configurations (described below in section 4.3.5.1). Similar approaches have been previously used to identify Antarctic haloarchaea phylotypes and ecotypes (DeMaere et al, 2013; Tschitschko et al, 2015; Tschitschko et al, 2016; Tschitschko et al, 2018).

The Ace Lake metagenomes were selected from different depths and time periods based on the overall abundance of *Synechococcus* OTU in them (Table 4.1). The reads from selected 3–20 and 0.8–3 µm-filter metagenomes from a time period and depth were combined to form merged metagenomes (Table 4.1). Ace Lake metagenomes from Dec 2006 were not used for this comparative analysis due to differences in sequencing methods and types of reads. The Dec 2006 metagenomes contained unpaired reads sequenced using Sanger and 454 sequencing methods, which generated ≤0.8 million reads (containing ≤500 million bases) per metagenome. On the other hand, all other

metagenomes contained paired-end reads sequenced using Illumina technology, which produced ≥13 million reads (containing ≥3 billion bases) per metagenome (Chapter 2 section 2.1.1; Appendix A: Table A1). This difference in total read and base counts could lead to read depth bias, making it difficult to assess SynAce01 low coverage regions in Dec 2006 metagenomes during FR analysis, where low read depth could indicate non-alignment of reads (suggesting low coverage genomic regions) or unavailability of reads that could align to the region (due to the use of low coverage sequencing data).

The reads in the merged metagenomes were aligned to the SynAce01 genome using BBMap v38.51 (https://sourceforge.net/projects/bbmap/) with 95% minimum alignment identity (minid=0.95), to generate SAM alignment files and base coverage files. The BAM and BAI alignment and index files were created from SAM files using Samtools v1.10. The total number of reads from a merged metagenome that aligned to SynAce01 genome were calculated from the alignments in the BAM files using the 'flagstat' function of Samtools v1.10. The BAM and BAI files were analysed using IGV and only the SNPs with variant frequency ≥0.9 (i.e., at least 90% of the aligned reads contained the mutation) were considered as fixed mutations during the analysis. The base coverage files were utilised to assess the read depth distribution of SynAce01 in Ace Lake merged metagenomes, using Python v3.6 scripts and plots to identify variable coverage regions. The data in the base coverage files were also used to generate circos plots in R v4.0.2 to highlight the variable coverage regions and to show the abundance of SynAce01 in merged metagenomes.

Table 4.1 List of Ace Lake metagenomes used for FR analysis of SynAce01. [A] The 3–20 and 0.8–3 µm-filter metagenomes from a lake depth and time period were combined to prepare the merged metagenomes shown in column three. [B] The relative abundance of *Synechococcus* in the selected metagenomes was calculated using the method described in Chapter 3 section 3.2.1. [C] The number of reads represents the total number of reads in the merged metagenomes. For comparative analysis, the selected metagenomes represented data from the upper oxic (Upper 3), oxycline (Interface), and lower anoxic zones (Lower 1, 2, 3) of Ace Lake as well as from summer (Dec), winter (Aug), and spring (Oct, Nov) seasons.

Lake depth	Time period	Merged metagenome name[A]	*Synechococcus* OTU relative abundance (%)[B]		Number of reads[C]
			3–20 µm-filter	0.8–3 µm-filter	

Upper 3	Nov 2008	Nov2008_U3	2	8	137,274,340
	Aug 2014	Aug2014_U3	12	16	52,404,196
	Oct 2014	Oct2014_U3	11	32	44,347,552
	Dec 2014	Dec2014_U3	19	44	47,658,964
Interface	Oct 2014	Oct2014_I	11	25	49,439,564
Lower 1	Dec 2014	Dec2014_L1	3	5	44,499,368
Lower 2	Dec 2014	Dec2014_L2	4	6	53,825,658
Lower 3	Nov 2013	Nov2013_L3	8	3	43,627,018

4.2.3 Phylogeny assessment

For the phylogenetic analysis of the Ace Lake *Synechococcus*, the *16S rRNA* genes from the MAGs and various species of marine cyanobacteria were used (Table 4.2). The gene sequences were aligned in MEGA X v10.1.7 software using ClustalW algorithm. The alignments were used to generate a maximum likelihood tree in MEGA X with default parameters and 1,000 bootstrap values.

Table 4.2 Marine cyanobacteria species used in the phylogenetic analysis of Ace Lake *Synechococcus*. [A] The accession IDs of the *16S rRNA* genes or the species genomes are provided in the last column. * The table includes the *16S rRNA* gene from SynAce01 as well as two distinct *16S rRNA* genes identified in the *Synechococcus* MAGs (referred to as 16S rRNA AL1 and 16S rRNA AL2; described below in section 4.3.2).

Organism	Length (in bp)	Accession ID[A]
Anabaena oscillarioides	1,418	AJ630426.1
Anabaenopsis elenkinii	1,461	KM020015.1
Chlorogloeopsis fritschii	1,149	NR_112176.1
Cyanobium gracile	1,476	NR_102447.1
Leptolyngbya valderiana	1,349	KY807918.1
Lyngbya cf. confervoides	1,331	AY599507.1
Microcoleus antarcticus	1,395	AF218373.1
Microcoleus glaciei	1,394	AF218374.1
Microcystis aeruginosa	1,489	NR_074314.1
Nodularia spumigena	1,438	NR_112106.1
Nostoc commune	1,446	AB088375.2
Oscillatoria princeps	1,367	AB045961.1
Prochlorococcus marinus subsp. *marinus*	1,483	NC_005042.1

Prochloron didemni	1,009	MT254065.1
Prochlorothrix hollandica	1,408	AJ007907.1
Prochlorothrix scandica	1,236	HQ316169.1
Synechococcus elongatus	1,489	NR_074309.1
Synechococcus sp. PCC7001	1,411	AB015058.1
Synechococcus sp. PCC7002	1,452	AJ000716.1
Synechococcus sp. PCC7003	1,414	AB015059.
Synechococcus sp. PCC7117	1,411	AB015060.1
Synechococcus sp. PCC73109	1,413	AB015061.1
Synechococcus sp. PCC7335	1,410	AB015062.1
Synechococcus sp. WH5701	1,440	AY172832.1
Synechocystis sp. PCC6803	1,238	AY224195.1
Synechococcus sp. SynAce01*	1,486	NZ_CP018091.1
Synechococcus (16S rRNA AL1)*	1,489	IMG taxon ID: 3300022857 Gene locus tag: Ga0222653_10001424
Synechococcus (16S rRNA AL2)*	1,489	IMG taxon ID: 3300023253 Gene locus tag: Ga0222695_100009760

4.2.4 Analysis of *Synechococcus* defence system genes

The annotated genes on the contigs of *Synechococcus* MAGs were manually parsed to assess the presence/absence and number of genes associated with various defence systems such as R-M system, DISARM, BREX system, and T-A system (specifically ABI mechanism) (Table 4.3). The gene assignments were verified by aligning them against reference proteins from the UniProtKB/Swiss-Prot database using the ExPASy BLAST+ online service (https://web.expasy.org/blast/). The verified defence genes were further assessed to identify the defence system subtype (Table 4.3). The CRISPR-Cas defence system was not investigated, as Ace Lake *Synechococcus* did not have *cas* genes (Chapter 3 section 3.3.5.5).

Table 4.3 Prokaryotic defence systems investigated in Ace Lake microbes. [A] The arrangement of defence genes in CRISPR-Cas defence gene clusters are shown in column three. Notably, CRISPR-Cas system genes were not identified in Ace Lake *Synechococcus* (Chapter 3 section 3.3.5.5), but Ace Lake *Chlorobium* contained CRISPR-Cas defence genes (described below in Chapter 5 section 5.4.3). [B] The data for defence system subtypes and the genes involved in them were taken from the publications cited in the last column. *cas*, CRISPR-associated gene; *dinG*, ATP-dependent DNA helicase; LS, leader sequence; *RT*, reverse

transcriptase; *tnsABCD*, Transposon Tn7 transposition genes; TPR, tetratricopeptide repeat; *tracrRNA*, transactivating CRISPR RNA; *wyl*, WYL-domain encoding gene.

Defense system	System subtype	Defense genes[A]	References[B]
CRISPR-Cas (class 1)	I-A	*cas6*, *cas11*, *cas7*, *cas5*, *cas8a1*, *cas3'*, *cas3''*, *cas2*, *cas4*, *cas1*, *cas4*	Makarova et al, 2020
	I-B	*cas6*, *cas8b1*, *cas7*, *cas5*, *cas3*, *cas4*, *cas1*, *cas2*	
	I-C	*cas3*, *cas5*, *cas8c*, *cas7*, *cas4*, *cas1*, *cas2*	
	I-D	*cas3'*, *cas3''*, *cas10d*, *cas7*, *cas5*, *cas6*, *cas4*, *cas1*, *cas2*	
	I-E	*cas3*, *cas8e*, *cas11*, *cas7*, *cas5*, *cas6*, *cas1*, *cas2*	
	I-F1	*cas1*, *cas2*, *cas3*, *cas8f1*, *cas5f1*, *cas7f1*, *cas6f*	
	I-F2	*cas1*, *cas2*, *cas3*, *cas7f2*, *cas5f2*, *cas6f*	
	I-F3	*tnsA*, *tnsB*, *tnsC*, *tnsD*, *cas8f3*/*cas5f3*, *cas7f3*, *cas6f*	
	I-G	*cas3*, *cas8u2*, *cas7*, *cas5*, *cas6*, *cas4*, *cas1*, *cas2*	
	III-A	*cas6*, *cas10*, *cas11*, *cas7*, *cas5*, *cas7*, *csm6*, *cas1*, *cas2*	
	III-B	*cas7*, *cas10*, *cas5*, *cas7*, *cas11*, *cas6*, *cas7*	
	III-C	*cas7*, *cas7*, *cas10*, *cas7*, *cas11*, *cas5*	
	III-D	*cas10*, *cas7*, *cas5*, *cas11*, *cas7*, *cas7*, *csx19*, *cas7*	
	III-E	TPR + caspase, *cas7*, *cas11*, *cas7*, *cas7*, *RT*, *cas1*, *cas2*	
	III-F	*cas10*, *cas5*, *cas11*, *cas7*	
	IV-A	*dinG*, *cas6*, *cas8*-like, *cas7*, *cas5*	
	IV-B	*cysH*-like, *cas8*-like, *cas11*, *cas7*, *cas5*	
	IV-C	LS, *cas11*, *cas7*, *cas5*	
CRISPR-Cas (class 2)	II-A	*cas9*, *cas1*, *cas2*, *csn2*, *tracrRNA*	Makarova et al, 2020
	II-B	*cas9*, *cas1*, *cas2*, *cas4*, *tracrRNA*	
	II-C1	*cas9*, *cas1*, *cas2*, *tracrRNA*	
	II-C2	*cas9*, *tracrRNA*, *cas4*, *cas2*, *cas1*	
	V-A	*cas12a*, *cas4*, *cas1*, *cas2*	

	V-B1	*cas12b1*, *cas4*, *cas1*, *cas2*, *tracrRNA*	
	V-B2	*cas4*, *cas1*, *cas2*, *cas12b2*, *tracrRNA*	
	V-C	*cas1*, *cas12c*	
	V-D	*cas1*, *cas12d*	
	V-E	*cas12e*, *cas4*, *cas1*, *cas2*, *tracrRNA*	
	V-F1	*cas1*, *cas2*, *cas4*, *cas12f1*, *tracrRNA*	
	V-F2	*cas12f2*, *cas1*, *cas2*, *cas4*	
	V-F3	*cas1*, *cas2*, *cas4*, *cas12f3*	
	V-G	*cas12g*, *tracrRNA*	
	V-H	*cas12h*	
	V-I	*cas12i*	
	V-K (V-U5)	*tnsB*, *tnsC*, *tniQ*, *cas12k*, *tracrRNA*	
	V-U1	*c2c4*	
	V-F1 (V-U3)	*c2c10*	
	V-U2	*c2c8*	
	V-U4	*c2c9*	
	VI-A	*cas13a*, *cas1*, *cas2*	
	VI-B1	*cas13b1*, *csx28*	
	VI-B2	*csx27*, *cas13b2*	
	VI-C	*cas13c*	
	VI-D	*wyl*, *cas13d*, *cas1*, *cas2*	
R-M	Type I	methyltransferase, restriction endonuclease	Koonin et al, 2017
	Type II	restriction, modification and specificity subunits	
	Type III	restriction subunit, methyltransferase	
	Type IV	AAA+ family GTPase, restriction endonuclease	
BREX	Type 1	*brxA*, *brxB*, *brxC*, *pglX*, *pglZ*, *brxL*	Goldfarb et al, 2015
	Type 2	*pglW*, *pglX*, *pglY*, *pglZ*, *brxD*, *brxHI*	
	Type 3	*brxF*, *brxC*/*pglY*, *pglXI*, *brxHII*, *pglZ*, *brxA*	
	Type 4	*brxP*, *brxC*/*pglY*, *pglZ, brxL*	
	Type 5	*brxA*, *brxC*/*pglY*, *brxB*, *brxC*/*pglY*, *pglX*, *pglZ*, *brxHII*	

	Type 6	*brxE*, *brxA*, *brxB*, *brxC*/*pglY*, *pglX*, *pglZ*, *brxD*, *brxHI*	
DISARM	Class 1	*drmD*, *drmMI*, *drmA*, *drmB*, *drmC*	Ofir et al, 2018
	Class 2	*drmE*, *drmA*, *drmB*, *drmC*, *drmMII*	
T-A	Type I, II, III, IV, V, VI, ABI systems	Numerous T-A system genes have been identified, including those involved in ABI mechanism	Yamaguchi et al, 2011; Koonin et al, 2017; Lopatina et al, 2020

4.3 Results

4.3.1 Overview of *Synechococcus* MAGs and SynAce01 genome

A total of 59 *Synechococcus* MAGs (~157 Mbp; Appendix A: Table A2) generated from 120 Ace Lake time-series of metagenomes (Appendix A: Table A1) were used for various analyses in this chapter (Appendix A: Table A1). Notably, all *Synechococcus* MAGs were from 3–20 μm-filter and 0.8–3 μm-filter metagenomes and none were from 0.1–0.8 μm-filter metagenomes, which was consistent with the size partitioning of the *Synechococcus* OTU (Chapter 3 section 3.3.5.5). The *16S rRNA* marker gene was identified in 19 *Synechococcus* MAGs. The genomic variation, viral defence potential, ANI, and AAI of the MAGs were explored using a total of 81,361 genes on 2,259 *Synechococcus* contigs. These genes and contigs were from 25 MAGs with ≥99% genome completeness as well as a *Synechococcus* MAG with 97% genome completeness and containing a distinct *16S rRNA* gene (16S rRNA AL2) (described below in section 4.3.2). The *Synechococcus* MAGs used for the preliminary analysis of genomic variation represented high- and medium-quality draft genomes that were nearly complete (≥99% genome completeness) and contained little contamination (≤4% bin contamination). Bin contamination indicates the number of multicopy marker genes in each marker set that was identified in a MAG, and it can represent copies of the marker genes from different strains that have been included in the MAG (Parks et al, 2015).

The complete genome of SynAce01 is available in NCBI (RefSeq ID: NZ_CP018091.1). The method used for the sequencing and assembly of SynAce01 complete genome was rigorous (Tang et al, 2019). It included the use of Illumina as well as PacBio technologies for generating two sets of contig assemblies, which were then compared to generate a closed genome. A post-genome assembly error correction step was included to correct any sequencing or assembly errors (Tang et al, 2019). Therefore, SynAce01 genome was considered to be of good quality and was used for the preliminary and FR analyses of genomic variation in Ace Lake *Synechococcus*. SynAce01 genome was 2,750,634 bp long, with 63.9% GC content. Overall, it contained 2,881 annotated genes, of which 2,732 were protein coding genes.

4.3.2 *Synechococcus 16S rRNA* gene identity, ANI, AAI, and phylogeny

The *16S rRNA* gene sequences taken from 19 *Synechococcus* MAGs were 1,489 bp length each, but only 18 of the 19 genes had identical sequences; this gene sequence is hereafter referred to as 16S rRNA AL1. The variant *16S rRNA* gene sequence (hereafter referred to as 16S rRNA AL2) was from a *Synechococcus* MAG generated from Dec 2014_Lower 1_3 µm-filter metagenome; it was 99.9% similar to 16S rRNA AL1 sequence with SNPs at positions 217 (A→T transversion) and 231 (G→T transversion) (Figure 4.1).

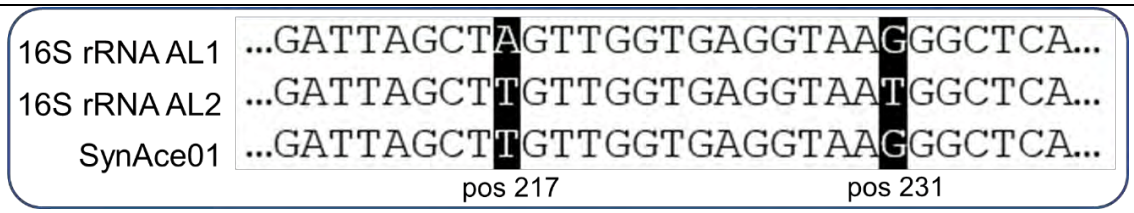


Figure 4.1 Differences in the sequence of *16S rRNA* genes from Ace Lake *Synechococcus*. The figure highlights the SNPs observed in the *16S rRNA* gene sequences from SynAce01 and *Synechococcus* MAGs (16S rRNA AL1 and 16S rRNA AL2). The positions (pos) indicate base positions on *16S rRNA* genes from *Synechococcus* MAGs, all of which were 1,489 bp long. The length of the SynAce01 *16S rRNA* gene was 1,486 bp.

The taxonomic analysis of Ace Lake OTUs had showed that the closest related species to the Ace Lake S*ynechococcus* OTU was SynAce01, with 99.9% *16S rRNA* gene identity and 99% ANI over 97% alignment fraction (Chapter 3 section 3.3.3). This relatedness was verified for the Ace Lake *Synechococcus* MAGs from different lake depths and time periods, by comparing their *16S rRNA* genes with that of SynAce01. The analysis showed that 16S rRNA AL1 and 16S rRNA AL2 were both 99.7% similar

to SynAce01 *16S rRNA* gene, with 3 additional nucleotides at their sequence ends and a SNP each — 16S rRNA AL1: position 217 T→A transversion and 16S rRNA AL2: position 231 G→T transversion (Figure 4.1). The *16S rRNA* gene-based phylogenetic analysis showed distinct clustering of *Synechococcus* marker sequences from Ace Lake, separate from all other marine cyanobacteria species analysed (Figure 4.2). The overall ANI of the *Synechococcus* MAGs was ≥99.4% when compared against each other and ≥99.2% (over 89–96% alignment fraction) when compared against the SynAce01 genome. The AAI of the MAGs against the SynAce01 proteome was ≥99.4% over 73–83% alignment fraction.

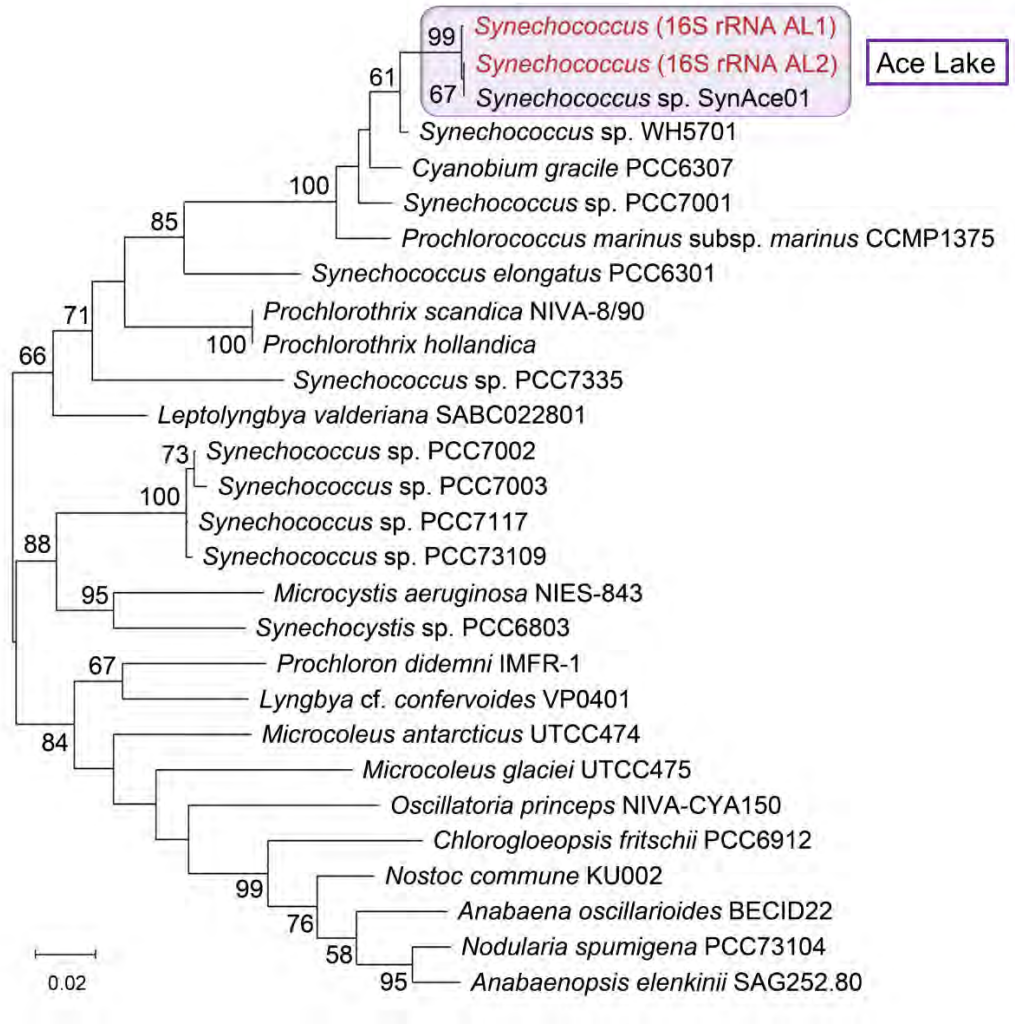


Figure 4.2 *16S rRNA* gene-based phylogenetic analysis of Ace Lake *Synechococcus*. The maximum-likelihood tree shows the *16S rRNA* gene-based phylogeny of various marine cyanobacteria (Table 4.2). The red-highlighted *16S rRNA* gene sequences (16S rRNA AL1 and 16S rRNA AL2) were from *Synechococcus* MAGs (described in section 4.3.2). The tree is drawn to scale and the scale length indicates the branch length. The numbers next to the

branches represent bootstrap values showing the percentage of trees in which the taxa clustered together. Only bootstrap values greater than 50% are shown here.

4.3.3 Analysis of sequence variations between *Synechococcus* MAGs

The alignment of all MAGs against the MAG with the highest total base pair count and ≥99% genome completeness showed at least 60 regions with SNPs, with the lengths of the variable sequence regions ranging from 300 bp to 34 kb. Most of the genes in these variable sequence regions were either of unknown function or coded for enzymes involved with mobile genetic elements (transposases). Genes predicted to be involved in cell wall biosynthesis (mostly glycosyltransferases) were also prevalent in the variable sequence regions. Some of the genes coded for membrane-associated proteins (Sec-translocase, efflux pump membrane proteins, outer membrane protein TolC, undecaprenyl-diphosphatase, HPP family protein, porins, type IV pilus assembly proteins) and substrate transporters (multiple sugar ABC transporter, MFS transporter family proteins, iron transporter FeoB, putative sulfate transporters, putative bicarbonate transporter, and putative ion antiporters). A few genes involved in metabolism and cell defence were also identified in the variable sequence regions.

4.3.4 Comparative analysis of *Synechococcus* MAGs and SynAce01

As SynAce01 and the *Synechococcus* MAGs were from Ace Lake, they were used for preliminary analysis of genomic variation in the Ace Lake *Synechococcus.* The alignment of the *Synechococcus* MAGs against the SynAce01 genome showed that the MAGs did not have a similar pattern of sequence alignment, with only two MAGs matching across 100% alignment fraction of the reference genome and multiple regions with alignment gaps (Figure 4.3). The genes identified in the most prominent alignment gaps mostly coded for enzymes associated with mobile elements (transposases), proteins of unknown functions (hypothetical and uncharacterised proteins), and cellular defence proteins (T-A system proteins) (Table 4.4). Apart from the alignment gaps, a few regions of the *Synechococcus* MAGs had low identity matches to the SynAce01 genome. One of these low identity regions (72–78% identity) starting at around 933 kb length of the SynAce01 genome was ~36 kb long and contained genes involved in cell membrane fluidity (two fatty acid desaturases) and substrate transport (three zinc ABC transporter proteins). Another region with low identity matches (73–81% identity) starting at around 1.3 Mb length of the SynAce01 genome contained genes potentially

involved in cell wall modification (two glycogen/starch/alpha-glucan phosphorylases and a mannose-1-phosphate guanylyltransferase/mannose-6-phosphate isomerase). Apart from the alignment gaps and low identity regions, some of the *Synechococcus* MAG contigs did not match the SynAce01 genome sequence. The genes annotated on these MAG contigs were mostly of unknown function (hypothetical proteins) and some were associated with cell wall modification (glycosyltransferases), mobile elements (transposases), and cellular defence systems (R-M and T-A system proteins). Genes involved in transport of substrates to and from the cell (ABC transporter and MFS transporter proteins) were also identified in these MAG contigs.

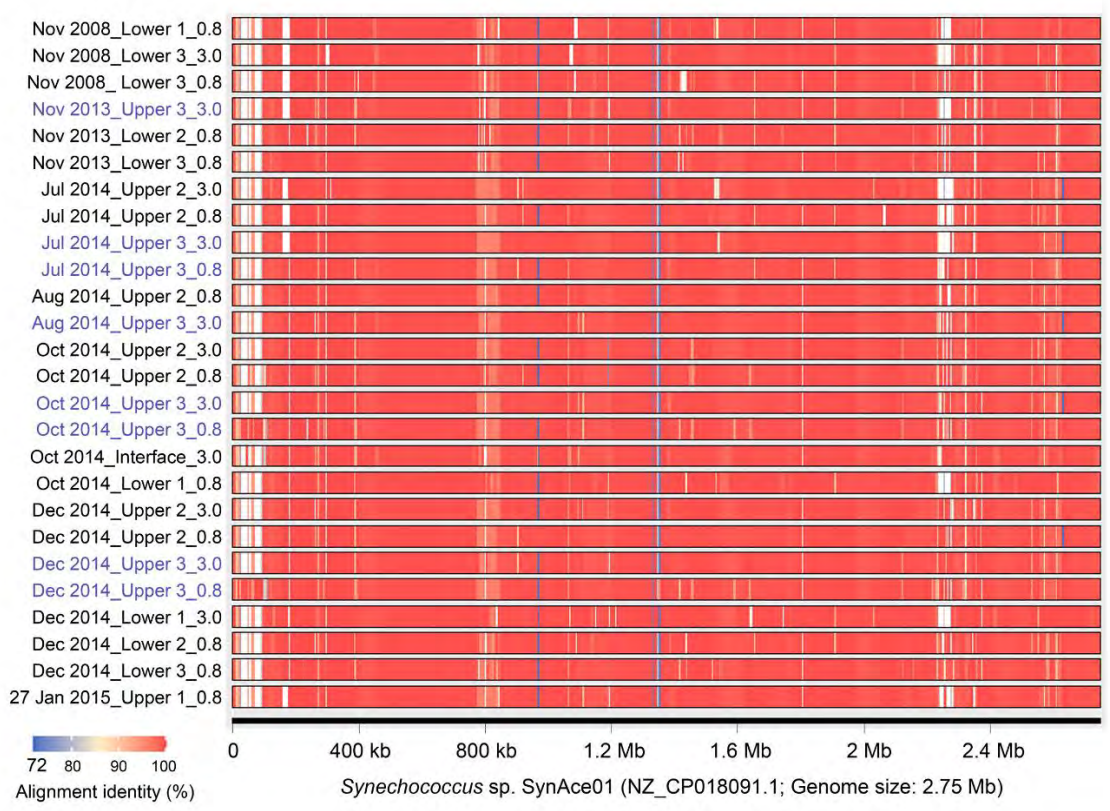


Figure 4.3 Alignment of *Synechococcus* MAGs against SynAce01 genome. The figure shows the alignment of 26 *Synechococcus* MAGs to the SynAce01 genome (x-axis, black line). The y-axis indicates the metagenomes (sample collection time, lake depth, and filter fraction) from which the MAGs were generated. The metagenomes are arranged from top to bottom in the order of sample collection time period from 2008 to 2015 and lake depth from Upper 1, 2, 3 to Interface to Lower 1, 2, 3. The genome completeness of the *Synechococcus* MAGs shown here was ≥99%, except for the MAG from 'Dec 2014_Lower 1_3.0' that had a genome completeness of 97% and contained 16S rRNA AL2 gene (section 4.3.2). The white regions in the alignment bands depict alignment gaps and indicate that the MAGs had no matches to those regions of the reference genome (section 4.3.4). The gradient bar indicates the percentage alignment identity.

The blue highlighted labels represent MAGs from Upper 3 zone of Ace Lake, where the population of *Synechococcus* was high in most time periods (Chapter 3 section 3.3.3). Filter fraction: 3, 3–20 μm; 0.8, 0.8–3 μm.

Table 4.4 Genes annotated on SynAce01 genomic regions associated with alignment gaps in *Synechococcus* MAGs. [A] The approximate starting positions and lengths of the MAG alignment gaps on the SynAce01 genome are provided in the second column (the alignment gaps can be seen as white regions in Figure 4.3). The regions are arranged from top to bottom in the order of their occurrence along the length of SynAce01 genome.

MAGs in which observed	Alignment gap starting position and length[A]	SynAce01 genes annotated in the alignment gap
All MAGs except Oct 2014_Upper 3_0.8 and Dec 2014_Upper 3_0.8)	~26 kb (22 kb length)	3 copies of DEAD/DEAH box helicases RES family NAD+ phosphorylase 3 Uncharacterized proteins 2 Hypothetical proteins TerB family tellurite resistance protein Class I SAM-dependent DNA methyltransferase
	~50kb (11 kb length)	AbrB family transcriptional regulator Thermonuclease family protein 2 copies of GNAT family N-acetyltransferases YjbQ family protein CDGSH iron-sulfur domain-containing protein Flavin reductase family protein Alpha/beta hydrolase 4 Hypothetical proteins 3 Uncharacterized proteins Transposase
	~68 kb (22 kb length)	19 Hypothetical proteins Terminase An uncharacterised protein 3'-5' exonuclease 2 copies of Helix-turn-helix domain-containing proteins 3 copies of Transposases C1 family peptidase IS5/IS1182 family transposase

	~93 kb (2.3 kb length)	2 Hypothetical proteins
8 out of 26 MAGs	~155 kb (23 kb length)	2 copies of IS481 family transposase Bile acid:sodium symporter family protein (a transmembrane protein) Rhodanese-like domain-containing protein 5 Uncharacterized proteins Conjugal transfer protein TrbI Deoxyribodipyrimidine photo-lyase FAD-dependent oxidoreductase 2 copies of Type II T-A system PemK/MazF family toxin IS5 family transposase Putative addiction module antidote protein Type II T-A system RelE/ParE family toxin RNA-directed DNA polymerase 4 Hypothetical proteins IS1595 family transposase
All MAGs	~2.3 Mb (76 kb length)	4 copies of type II T-A system VapBC toxin 3 Hypothetical proteins 2 copies of ISAs1 family transposase Uma2 family endonuclease (putative restriction endonuclease) 2 copies of N-acetylmuramoyl-L-alanine amidase AbrB/MazE/SpoVT family DNA-binding domain-containing protein (antitoxin component of the Phd-Doc family type II T-A system) Cellulose synthase catalytic subunit 2 Uncharacterized protein AI-2E family transporter IS1595 family transposase Fatty acid desaturase Rhomboid family intramembrane serine protease IS30 family transposase

4.3.5 FR analysis of SynAce01 in Ace Lake metagenomes

The recruitment analysis of reads from Ace Lake merged metagenomes from different time periods and lake depths to the SynAce01 genome was used for a more in-depth analysis of *Synechococcus* genomic variation, including variable coverage regions and SNPs. The coverage pattern of SynAce01 was similar to the relative abundance pattern of the *Synechococcus* OTU in the merged metagenomes, showing high coverage in the upper oxic zone compared to the lower anoxic zone (Figures 4.4a, b, c). The abundance of *Synechococcus* was generally low in the Interface zone of Ace Lake (<6%; Chapter 3 Figure 3.6), except in the metagenomes from Oct 2014 used here for FR analysis.

4.3.5.1 Variable coverage regions

The alignment of metagenomic reads to SynAce01 genome showed presence of regions with variable coverage — seven LCRs and 14 high coverage regions (Figure 4.4d). The high coverage regions mainly contained genes associated with mobile elements (transposases) and a few duplicate genes (rRNAs and photosystem-associated proteins) (Table 4.5). The LCRs contained genes involved in a variety of functions including DNA/RNA/protein modification, DNA replication and repair, cell wall biosynthesis, assembly, and modification, metabolism, substrate transport, as well as cell defence (Table 4.5). However, most of the genes in the LCRs were of unknown function or were associated with mobile elements. Genes associated with transfer RNAs and rRNAs were also present in the LCRs along with a bacteriophage-associated gene (terminase). *Synechococcus* had two *16S rRNA* genes, of which one was present in the LCR starting at ~1.8 Mb position on SynAce01 genome, whereas the other was in a high coverage region starting at ~800 kb position on SynAce01 genome (Table 4.5). The high coverage *Synechococcus 16S rRNA* gene was an inverted duplicate of the low coverage *16S rRNA* gene in SynAce01, and variable read depth is often associated with this sequence configuration (http://software.broadinstitute.org/software/igv/interpreting_pair_orientations).

4.3.5.2 SNPs

For the analysis of SNPs in *Synechococcus* from different time periods and lake depths, only the mutations that were present in at least 90% of the metagenomic reads aligned to the reference base were considered. A total of 494 out of 2881 SynAce01 genes showed at least one SNP in at least one merged metagenome. Nearly all of these SNPs were observed in regions where the read depth did not vary, i.e., non-variable coverage

regions, and almost one-fifth of the genes containing SNPs (103 out of 494) were of unknown function (hypothetical and uncharacterised proteins). The rest of the genes were mainly associated with various metabolic functions, substrate transport, DNA/RNA/protein modification, cell wall biosynthesis and modification, and cell defence.

For the analysis of sequence variations in *16S rRNA* gene of *Synechococcus*, the marker gene sequence in the high coverage region (starting at ~800 kb position on SynAce01 genome) was examined. Notably, the SNP at position 217 of *16S rRNA* gene (Figure 4.1) was present in a larger *Synechococcus* population (79–96% of the reads aligned to the reference base) than the SNP at the position 231 (6–21% of the reads aligned to the reference base) in all merged metagenomes, except the metagenome from Lower 1. In Dec 2014 Lower 1 merged metagenome, the SNP at position 217 was present in 53% of the reads aligned to the reference base, whereas the SNP at position 231 was present in 47% of the aligned reads. On closer inspection, it was observed that the two SNPs rarely occurred on the same reads, suggesting that they probably represented data from two different subpopulations of *Synechococcus*, of which the ones carrying the mutation at position 217 were more prevalent. This was consistent with the observation that 18 out of 19 *Synechococcus* MAGs had the *16S rRNA* gene mutation at position 217 (16S rRNA AL1 containing T→A transversion; Figure 4.1), whereas the remaining MAG had the *16S rRNA* gene mutation at position 231 (16S rRNA AL2 containing T→G transversion; Figure 4.1).

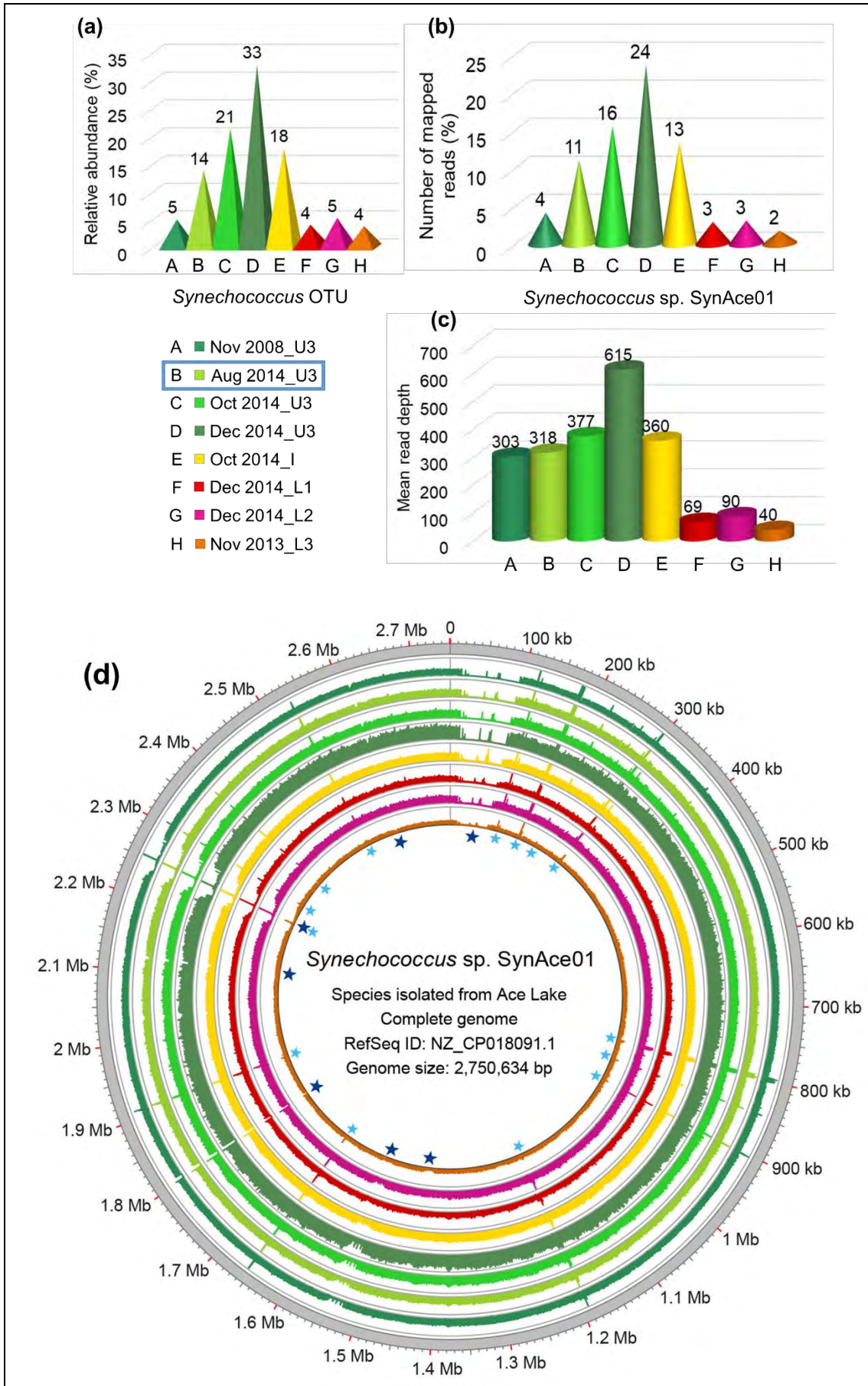

(a)
Relative abundance (%)
35
30
25
20
15
10
5
0
5
14
21
33
18
4
5
4
A B C D E F G H
Synechococcus OTU
(b)
Number of mapped reads (%)
25
20
15
10
5
0
4
11
16
24
13
3
3
2
A B C D E F G H
Synechococcus sp. SynAce01
A Nov 2008_U3
B Aug 2014_U3
C Oct 2014_U3
D Dec 2014_U3
E Oct 2014_I
F Dec 2014_L1
G Dec 2014_L2
H Nov 2013_L3
(c)
Mean read depth
700
600
500
400
300
200
100
0
303
318
377
615
360
69
90
40
A B C D E F G H
(d)
2.7 Mb
0
100 kb
200 kb
300 kb
400 kb
500 kb
600 kb
700 kb
800 kb
900 kb
1 Mb
1.1 Mb
1.2 Mb
1.3 Mb
1.4 Mb
1.5 Mb
1.6 Mb
1.7 Mb
1.8 Mb
1.9 Mb
2 Mb
2.1 Mb
2.2 Mb
2.3 Mb
2.4 Mb
2.5 Mb
2.6 Mb
Synechococcus sp. SynAce01
Species isolated from Ace Lake
Complete genome
RefSeq ID: NZ_CP018091.1
Genome size: 2,750,634 bp

Figure 4.4 *Synechococcus* abundance, coverage distribution, and genomic variation in Ace Lake metagenomes from different lake depths and time periods. (a) The bar-chart shows the relative abundance distribution of *Synechococcus* OTU (coloured pyramids) in merged metagenomes from different lake depths (U3, I, L1, L2, L3) and seasons (summer, Dec; winter, Aug; spring, Oct and Nov) (x-axis). *Synechococcus* OTU relative abundances in merged metagenomes were calculated from the absolute abundances of *Synechococcus* contigs in 3–20 and 0.8–3 µm-filter metagenomes relative to the total abundance of all contigs in the two metagenomes (method described in Chapter 3 section 3.2.1). In the colour key, the merged metagenome from Aug 2014 is shown in a blue box to highlight its winter origin. **(b and c)** The bar-charts show the number of mapped reads (**b**, coloured cones) and mean read depth (**c**, coloured cylinders) of SynAce01 in the Ace Lake merged metagenomes from different lake depths and time periods. The y-axis in **(b)** indicates the total number of reads that aligned to the SynAce01 genome, whereas the y-axis in **(c)** denotes the mean of read depths from each position on SynAce01 genome. Read depth values were calculated from the data in the base coverage files generated using BBMap v38.51 (section 4.2.2). **(d)** The circos plot depicts the coverage distribution of SynAce01 genome in Ace Lake merged metagenomes (coloured rings). The outermost ring (grey) depicts the backbone of SynAce01 genome and the x-axis scale for all rings is drawn around its circumference. Read depth values are plotted on linear scale y-axes. As the main purpose of the figure was to highlight the distribution of variable coverage regions, the y-axes scales of circos rings vary. Read depth values greater than the y-axes limits were truncated to the y-axis maximum values. Variable coverage regions are marked by the dark blue (low coverage) and light blue (high coverage) stars inside the innermost ring of the circos plot. Merged metagenomes and their y-axis scale ranges, outer to inner ring: Nov 2008_U3 (A, ■, 0–1000); Aug 2014_U3 (B, ■, 0–1000); Oct 2014_U3 (C, ■, 0–1000); Dec 2014_U3 (D, ■, 0–1000); Oct 2014_I (E, ■, 1–1000); Dec 2014_L1 (F, ■, 0–300); Dec 2014_L2 (G, ■, 0–300); Nov 2013_L3 (H, ■, 0–300). Lake depths: U3, Upper 3; I, Interface; L1, Lower 1; L2, Lower 2; L3, Lower 3.

Table 4.5 Genes annotated on variable coverage regions of SynAce01 genome. [A] The approximate starting positions and lengths of the variable coverage regions on the SynAce01 genome are provided in the first column (the variable coverage regions are labelled as blue stars in Figure 4.4d). The LCRs are shown with a light blue background colour, whereas high coverage regions are shown with a light orange background colour. [B] The metagenomes mentioned in the second column refer to the merged metagenomes used for FR analysis of SynAce01 — Nov 2008 Upper 3, Aug 2014 Upper 3, Oct 2014 Upper 3, Dec 2014 Upper 3, Oct 2014 I, Dec 2014 Lower 1, Dec 2014 Lower 2, Nov 2013 Lower 3 (Table 4.1). The regions are

arranged from top to bottom in the order of their occurrence along the length of SynAce01 genome.

Starting position and length of variable coverage region[A]	Metagenomes in which observed[B]	SynAce01 genes annotated in the variable coverage region
~20 kb (87 kb length)	All	46 Hypothetical proteins 9 Uncharacterised proteins 3 copies of Helix-turn-helix domain-containing protein 3 copies of DEAD/DEAH box helicase 2 copies of GNAT family N-acetyltransferase 4 Transposases IS5/IS1182 family transposase IS1182 family transposase Collagen-like protein C39 family peptidase C1 family peptidase RES family NAD+ phosphorylase TerB family tellurite resistance protein Class I SAM-dependent DNA methyltransferase Thermonuclease family protein YjbQ family protein CDGSH iron-sulfur domain-containing protein Flavin reductase family protein Alpha/beta hydrolase Peptide-methionine (S)-S-oxide reductase MsrA LysM peptidoglycan-binding domain-containing protein Terminase 3'-5' Exonuclease
~118 kb (939 bp length)	All	IS481 family transposase
~176 kb	All	IS1595 family transposase

(954 bp length)		
~223 kb (938 bp length)	All	IS481 family transposase
~297 kb (2 kb length)	All	IS30 family transposase Transposase
~796 kb (5 kb length)	All	23S ribosomal RNA tRNA-Ala tRNA-Ile 16S ribosomal RNA
~846 kb (1 kb length)	All	Photosystem II D2 protein (photosystem q(a) protein)
~902 kb (1 kb length)	All	IS3 family transposase
~1.19 Mb (1 kb length)	All except Nov 2013 Lower 3	IS5 family transposase
~1.42 Mb (18 kb length)	Only Nov 2013 L3	2 copies of GDP-mannose 4,6-dehydratase 3 copies of Glycosyltransferases Glycosyltransferase family 4 protein 2 copies of ISAs1 family transposase IS66 family transposase GNAT family N-acetyltransferase tRNA-Gly FkbM family methyltransferase SAM-dependent methyltransferase Methyltransferase domain-containing protein ABC transporter ATP-binding protein (O-antigen export system ATP-binding protein RfbB) ABC transporter permease (O-antigen export system permease protein RfbA) Hypothetical protein
~1.52 Mb (21 kb length)	All except Dec 2014 Lower 1 and Nov 2013 Lower 3	4 Hypothetical proteins 3 copies of Glycosyltransferases Glycosyltransferase family 4 protein Glycosyltransferase family 2 protein

		ABC transporter ATP-binding protein (Lipid A export ATP-binding/permease protein MsbA) O-antigen ligase family protein
~1.65 Mb (1 kb length)	All	Photosystem II q(b) protein
~1.81 Mb (5 kb length)	All	16S ribosomal RNA tRNA-Ile tRNA-Ala 23S ribosomal RNA
~1.9 Mb (1.5 kb length)	All except Nov 2013 Lower 3	IS5 family transposase
~2.12 Mb (4 kb length)	Aug 2014 Upper 3, Oct 2014 Upper 3, Dec 2014 Upper 3, and Oct 2014 Interface	3 Hypothetical proteins DNA polymerase III subunit gamma/tau
~2.23 Mb (44 kb length)	All	9 Hypothetical proteins 11 copies of Glycosyltransferases Glycosyltransferase family 2 protein Glycosyltransferase family 4 protein 2 copies of IS1595 family transposase 2 copies of ISAs1 family transposase IS3 family transposase 4 copies of FkbM family methyltransferase 4 copies of type II toxin-antitoxin system VapC family toxin Asparagine synthase (glutamine-hydrolyzing) Class I SAM-dependent methyltransferase Alpha-1,2-fucosyltransferase Glycoside hydrolase family 99-like domain-containing protein Polysaccharide pyruvyl transferase family protein Nitroreductase family protein ABC transporter ATP-binding protein (Lipid A export ATP-binding/permease protein MsbA) Uma2 family endonuclease

		N-acetylmuramoyl-L-alanine amidase AbrB/MazE/SpoVT family DNA-binding domain-containing protein
~2.25 Mb (1.6 kb length)	All	IS3 family transposase
~2.3 Mb (1 kb length)	All	Photosystem II D2 protein (photosystem q(a) protein)
~2.38 Mb (380 kb length)	All except Nov 2008 Upper 3 and Nov 2013 Lower 3	Hypothetical protein
~2.53 Mb (1 kb length)	All except Nov 2013 Lower 3	IS3 family transposase
~2.61 Mb (1 kb length)	All Upper 3	Photosystem II q(b) protein

4.3.6 Defence genes in Ace Lake *Synechococcus*

The genes annotated in *Synechococcus* MAGs were manually parsed to identify the defence genes (Table 4.6). *Synechococcus* MAGs contained multiple copies of the genes that coded for the methyltransferase, sequence specificity, and restriction endonuclease subunits of a type I R-M system. The genes coding for a type II R-M system methylase subunit as well as a type III R-M system restriction enzyme were also identified in the *Synechococcus* MAGs, but most of these had low identity matches to the reference proteins in the UniProtKB/Swiss-Prot database (Table 4.6). Some MAG gene annotations matched previously reported DISARM gene annotations (Ofir et al, 2018). For example, gene annotations similar to *drmD* (SNF2 family helicase), *drmC* (phospholipase D), and *drmMII* (5-cytosine DNA methyltransferase) were identified among the *Synechococcus* MAG gene annotations (Table 4.6). The verification of the gene functions by matching their proteins to UniProtKB/Swiss-Prot database showed that the functional annotation of phospholipase D was correct, but it was not specifically associated with DISARM. The genes annotations of SNF2 family helicases and 5-cytosine DNA methyltransferase showed that they were associated with different helicases and methylases and neither were specifically associated with DISARM mechanism (Table 4.6).

Genes associated with a variety of T-A systems, including multiple copies of the genes coding for VapBC, MazEF, and HigAB type II T-A systems, were found in the MAGs, along with the genes coding for the YefM antitoxin and HipA, PemK and RelE toxins of type II T-A systems (Table 4.6). Previous studies have reported that HEPN domain containing T-A system proteins could be involved in the ABI mechanism of viral infection disruption (Koonin et al, 2017). Therefore, genes coding for HEPN domain containing proteins were identified in the *Synechococcus* MAGs, but none of them were found to be a part of the ABI mechanism. The genes associated with CRISPR-Cas system were not present in the *Synechococcus* MAGs (Chapter 3 section 3.3.5.5).

Apart from the defence genes identified on the *Synechococcus* MAGs, genes coding for two BREX proteins (BrxC and PglX) were present on the SynAce01 genome. The BrxC protein had 50% protein sequence similarity to BREX system P-loop protein BrxC from a *Verrucomicrobia* bacterium, whereas the PglX protein had 52% protein sequence similarity to BREX-1 system adenine-specific DNA-methyltransferase PglX from *Leptospira ognonensis*. As previous studies have shown that phage resistance genes are usually clustered together in defence islands (Makarova et al, 2011), the genes neighbouring the BREX genes were assessed, leading to the identification of *brxA* and *brxB*. The *brxA* gene coded for a DUF1819 family protein of unknown function, whereas *brxB* gene coded for a DUF1788 domain-containing protein of unknown function; these annotations were consistent with their previous gene function assignments (Goldfarb et al, 2015). The gene coding for BrxL (a Lon-like protease) was also present in SynAce01, albeit at a different location on the genome, but the core *pglZ* gene was not identified. Notably, the *brxC* and *pglX* BREX genes in SynAce01 flanked a toxin gene and an antitoxin gene of a putative RelBE type II T-A system as well as an antitoxin gene (*abiEi*) of a type IV T-A system possibly involved in ABI mechanism (Figure 4.5a). The *pglX* gene was truncated and appeared to be disrupted by an *IS481* family transposase gene, which flanked its truncated side. The *brxL* gene was also truncated and coded for a BrxL protein that was less than one-third of the median size of most BrxL proteins (median protein length being 682 aa; Goldfarb et al, 2015). The FR analysis showed alignment of reads from different merged metagenomes to these SynAce01 BREX genes with no SNPs and no variable coverages, suggesting that probably all Ace Lake *Synechococcus* contained this incomplete type I BREX system cassette (Figure 4.5b).

To determine whether the BREX gene cassette identified in SynAce01 was present in the *Synechococcus* MAGs, the alignment of the *Synechococcus* MAGs to the SynAce01 genome was reanalysed. The annotated genes in the selected regions of *Synechococcus* MAGs were manually reannotated to verify their potential function (Table 4.6). Interestingly, some *Synechococcus* MAG contigs also contained a *pglZ* gene clustered with a non-truncated *brxL* gene, although these two genes were not near the incomplete BREX defence cassette. Genes coding for an ATP-dependent Lhr-like helicase, a serine/threonine protein kinase, and a phosphoadenosine phosphosulfate reductase were also identified among the *Synechococcus* MAGs, however, their manual annotation could not verify them as BREX genes (Table 4.6). Moreover, their neighbouring genes were involved in various metabolic functions, but not cell defence.

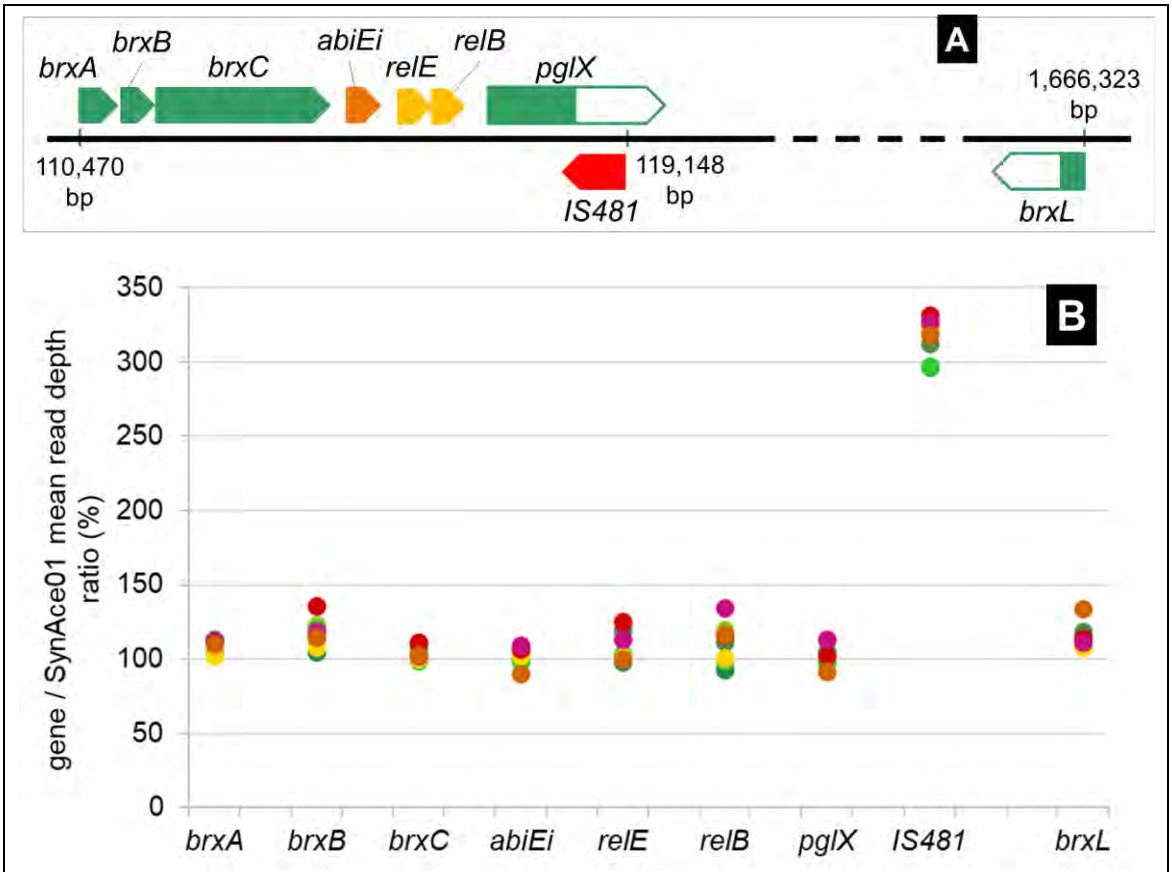


Figure 4.5 BREX defence system genes in SynAce01 and their coverage in Ace Lake merged metagenomes. (A) The schematic shows the location of BREX genes (green) on SynAce01 genome (black line). The BREX defence cassette contained *brxA*, *brxB*, *brxC*, and truncated *pglX* genes, although the cassette was interrupted by the presence of genes associated with T-A system (type IV T-A gene, orange; type II T-A genes, yellow). A transposase gene (red) flanked the defence cassette on one side. A truncated *brxL* gene was located far from the defence gene cassette. The empty regions with green borders in *pglX* and *brxL* genes indicate

truncated parts of these BREX genes. **(B)** The scatter-plot displays the mean read depths of the genes identified in the SynAce01 BREX cassette (genes along x-axis; also shown in **A**) relative to the mean read depths of SynAce01 in merged metagenomes from Ace Lake. Read depth values were calculated from the data in the base coverage files generated using BBMap v38.51 (section 4.2.2). The gene/SynAce01 mean read depth ratio was calculated by dividing the mean read depth of a gene by the overall mean read depth of SynAce01 in a merged metagenome (% ratio along y-axis). Most gene/SynAce01 ratio values were around 100% suggesting that the mean read depths of these genes were similar to the SynAce01 mean read depth in a merged metagenome. This, in turn, indicated that the SynAce01 BREX cassette was probably present in all *Synechococcus* in Ace Lake. The *IS481*/SynAce01 ratio was ≥300%, which indicated that the mean read depth of this transposase gene was three-times the SynAce01 mean read depth in merged metagenomes. Ace Lake merged metagenomes: Nov 2008_Upper 3 (■); Aug 2014_Upper 3 (■); Oct 2014_Upper 3 (■); Dec 2014_Upper 3 (■); Oct 2014_Interface (■); Dec 2014_Lower 1 (■); Dec 2014_Lower 2 (■); Nov 2013_Lower 3 (■).

Table 4.6 Defence genes annotated in *Synechococcus* MAGs. [A] The gene annotations were provided by JGI's IMG system. [B] The gene functions were verified against reference proteins in UniProtKB/Swiss-Prot database using the ExPASy BLAST+ online service (https://web.expasy.org/blast/). The proteins with poor matches to reference proteins in UniProtKB/Swiss-Prot database were aligned to the complete UniProtKB database for verification of function. The highlighted genes had functions similar to some of the DISARM as well as BREX system genes.

Defence system	Subsystem type	Annotated gene[A]	Gene function and protein sequence identity (%)[B]
R-M system	**Type I** R-M system	Type I restriction enzyme M protein	35% Probable type I restriction enzyme BthVORF4518P M protein *Bacteroides thetaiotaomicron*
		Type I restriction enzyme S subunit	30% Type-1 restriction enzyme EcoBI specificity protein *Escherichia coli*
		Type I restriction enzyme M protein	51% Putative type I restriction enzyme HindVIIP M protein *Haemophilus influenzae*
		Type I restriction enzyme S subunit	29% Putative type-1 restriction enzyme MjaXP specificity protein *Methanocaldococcus jannaschii*

		Type I restriction enzyme R subunit	43% Putative type I restriction enzyme HindVIIP R protein *Haemophilus influenzae*
		Type I restriction enzyme S subunit	23% Putative type I restriction enzyme specificity protein HI_0216 *Haemophilus influenzae*
		Type I restriction enzyme M protein	36% Type I restriction enzyme EcoEI M protein *Escherichia coli*
		Type I restriction enzyme R subunit	39% Type I restriction enzyme EcoEI R protein *Escherichia coli*
		Type I restriction enzyme R subunit	28% Type I restriction enzyme EcoKI R protein *Escherichia coli*
		Type I restriction enzyme S subunit	28% Type-1 restriction enzyme EcoBI specificity protein *Escherichia coli*
		Type I restriction enzyme M protein	32% Type I restriction enzyme EcoEI M protein *Escherichia coli*
		Type I restriction enzyme R subunit	37% Type-1 restriction enzyme R protein *Staphylococcus epidermidis*
	Putative type II R-M genes	Type I restriction-modification system DNA methylase subunit	33% Type IIS restriction enzyme Eco57I *Escherichia coli*
		Type II restriction/modification system DNA methylase subunit YeeA	25% Putative DNA methyltransferase YeeA *Bacillus subtilis*
		Type II restriction/modification system DNA methylase subunit YeeA	23% Putative DNA methyltransferase YeeA *Bacillus subtilis*
	Putative type III R-M gene	Type III restriction enzyme	26% Type III restriction-modification system EcoPI enzyme res Escherichia phage P1
CRISPR-Cas system	No Cas genes identified.	-	-

BREX system	**Type 1** BREX system	ATP-dependent Lon protease (fragment)	27% Lon protease 2 *Myxococcus xanthus*
		ATP-dependent Lon protease	86%; ATP-dependent Lon protease *Cyanobium* sp. (in UniProtKB)
		Bisphosphoglycerate-independent phosphoglycerate mutase (AlkP superfamily)	62%; PglZ domain-containing protein *Cyanobium* sp. (in UniProtKB)
		Putative inner membrane protein DUF1819	100%; Putative inner membrane protein DUF1819 *Synechococcus* sp. Ace-Pa (in UniProtKB)
		Uncharacterized protein DUF1788	100%; Uncharacterized protein DUF1788 *Synechococcus* sp. Ace-Pa (in UniProtKB)
		Hypothetical protein	50%; BREX system P-loop protein BrxC *Verrucomicrobia* bacterium (in UniProtKB)
		Type II restriction/modification system DNA methylase subunit YeeA	53%; Site-specific DNA-methyltransferase (adenine-specific) *Nitrospira lenta* (in UniProtKB)
		ATP-dependent Lhr-like helicase	26% Uncharacterized ATP-dependent helicase MJ0294 *Methanocaldococcus jannaschii*
		SNF2 family DNA or RNA helicase	26% Uncharacterized ATP-dependent helicase YqhH *Bacillus subtilis*
		SNF2 family DNA or RNA helicase	40% Uncharacterized ATP-dependent helicase YwqA *Bacillus subtilis*
		SNF2 family DNA or RNA helicase	27% Uncharacterized ATP-dependent helicase YwqA *Bacillus subtilis*
		SNF2 family DNA or RNA helicase/ERCC4-related helicase	30% Uncharacterized ATP-dependent helicase YqhH *Bacillus subtilis*

		SNF2 family DNA or RNA helicase/ERCC4-related helicase	46% Uncharacterized ATP-dependent helicase YqhH *Bacillus subtilis*
		Serine/threonine-protein kinase	33% Serine/threonine-protein kinase PknA *Nostoc* sp.
		Phosphoadenosine phosphosulfate reductase	53% Phosphoadenosine phosphosulfate reductase *Synechococcus* sp.
DISARM system	No DISARM defence cassette identified. Some genes with functions similar to DISARM system genes were found.	ATP-dependent RNA helicase RhlE	64% ATP-dependent RNA helicase RhlE *Escherichia coli*
		Primosomal protein N' (replication factor Y)	45% Primosomal protein N' *Synechocystis* sp.
		ATP-dependent DNA helicase RecQ	48% ATP-dependent DNA helicase RecQ *Escherichia coli*
		Phosphatidylserine/phosphatidylglycerophosphate/ cardiolipin synthase-like enzyme	28% Phospholipase D *Rickettsia prowazekii*
		SNF2 family DNA or RNA helicase	26% Uncharacterized ATP-dependent helicase YqhH *Bacillus subtilis*
		SNF2 family DNA or RNA helicase	40% Uncharacterized ATP-dependent helicase YwqA *Bacillus subtilis*
		SNF2 family DNA or RNA helicase	27% Uncharacterized ATP-dependent helicase YwqA *Bacillus subtilis*
		SNF2 family DNA or RNA helicase/ERCC4-related helicase	30% Uncharacterized ATP-dependent helicase YqhH *Bacillus subtilis*
		SNF2 family DNA or RNA helicase/ERCC4-related helicase	46% Uncharacterized ATP-dependent helicase YqhH *Bacillus subtilis*
		DNA (cytosine-5)-methyltransferase 1	28% Modification methylase AplI *Arthrospira platensis*

T-A system	**VapBC type II** T-A system	Antitoxin VapB	50% Virulence-associated protein B *Dichelobacter nodosus*
		Antitoxin of toxin-antitoxin stability system	52% Antitoxin VapB22 *Mycobacterium tuberculosis*
		Arc/MetJ family transcription regulator	59% Antitoxin VapB32 *Mycobacterium tuberculosis*
		PIN domain nuclease of toxin-antitoxin system	32% Ribonuclease VapC22 *Mycobacterium tuberculosis*
		PIN domain nuclease of toxin-antitoxin system	30% Ribonuclease VapC22 *Mycobacterium tuberculosis*
		PIN domain nuclease of toxin-antitoxin system	37% Ribonuclease VapC22 *Mycobacterium tuberculosis*
	MazEF type II T-A system	Antitoxin component of MazEF toxin-antitoxin module	34% Antitoxin MazE *Escherichia coli*
		mRNA interferase MazF	47% Endoribonuclease MazF9 *Mycobacterium tuberculosis*
		mRNA interferase MazF	36% Endoribonuclease MazF *Staphylococcus epidermidis*
		mRNA interferase MazF	49% Probable endoribonuclease MazF *Mycolicibacterium smegmatis*
		mRNA interferase MazF	51% Probable endoribonuclease MazF *Mycolicibacterium smegmatis*
	HigAB type II T-A system	Plasmid maintenance system antidote protein VapI	53% Virulence-associated protein I *Dichelobacter nodosus*
		Plasmid maintenance system antidote protein VapI	49% Virulence-associated protein I *Dichelobacter nodosus*
		Proteic killer suppression protein	56% Toxin HigB-1 *Vibrio cholerae*
		Proteic killer suppression protein	53% Toxin HigB-1 *Vibrio cholerae*

	Antitoxin **YefM** of the Phd antitoxin superfamily of **type II** T-A systems	Antitoxin of toxin-antitoxin stability system	35% Orphan antitoxin YefM *Salmonella typhimurium*
	Toxin modules of **HipBA**, **PemIK**, and **RelBE type II** T-A systems	Serine/threonine-protein kinase HipA	42% Serine/threonine-protein kinase toxin HipA *Escherichia coli*
		mRNA interferase MazF	50% Endoribonuclease PemK *Escherichia coli*
		mRNA interferase MazF	37% Endoribonuclease PemK *Escherichia coli*
		mRNA-degrading endonuclease RelE of RelBE toxin-antitoxin system	30% Toxin RelE *Mycobacterium tuberculosis*

4.4 Discussion

4.4.1 *Synechococcus* genomic variation — phylotypes and potential ecotypes

The *Synechococcus* MAGs and SynAce01 genome represented the same species of *Synechococcus* in Ace Lake, which was evident from the comparison of *16S rRNA* genes from *Synechococcus* MAGs and SynAce01 as well as their ANI and AAI (section 4.3.2). SNPs in the *Synechococcus 16S rRNA* marker gene indicated that at least two distinct, but closely related (99.9% marker gene similarity), subpopulations of *Synechococcus* existed in Ace Lake (section 4.3.5.2). The genomic variation identified during comparative analysis of *Synechococcus* MAGs with each other and SynAce01 as well as the FR analysis of metagenomic reads to SynAce01 also suggested the presence of different phylotypes and potential ecotypes of Ace Lake *Synechococcus*. However, the data did not indicate any season-based segregation of the *Synechococcus* subpopulations.

4.4.1.1 *Synechococcus* subpopulations representing a potential ecotype

A few genes involved in a broad range of metabolic functions were present in the LCRs (Table 4.5). These included genes coding for a YjbQ family protein, flavin reductase and nitroreductase family proteins, and a photosystem II q(b) protein; additional copies of these genes were also present in the non-variable regions of SynAce01. A single-copy gene coding for glutamine-hydrolyzing asparagine synthase (AsnB) was present in the LCRs. The SynAce01 AsnB protein sequence had 30% similarity to the asparagine synthetase [glutamine-hydrolyzing] gene from *Bacillus subtilis*. This enzyme catalyses the ATP-dependent biosynthesis of L-asparagine, where it converts aspartate to asparagine using glutamine or ammonium as a nitrogen source, preferably glutamine. An alternate reaction for biosynthesis of L-asparagine involves ammonium-hydrolyzing asparagine synthase (AsnA), which uses ammonium for conversion of aspartate to asparagine. The asparagine produced from either reaction can be used for the biosynthesis of amino-acids. Due to its stability and high nitrogen to carbon ratio, asparagine has also been considered to be suitable for nitrogen storage, although high concentration of asparagine inhibits AsnA and AsnB enzyme activities (Reitzer and Magasanik, 1982; Gaufichon et al, 2010). A study has shown that in environments with nitrogen as the limiting nutrient, the glutamine-dependent AsnB enzyme activity could be important for asparagine synthesis from glutamine (Reitzer and Magasanik, 1982). As bioavailable nitrogen is a limiting nutrient in the Ace Lake upper oxic zone (Rankin et al, 1999), the *Synechococcus* subpopulation containing the *asnB* gene (hereafter referred to as *Synechococcus* AsnB subpopulation) could represent a distinct ecotype with an improved capacity to store and utilise nitrogen.

The *Synechococcus* AsnB subpopulation had low relative coverage (≤10%) compared to the overall mean read depth of SynAce01 in the merged metagenomes, except in the bottom-most depth of Ace Lake where nearly 40% of the population had an *asnB* gene (Figure 4.6). The Ace Lake Lower zone has a high concentration of ammonium, peaking at around 15 m depth, compared to the Upper zone (Burton, 1980; Hand and Burton, 1981). A putative ammonium transporter was identified in ≥85% of the *Synechococcus* population in each merged metagenome, indicating their capacity for ammonium uptake. As AsnB can use both ammonium and glutamine as a source of ammonia for biosynthesis of asparagine, the availability of reduced nitrogen could probably sustain a larger *Synechococcus* AsnB subpopulation in the Lower zone of Ace Lake. A similar *Synechococcus* AsnB subpopulation was also found among the

Synechococcus species isolated by Callieri et al, (2019) from the dark, anoxic waters of Black Sea. An assessment of the annotated genes in the draft genomes of two *Synechococcus* strains from Black Sea showed that one strain (*Synechococcus* sp. BS55D; RefSeq assembly ID: GCF_004332415.1) had an *asnB* gene, but the other (*Synechococcus* sp. BS56D; RefSeq assembly ID: GCF_004332405.1) did not, and both had two ammonium transporter genes. The *Synechococcus* AsnB subpopulation also appeared to be stable in Ace Lake, as it was observed in metagenomes from time periods spanning seven years from 2008 to 2014 and from all three depth zones (oxic, oxycline, anoxic) of Ace Lake.

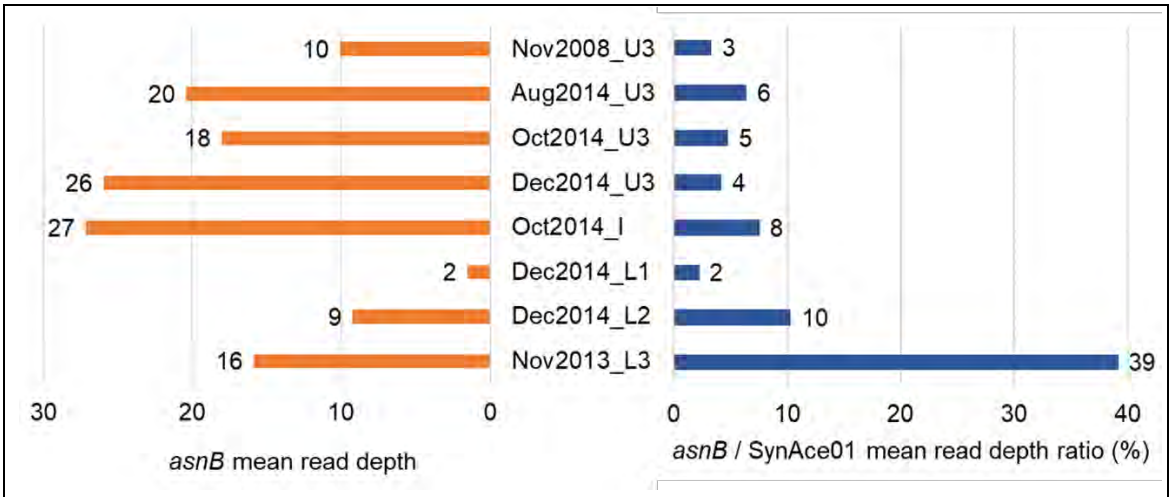


Figure 4.6 The relative coverage of SynAce01 *asnB* gene in Ace Lake merged metagenomes. The bar charts show the mean read depth of *asnB* gene (orange bars) and its coverage relative to the mean read depth of SynAce01 (blue bars) in Ace Lake merged metagenomes, indicated along the y-axis. The *asnB*/SynAce01 mean read depth ratio represented the relative coverage of *asnB* gene in the merged metagenomes and indicated the approximate percentage of the *Synechococcus* subpopulation that probably had the *asnB* gene. Read depth values were calculated from the data in the base coverage files generated using BBMap v38.51 (section 4.2.2). The *asnB*/SynAce01 mean read depth ratio was calculated by dividing the mean read depth of *asnB* by the overall mean read depth of SynAce01. Lake depths: U3, Upper 3; I, Interface; L1, Lower 1; L2, Lower 2; L3, Lower 3.

4.4.1.2 *Synechococcus* subpopulations with varying cell wall composition

The genomic variation, including LCRs and SNPs, indicated that potential *Synechococcus* phylotypes that differ in their cell wall composition might exist in Ace Lake. Multiple SynAce01 genes associated with cell wall biosynthesis, assembly, and modification (mostly multicopy genes) had low coverage in the merged Ace Lake metagenomes, suggesting that only a subpopulation of *Synechococcus* contained

additional copies of these genes (Table 4.5). The genes coded for collagen-like protein, LysM domain-containing protein, GDP-mannose 4,6-dehydratase, O-antigen ligase, ABC transporters (O-antigen and Lipid A), N-acetylmuramoyl-L-alanine amidase, and glycosyltransferase family proteins.

The collagen-like protein has been previously associated with bacterial outer membrane and is known to assume a thermostable triple helix shape (Yu et al, 2014). A single copy of its gene was identified in a SynAce01 LCR (Table 4.5). This protein has been identified in bacteria from various habitats including Antarctic sediments and glaciers (Kananavičiūtė et al, 2020). LysM domain-containing protein could also be associated with the bacterial outer membrane, as LysM domains bind peptidoglycans and are usually found on extracellular proteins or receptors (Mesnage et al, 2014). GDP-mannose 4,6-dehydratase is involved in the biosynthesis of GDP-fucose, which is used for the synthesis of extracellular polysaccharides and glycoconjugates, and has been previously identified in another *Synechococcus* (Kramm et al, 2012). O-antigen ligase family protein is involved in the production of O-antigen, an outer membrane lipopolysaccharide in bacteria. The SynAce01 genes coding for an O-antigen export system were present in LCRs in Nov 2013 Lower 3 merged metagenome, whereas the genes coding for a lipid export system was present in LCRs in all merged metagenomes. N-acetylmuramoyl-L-alanine amidase can break down peptidoglycan, and is probably involved in cell wall degradation.

The functional annotations of some of the glycosyltransferase genes in the LCRs suggested that they were involved in cell wall biosynthesis and others might play a role in cell wall modification. SNPs were also observed in a number of glycosyltransferase genes throughout the SynAce01 genome. In bacteria, glycosyltransferase gene mutations can affect its substrate specificity, which can in turn affect the type of sugar selected for glycosylation (Schmid et al, 2016). Similar variations in glycosyltransferase genes have been previously reported in an Antarctic haloarchaea population (Tschitschko et al, 2018). The genes discussed in this section can directly or indirectly impact cell wall composition, and subpopulations of *Synechococcus* that do not contain these genes or contain a lower copy number of these genes could have a different cell wall structure.

4.4.1.3 *Synechococcus* subpopulations with varying capacity for cell defence and immunity

The genomic variation observed in Ace Lake *Synechococcus* also pointed toward subpopulations of *Synechococcus* that might differ in their capacity for cell defence and immunity (Table 4.5). The genes identified in the LCRs that might be involved in some form of defence system (mostly multicopy genes) included C39 family peptidase, tellurite resistance gene (TerB family), DEAD/DEAH-box helicase, Uma2 family endonuclease, RES family toxin, MazE family antitoxin, and VapC family toxins. Of these, C39 family peptidase is usually found on ABC transporters for bacteriocin, which is a secondary metabolite with antimicrobial properties that can inhibit the growth of other nearby closely related bacteria (Dirix et al, 2004; Cotter et al, 2013). Bacteriocin gene clusters have been identified in a number of marine cyanobacteria, including various *Synechococcus* spp. (Wang et al, 2011). TerB family proteins can confer immunity against tellurite, a rare compound made of tellurium dioxide and highly toxic to most bacteria, as it generates of reactive oxygen species (Taylor, 1999; Chasteen et al, 2009). A number of tellurite-resistant bacteria have been isolated from Antarctica previously (Arenas et al, 2014).

The SynAce01 Uma2 family endonuclease was verified as a putative R-M system endonuclease through alignment to reference proteins in UniProtKB/Swiss-Prot database. Moreover, SNPs were identified in another copy of Uma2 family endonuclease gene in SynAce01. SNPs in restriction enzymes have been previously reported and it has been suggested that point mutations can help to improve their target sequence specificity (Saravanan et al, 2008). RES and VapC family toxins and MazE family antitoxin belong to type II T-A systems. Of the four copies of VapC toxin genes present in the LCRs, two genes had SNPs as well. Variations in the sequence of VapC toxin have been observed previously and it has been suggested that environmental conditions might contribute toward the evolution of T-A system modules (Lopes et al, 2019). DEAD/DEAH-box helicases are generally involved in RNA metabolism, but some have been reported to contribute toward cell innate immunity as well as viral interactions (Perčulija and Ouyang, 2019). The genes discussed in this section were probably involved in a variety of defence systems that can affect the endurance and growth of *Synechococcus*. See below section 4.4.2 for further discussion on *Synechococcus* cell defence and viral associations.

4.4.2 *Synechococcus* potential for defence against viruses

The analysis of the *Synechococcus* MAGs showed the presence of many genes associated with various defence systems that potentially provided viral immunity to the bacteria (Table 4.6). Ace Lake *Synechococcus* does not contain CRISPR-Cas system genes, which is consistent with previous findings in marine cyanobacteria (Cai et al, 2013). A number of genes coding for type I R-M system and some putative type II and type III R-M system genes were present in *Synechococcus*, some of which contained SNPs. Similar point mutations in restriction enzymes have been reported to improve their capacity to detect and eliminate foreign DNA, including viruses (Saravanan et al, 2008). Multiple copies of the genes coding for various T-A system proteins, including VapBC, MazEF, and HigAB type II T-A systems, YefM antitoxin, and HipA, PemK, and RelE toxins, were identified in *Synechococcus*. Of these, the MazEF type II T-A system is known to be involved in the ABI mechanism of viral infection disruption, which causes the death of the infected host cell to prevent the spread of viral infection to other host cells in the population (Hazan and Engelberg-Kulka, 2004; Engelberg-Kulka et al, 2005). Although a number of helicase and methylase genes were present in *Synechococcus* MAGs, none of them were found to be associated with DISARM system, suggesting that this defence system was probably not present in Ace Lake *Synechococcus*.

The FR analysis to SynAce01 genome showed that Ace Lake *Synechococcus* contained genes for a type I BREX system (Figure 4.5). This defence system is known to prevent viral DNA replication after the virus has invaded the host cell (Goldfarb et al, 2015). The analysis of SynAce01 BREX genes showed that the BREX system was probably inactivate, as the defence cassette (i) was missing the core *pglZ* gene; (ii) had a truncated *pglX* gene disrupted by the *IS481* transposase gene flanking it on one side; (iii) contained a RelBE type II T-A system between *pglX* and *brxC* genes; and (iv) had a truncated *brxL* gene located far from the defence cassette (Figure 4.5). A similar configuration of genes has been reported in the type 5 BREX system of some Antarctic haloarchaea from Deep Lake in the Vestfold Hills, containing a transposon-disrupted *pglX* gene and presence of VapBC type II T-A system genes (Tschitschko et al, 2015). In this study, the *pglX* gene from the genome of *Hrr. lacusprofundi* had a low coverage in the Deep Lake metagenomes, which along with the identification of a *Hrr. lacusprofundi* contig with an intact *pglX* gene indicated the presence of haloarchaea subpopulations capable of producing functional PglX proteins. This might not be the

case with *Synechococcus* in Ace Lake, as all BREX genes were identified in non-variable coverage genomic regions of SynAce01 (Figure 4.5b), and the *pglX* genes in the *Synechococcus* MAGs were also truncated and were usually present on one end of the MAG contigs. Such reordering and/or disruption of *pglX* gene, including presence of SNPs, has been reported previously, and it has been speculated that the *pglX* gene is either the specificity module of BREX systems or it is highly toxic (Laity et al, 1993; Sumby and Smith, 2003; Goldfarb et al, 2015). As most phage-related defence genes are either toxic or apply fitness costs to the hosts, it has been suggested that the loss or truncation of BREX genes and/or their genomic reorganization in the host can serve to elevate their toxic effects in the absence of selection pressure imposed by phage (Hallet, 2001; Cerdeño-Tárraga et al, 2005; Gomez and Buckling, 2011; Hall et al, 2011; Stern and Sorek, 2011; Bikard & Marraffini, 2012; Makarova et al, 2012; Goldfarb et al, 2015).

Notably, some of the *Synechococcus* MAGs, but not SynAce01 genome, contained *pglZ* core gene alongside a complete *brxL* gene, indicating the ability of a *Synechococcus* subpopulation to produce functional PglZ and BrxL proteins. These two genes (*pglZ* and *brxL*) are known to be co-transcribed in BREX systems as are *brxA-brxB-brcC-pglX* (Goldfarb et al, 2015). Overall, Ace Lake *Synechococcus* had all defence genes associated with a type 1 BREX system — containing *brxA*, *brxB*, *brxC* (core), *pglZ* (core), and *brxL* as well as a truncated *pglX* gene (containing only a portion of the methylase domain). However, the lack of a complete *pglX* gene in the *Synechococcus* population indicated their inability to produce functional PglX protein. As PglX is involved in methylation of host DNA to differentiate it from phage DNA, it has been recognised as being essential for BREX-mediated virus resistance (Goldfarb et al, 2015). Therefore, in the absence of functional PglX proteins, the Ace Lake *Synechococcus* BREX system would be inactive. However, it has been previously reported that the BREX defence cassettes are readily exchanged through HGT and the defence genes in them tend to co-evolve (Goldfarb et al, 2015). If intact BREX defence genes exist in the microbial population of Ace Lake, it might be possible for *Synechococcus* to reacquire the BREX defence system when under phage pressure.

Other than various cell defence and immunity systems, SNPs in the genes coding for membrane-associated proteins, such as outer membrane proteins, pilus assembly proteins, and substrate transport proteins, as well as genes associated with cell wall

modification, such as glycosyltransferases, were observed in Ace Lake *Synechococcus*. SNPs in glycosyltransferase genes can potentially lead to changes in cell surface structure (Schmid et al, 2016). Therefore, mutations in these membrane- and cell wall-associated genes could provide immunity against viruses by changing cell surface composition. This strategy is known to be employed by the marine cyanobacteria *Prochlorococcus* that evades viruses through mutations in its cell-surface genes, which prevents virus attachment by changing the cell surface structure (Avrani et al, 2011). Similar variations in the cell surface proteins including S-layer, archaella, and adhesin proteins as well as glycosyltransferases have been observed in the Antarctic haloarchaea from Deep Lake and were considered to be a method for viral evasion (Tschitschko et al, 2015; Tschitschko et al, 2018).

Interestingly, a terminase gene (flanked by hypothetical and uncharacterised genes) was identified in the LCR starting at ~20 kb length of SynAce01 genome, suggesting the presence of this phage packaging gene in a subpopulation of Ace Lake *Synechococcus* (Table 4.5). A replication-defective prophage (phiSynAce1) has been previously reported at this position in the SynAce01 genome (Tang et al, 2019). Phage have been shown to be involved in the horizontal transfer of genetic material in a marine *Synechococcus*, including transfer of genes associated with modification of cell surface composition (Palenik et al, 2003). The potential role of viruses in HGT was also observed in Antarctic haloarchaea from Deep Lake in the Vestfold Hills, where a defective prophage (Hlac-Pro1) associated with the archaeal BJ1 virus was identified in the genome of *Hrr. lacusprofundi* (DeMaere et al, 2013; Tschitschko et al, 2015). Cyanophages can drive the evolution of marine cyanobacteria, which in turn can enable co-existence of host and virus due to the presence of virus susceptible as well as resistant host populations (Coleman et al, 2006; Avrani et al, 2011; Zborowskya and Lindell, 2019). As changes in cell wall structure could help evade viruses, it can be speculated that *Synechococcus* subpopulations containing additional genes for cell wall modification and cell defence possibly represent virus resistant populations. On the other hand, *Synechococcus* subpopulations that do not contain these additional genes could probably represent virus susceptible populations. Moreover, it has been previously reported that the viral predators of marine cyanobacteria can be host-specific (specialist) or have a broad range of hosts (generalist), and that host cyanobacteria display resistance to these two types of viruses at the extracellular- or intracellular-level,

respectively (Zborowskya and Lindell, 2019). Therefore, it is possible that *Synechococcus* subpopulations use cell wall modifications as a means to resist specialist viruses, whereas cell defence modifications are used for resistance against generalist viruses (Figure 4.7). In any case, the presence of a potential prophage in Ace Lake *Synechococcus* and the lack of a linear correlation between *Synechococcus* and the Ace Lake cyanophage (Chapter 3 section 3.3.5.5) might indicate a complex pattern of interaction between these cyanobacteria and their viral predators.

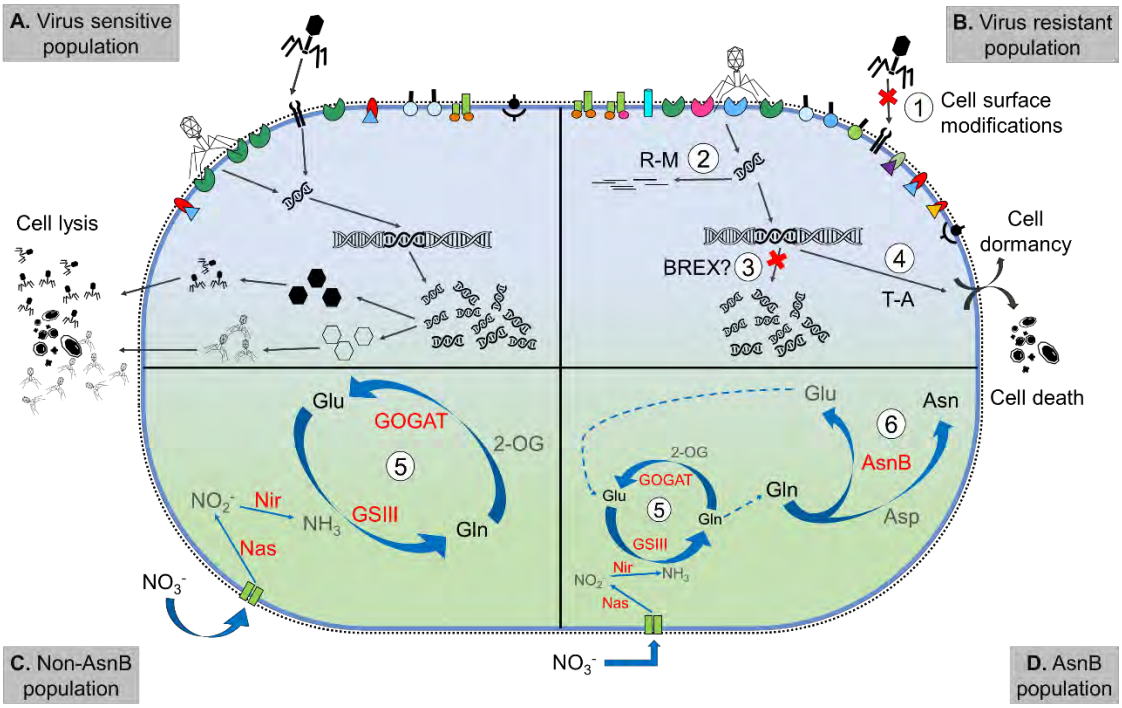


Figure 4.7 Potential *Synechococcus* populations in Ace Lake. Panels **(A, B)** highlight the types of viruses and their infection mechanisms, and *Synechococcus* capacity for different types of viral defence mechanisms. The virus-sensitive population (**A**) could include bacterial cells prone to viral attacks by host-specific viruses (black virions) and/or general viruses with broad host range (light grey virions). The viruses invade host cells, use host machinery to propagate, and then lyse the host cell to release newly formed virions. The virus-resistant population (**B**) could include bacterial cells that are capable of evading viruses or neutralising the invading viral genetic material. The *Synechococcus* phylotypes that have additional and/or modified copies of genes involved in cell surface modification and cell defence might be able to fend off phage attacks (section 4.4.2). This has been previously reported for other bacteria and archaea (Avrani et al, 2011; Tschitschko et al, 2015; Tschitschko et al, 2018; Saravanan et al, 2008). The resistant cells could: (1) evade viruses, especially host-specific viruses, through modification of cell surface proteins; (2) degrade invading viral genetic material using modified R-M defence systems with improved target specificity; (3) prevent viral DNA replication through defence mechanisms such as BREX defence system; or (4) prevent further spread of viruses in the host

population by triggering cell death or dormancy through T-A system proteins involved in ABI mechanism. **(C, D)** The Ace Lake *Synechococcus* has the capacity for assimilatory nitrate reduction and glutamine production via the GS/GOGAT pathway (5) as part of nitrogen cycling. A subpopulation of *Synechococcus* contained the *asnB* gene **(D)** coding for an asparagine synthetase [glutamine hydrolyzing] enzyme that catalyses the production of asparagine from glutamine (6). In the pathway reactions, enzymes are shown in red font, whereas the main substrates and products are shown in black font. The icons for virions, DNA, and degraded cell were taken from The Noun Project website (https://thenounproject.com/). 2-OG, 2-oxoglutarate; Asn, asparagine; AsnB, asparagine synthetase [glutamine hydrolyzing]; Asp, aspartate; Gln, glutamine; Glu, glutamate; GOGAT, glutamine-2-oxoglutarate-amido transferase (or glutamate synthase); GSIII, glutamine synthetase III.

4.5 Conclusion

Synechococcus is the second most abundant microbe in Ace Lake, and the most abundant microbe in the upper oxic zone of the lake (Chapter 3 section 3.3.3). The genetic composition of the LCRs in SynAce01 suggested the presence of different *Synechococcus* phylotypes and potential ecotypes in Ace Lake, including subpopulations with varying cell wall composition, cell defence capacity, and/or the ability to utilise glutamine as a nitrogen source for asparagine production. A number of sequence variations were also observed in Ace Lake *Synechococcus*, most of which were in genes associated with cell wall assembly and modification, membrane proteins, substrate transporters, and mobile elements. Variations in a similar set of genes have also been reported in three haloarchaea from Deep Lake (DeMaere et, 2013).

The Ace Lake *Synechococcus* contained a variety of defence genes (R-M, BREX, T-A systems) to prevent or disrupt viral infection, but did not contain any CRISPR-Cas genes (Table 4.6), consistent with previously reported data from marine cyanobacteria (Cai et al, 2013). These intracellular defence genes could provide immunity against viruses with a broad host range; as previously observed in other marine bacteria (Zborowskya and Lindell, 2019). SNPs observed in some of the genes associated with cell wall structure could lead to the modification of cell surface composition, thereby providing immunity against host-specific viruses that attach to receptors on the host cell (Avrani et al, 2011; Schmid et al, 2016; Tschitschko et al, 2015; Tschitschko et al, 2018; Zborowskya and Lindell, 2019). Overall, the findings in this chapter suggested

that a single species of *Synechococcus* is prevalent in Ace Lake, with subtle genomic variation leading to subpopulations with better capacity to evade viruses (virus resistant population) and/or to thrive in the nitrogen-limiting environment of the lake (AsnB population) (Figure 4.7). The subpopulations of *Synechococcus* from different seasons did not appear to differ, but the abundance of *Synechococcus* AsnB subpopulation increased with lake depth suggesting some depth-based variations in *Synechococcus* population.

A similar analysis of the most abundant microbe in Ace Lake, namely *Chlorobium*, along with an analysis of the endemicity of Ace Lake *Chlorobium* to the Vestfold Hills, are discussed in Chapter 5.

5. Ace Lake *Chlorobium* — genomic variation, defence against viruses, and endemism in the Vestfold Hills

5.1 Introduction

In Ace Lake, *Chlorobium* closely related to *C. phaeovibrioides* DSM 265 (hereafter referred to as C-phaeov) was found to be the most abundant microorganism, with very high abundance at the oxycline (Interface) of the lake, especially in summer and spring seasons (Chapter 3 section 3.3.3; Appendix G). This is consistent with previous reports of high abundance of this GSB in the Ace Lake oxycline (Burke and Burton, 1988a; Coolen et al, 2006; Ng et al, 2010; Lauro et al, 2011). *Chlorobium* are GSB (family *Chlorobiaceae*) belonging to the *Chlorobiales* order of *Chlorobia* class in the *Chlorobi* phylum of bacteria. C-phaeov is a facultative anaerobe and a mesophile that was isolated from a saline intertidal flat in Germany (IMG taxon ID: 640427130). The *16S rRNA* gene as well as BclA (bacteriochlorophyll A) protein-based phylogeny of the *Chlorobiaceae* family and the functional potential of Ace Lake *Chlorobium* have been analysed before (Imhoff, 2003; Coolen et al, 2006; Ng et al, 2010; Lauro et al, 2011). The Ace Lake *Chlorobium* relative abundance varied with season — very high in summer, low in winter, decreased further in early spring and revived to higher abundance in following late spring and summer (Chapter 3 section 3.3.4). *Chlorobium* is a key species in Ace Lake, based on its high abundance (Figure 3.6) and contribution to various nutrient cycles in the lake (Coolen et al, 2006; Ng et al, 2010; Lauro et al, 2011). Its ability to recover from very low abundance in early spring (<1%, Figure 3.6) to very high abundance in summer (>50%, Figure 3.6) indicated that the Ace Lake *Chlorobium* population (or a subpopulation) might have a distinctive genomic capacity to efficiently use available light for fast growth in summer. In this chapter, the genomic variation within the Ace Lake *Chlorobium* population was assessed, comparing *Chlorobium* identified in metagenomes from different seasons. Here, ecotypes and phylotypes refer to *Chlorobium* with subtle genomic differences that may or may not affect their metabolic capacity, respectively; similar to *Synechococcus* phylotypes and ecotypes (Chapter 4 section 4.1). *Chlorobium* phylotypes and ecotypes have also been referred to as *Chlorobium* subpopulations in the chapter.

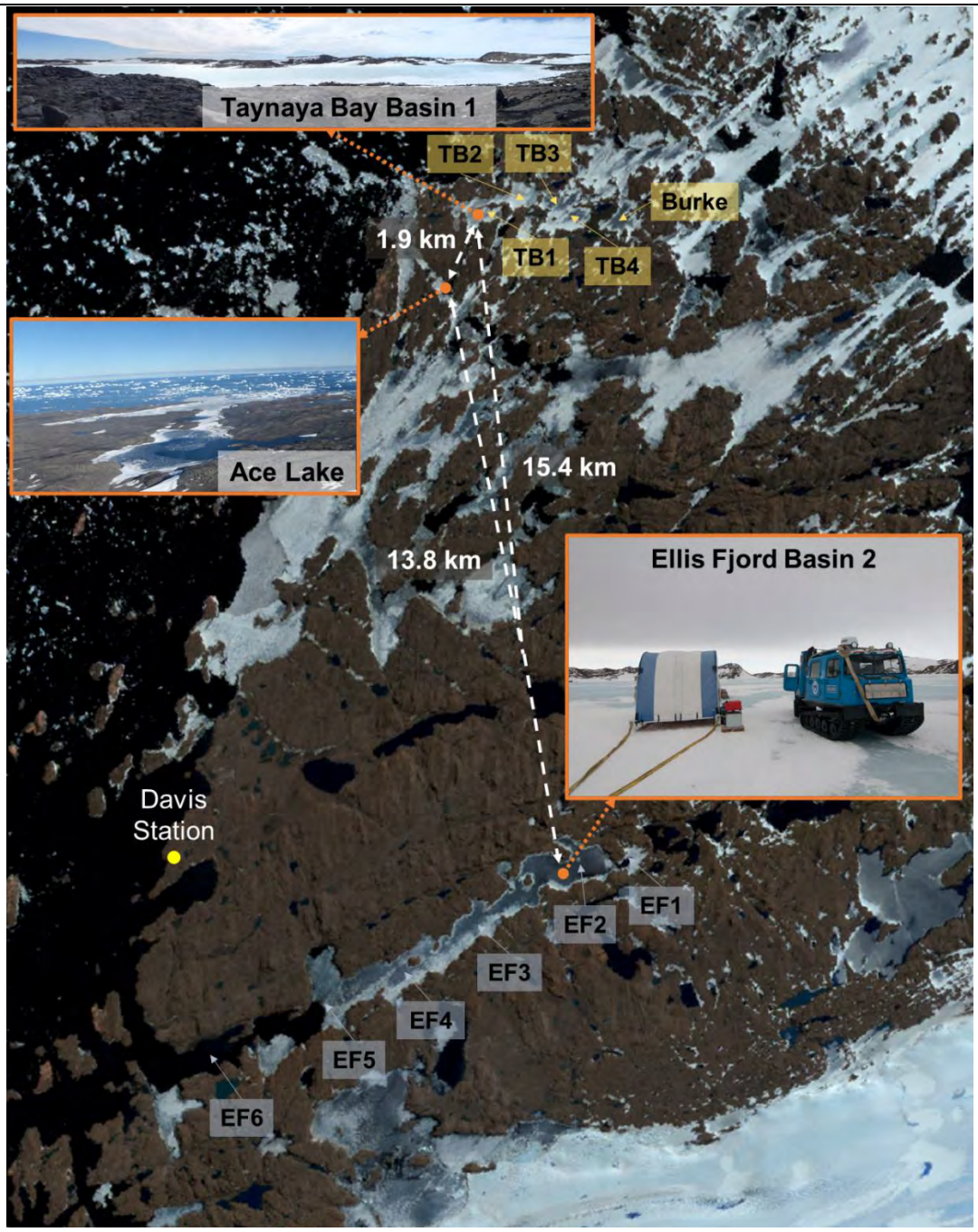


Figure 5.1 Location of Ace Lake, Ellis Fjord, and Taynaya Bay in the Vestfold Hills. The location of Davis Station is shown with a yellow dot, whereas Ace Lake, Ellis Fjord Basin 2, and Taynaya Bay Basin 1 are shown with orange dots and arrows. The distance between the three aquatic systems is shown in white. Ellis Fjord has six basins, marked as EF1–EF6 in the figure, of which two are meromictic (EF1 and EF2). The five meromictic basins of Taynaya Bay are also shown, marked as TB1–TB4 and Burke. The satellite map of the Vestfold Hills and the distance measurements were produced using the interactive atlas available on Landsat Image Mosaic of Antarctica website (https://lima.usgs.gov/antarctic_research_atlas/). The locations of Ellis Fjord and Taynaya Bay basins were taken from the data published by Gallagher and

Burton (1988) and Gibson (1999). The photos of the three aquatic systems were taken by Sarah Brazendale.

Ace Lake is a stratified lake in the Vestfold Hills (78°15′ E, 68°33′ S), which lie along the east coast of Antarctica and cover approximately 411 km^2 area, being mostly free of ice. The Vestfold Hills are well-known for their repertoire of stratified lakes and marine basins, including Ace Lake, Ellis Fjord, and Taynaya Bay. These three aquatic systems lie within a 20 km radius around the Davis Station; the Ace Lake and Taynaya Bay Basin 1 are <2 km apart, but are 14–15 km away from Ellis Fjord Basin 2 (Figure 5.1).

Ellis Fjord (68°36' S, 78°07' E) is a ~10 km long, up to 117 m deep, narrow water inlet of marine origin in the Vestfold Hills (Figure 5.1). The fjord is covered by ice for most of the year and has six basins, of which the two inner basins (Basins 1 and 2) are meromictic (Gallagher and Burton, 1988). The entrance to Ellis Fjord is restricted by a shallow sill at 4 m depth and its six marine basins are separated by sills at depths 1 to 30 m, which together allow for the stable stratification of the meromictic basins of Ellis Fjord (Burke and Burton, 1988a; Gallagher and Burton, 1988; Gallagher et al, 1989; Gibson 1999). The maximum recorded depth of its meromictic Basin 1 (also called Small meromictic basin) is 13 m and of meromictic Basin 2 (also called Deep meromictic basin) is 110 m (Gibson, 1999; Gallagher and Burton, 1988). The Basin 2 of Ellis Fjord is separated from Basin 1 on one side by a shallow sill (1 m deep) and from the outer basins of Ellis Fjord on the other side by a sill at around 30 m depth (Gallagher and Burton, 1988). The thermocline and halocline of Ellis Fjord Basin 2 lie around 50 m depth, whereas its oxic-anoxic interface varies between 30 m (Dec 1983 data) to 45 m (Oct 1994 data) depth (Burke and Burton, 1988a; Gallagher and Burton, 1988; Gibson, 1999).

Taynaya Bay (68°27' S, 78°17' E) is a marine water inlet in the Vestfold Hills, with a maximum depth of up to 80 m (Gibson, 1999). The bay is covered by ice for nearly the whole year and has six basins, of which five basins (Burke Basin and Basins 1, 2, 3, and 4) are meromictic (Gallagher and Burton, 1988; Gibson, 1999). The maximum recorded depths of these meromictic basins of Taynaya Bay vary — Burke Basin, 35 m; Basin 1, 12 m; Basin 2, 80 m; Basin 3, 55 m; and Basin 4, 20 m (Gibson, 1999). The oxic-anoxic interface of Taynaya Bay Basin 1 was around 11 m depth in 1983 but started at around 7 m depth in 1994 (Burke and Burton, 1988a; Gibson, 1999). Moreover, the Basin 1

waters did not show strong thermal or salinity gradients (Gibson, 1999; McMinn et al, 2000).

All three systems (Ace Lake, Ellis Fjord and Taynaya Bay) contain members of *Chlorobiaceae* family (Burke and Burton, 1988a; Ng et al, 2010; Lauro et al, 2011). In this chapter, the GSB identified in the three systems were compared to each other to determine whether they belonged to the same species. The GSB were also compared to their closest related non-Antarctic species and IMG metagenomic and genomic data to evaluate their endemism to the Vestfold Hills. Here, *Chlorobium* endemism has been used to indicate that the *Chlorobium* identified in Ace Lake, Ellis Fjord and Taynaya Bay were probably native to the Vestfold Hills and not found elsewhere.

Potential viruses associated with *Chlorobium* were identified in Ace Lake (Chapter 3 section 3.3.5). Of the prokaryotic defence systems discussed earlier (Table 4.3), Ace Lake *Chlorobium* has been shown to harbour a CRISPR-Cas system (Ng et al, 2010; Lauro et al, 2011). Considering its very high abundance in Ace Lake Interface, it is likely that this GSB harbours more defence systems that might protect it from viral predation. Therefore, the genomic composition of Ace Lake *Chlorobium* was further investigated to identify other bacterial defence systems (described below in section 4.2.4). Potential GSB viruses in Ellis Fjord and Taynaya Bay were also analysed.

5.1.1 Aims

The main aim of this chapter was to assess any genomic variation within the Ace Lake *Chlorobium* population from different seasons (summer *vs* winter *vs* spring), to identify its potential phylotypes or ecotypes in the lake. For this purpose, the *Chlorobium* MAGs generated from the Ace Lake metagenomes were compared to each other in a preliminary analysis (see below section 5.3.1 for description of *Chlorobium* MAGs). This was followed by a more in-depth analysis of genomic variation using FR of the metagenomic reads from different seasons and Ace Lake Interface.

The specific aims were:

- To assess genomic variation in *Chlorobium* populations from Ace Lake, Ellis Fjord and Taynaya Bay. This analysis was performed to assess how similar these microbes were and whether they represented a *Chlorobium* species potentially endemic to the Vestfold Hills. To this end, FR analysis of the metagenomic reads from Ace Lake, Ellis Fjord Basin 2, and Taynaya Bay Basin 1 was performed, to evaluate the

similarities and differences between *Chlorobium* from the three systems. To assess *Chlorobium* endemicity to the Vestfold Hills, the *Chlorobium* MAGs marker genes were compared to metagenomic and genomic data from IMG. Moreover, the *Chlorobium* MAGs generated from Ace Lake, Ellis Fjord, and Taynaya Bay metagenomes were compared to the genome of C-phaeov, a non-Antarctic species and the closest related organism to Ace Lake *Chlorobium* OTU (Appendix G).

- To identify potential viruses of Ellis Fjord and Taynaya Bay *Chlorobium*, to compare them to Ace Lake *Chlorobium* viruses and assess the similarities in the virus-host dynamics of the three systems. The types of defence genes in the *Chlorobium* from the three systems were also evaluated, to assess their capacity for defence against viruses. Moreover, the *Chlorobium* CRISPR spacers identified in Ace Lake metagenomes from different time periods were used to analyse any seasonal pattern of spacer acquisition. The *Chlorobium* CRISPR spacers identified in Ace Lake, Ellis Fjord, and Taynaya Bay metagenomes were also compared to the spacer database to examine the biogeographic distribution of potential *Chlorobium* viruses (see Chapter 3 section 3.2.6 for description of spacer database). This analysis was performed to evaluate the endemicity of the potential *Chlorobium* viruses to the Vestfold Hills.

5.2 Methods

5.2.1 *Chlorobium* OTU bin refinement and abundance calculation in Ellis Fjord and Taynaya Bay metagenomes

A total of 12 Ellis Fjord metagenomes and four Taynaya Bay metagenomes were used for these analyses (Appendix A: Table A1). The water samples from Ellis Fjord Basin 2 were collected from 5, 18, 45, and 60 m depths onto large format filters of sizes 20, 3, 0.8, and 0.1 μm. The water sampling, sequencing, assembly, and annotation were performed as described in Chapter 2 section 2.1.1.

The water samples from Taynaya Bay Basin 1 were collected from 5 and 11 m depths; the water was passed through a 20 μm size prefilter and the biomass was collected on Sterivex cartridges of 0.22 μm filter size. The Sterivex cartridges were preserved at -80 ºC during transportation from Davis Station to Australia. The DNA was extracted in accordance with the xanthogenate-SDS (XS) DNA extraction protocol (Tillett and

Neilan, 2000). DNA quality and yield were evaluated using agarose gel electrophoresis and Qubit dsDNA BR Assay Kit (Thermo Fisher Scientific), respectively. The metagenomes were sequenced at the Australian Centre for Ecogenomics using Illumina technology (150 bp paired-end reads). The metagenomic reads were QC filtered using Trimmomatic v0.38 (Bolger et al, 2014), and the filtered reads were assembled using metaSPAdes. The Ace Lake and Ellis Fjord metagenomes had been assembled from QC filtered and error-corrected reads. Therefore, for the purpose of data consistency during comparative analyses of the metagenomes from the three systems, the Taynaya Bay filtered reads were also corrected using BFC v181 (Li, 2015) and then assembled using metaSPAdes. A total of four Taynaya Bay metagenomes, two read-corrected and two direct assemblies, were generated from the two Taynaya Bay samples. All assembled metagenomes were annotated by JGI's IMG system. For comparative analyses using Ace Lake, Ellis Fjord, and Taynaya Bay metagenomes, only the read-corrected assemblies were used. For CRISPR spacer analyses, both read-corrected and direct assemblies from Taynaya Bay were used.

The taxonomic classification of contigs in the Ellis Fjord and Taynaya Bay metagenomes was performed using the methods described in Chapter 3 section 3.2.1. *Chlorobium* bin refinement and abundance calculation in these metagenomes was performed using the methods described in Chapter 3 section 3.2.2.

5.2.2 Preliminary analysis of genomic variation within Ace Lake *Chlorobium* population using MAGs

The *Chlorobium* MAGs (Appendix A: Table A2) were downloaded from the Ace Lake time-series metagenomes (Appendix A: Table A1) available on JGI's IMG/M website (https://img.jgi.doe.gov/cgi-bin/m/main.cgi). *Chlorobium* MAGs with ≥99% genome completeness were used for the preliminary analysis of genomic variation in Ace Lake *Chlorobium*. A more detailed analysis of *Chlorobium* genomic variation within Ace Lake was performed using FR (described below in section 5.2.3.1). The *Chlorobium* MAG contigs were aligned to the contigs of the Ace Lake *Chlorobium* MAG generated from Dec 2014_Lower 2_0.1 µm-filter metagenome (hereafter referred to as AL_ref MAG). The MAG was selected because it had the highest total base pair count among all *Chlorobium* MAGs from Ace Lake and had ≥99% genome completeness. The methodology for this analysis as well for ANI calculation is described in Chapter 4 section 4.2.1.

5.2.3 FR analyses

5.2.3.1 Determining genomic variation within Ace Lake *Chlorobium* population

FR was performed to analyse genomic variation in Ace Lake *Chlorobium* and to assess any differences in *Chlorobium* populations from different seasons. As *Chlorobium* was present in 3–20, 0.8–3, and 0.1–0.8 µm-filter metagenomes, the reads from all three filter fraction metagenomes from Ace Lake Interface were combined for each time period to prepare pooled metagenomes (Table 5.1). These merged metagenomes from Ace Lake Interface from each time period covered biomass sizes ranging from 0.1–20 µm. For comparative analysis, these merged metagenomes represented data from summer (Dec 2014), winter (Jul 2014, Aug 2014), and spring (Nov 2008, Nov 2013, Oct 2014). The reads from the merged metagenomes were aligned to AL_ref MAG for in-depth analysis of genomic variation in Ace Lake *Chlorobium*. The alignment of the reads to AL_ref MAG was performed as described in Chapter 4 section 4.2.2. The base coverages of AL_ref MAG in Ace Lake merged metagenomes were generated from BAM files using the 'depth' function of Samtools v1.10 and were plotted on a circos plot using R v4.0.2 (Figure 5.2d).

5.2.3.2 Determining genomic variation in *Chlorobium* populations from Ace Lake, Ellis Fjord and Taynaya Bay

FR was performed to analyse genomic variation in *Chlorobium* from Ace Lake, Ellis Fjord, and Taynaya Bay, and to eventually assess whether the *Chlorobium* populations from the three systems were endemic to the Vestfold Hills. For this analysis, the Ellis Fjord and Taynaya Bay metagenomes were selected based on the overall relative abundance of *Chlorobium* OTU in them. Similar to the Ace Lake Interface merged metagenomes, the reads from 3–20, 0.8–3, and 0.1–0.8 µm-filter metagenomes from Ellis Fjord 45 m depth were pooled to form a merged metagenome (Table 5.1). The reads from the merged Ace Lake and Ellis Fjord metagenomes as well as the Taynaya Bay metagenome from 11 m depth were aligned to the *Chlorobium* MAG generated from 3 µm-filter metagenome from 45 m depth in Ellis Fjord (hereafter referred to as EF_ref MAG). The MAG was selected because it had the highest total base pair count among the *Chlorobium* MAGs from Ace Lake, Ellis Fjord, and Taynaya Bay metagenomes and had ≥99% genome completeness. Each merged metagenome from Ace Lake and Ellis Fjord covered biomass sizes ranging from 0.1–20 µm, whereas the

Taynaya Bay 11 m metagenome covered a smaller range of biomass sizes from 0.22–20 µm. Moreover, the amount of biomass captured on the large format filters used for water sampling in Ace Lake and Ellis Fjord was more than the amount of biomass captured on Sterivex cartridges used for water sampling in Taynaya Bay. The alignment of the reads to EF_ref MAG was performed as described in Chapter 4 section 4.2.2. The base coverages of EF_ref MAG in Ace Lake and Ellis Fjord merged metagenomes and in Taynaya Bay 11 m metagenome were generated from BAM files using the 'depth' function of Samtools v1.10 and were plotted on a circos plot using R v4.0.2 (Figure 5.6).

Table 5.1 Ace Lake, Ellis Fjord, and Taynaya Bay metagenomes used for FR analyses of *Chlorobium* MAGs. [A] The 3–20, 0.8–3, and 0.1–0.8 µm-filter metagenomes from Ace Lake Interface from each time period and Ellis Fjord 45 m depth were combined to prepare the merged metagenomes shown in column three. [B] The relative abundance of *Chlorobium* OTU in the selected metagenomes was calculated using the method described in Chapter 3 section 3.2.1. [C] The number of reads represents the total number of reads in the merged metagenomes from Ace Lake and Ellis Fjord and in Taynaya Bay (TB_11m) metagenome. All Ace Lake merged metagenomes were used for FR analysis described in section 5.2.3.1, whereas all Ace Lake and Ellis Fjord merged metagenomes plus TB_11m were used for FR analysis described in section 5.2.3.2.

System	Sample collection time period and depth	Merged metagenome name[A]	*Chlorobium* OTU relative abundance (%)[B]			Number of reads[C]
			3–20 µm-filter	0.8–3 µm-filter	0.1–0.8 µm-filter	
Ace Lake	Nov 2008 12.8 m	AL Nov2008_I	42	62	81	204,878,852
	Nov 2013 13.5 m	AL Nov2013_I	12	21	33	86,383,986
	Jul 2014 13.5 m	AL Jul2014_I	2	5	6	78,035,526
	Aug 2014 14.5 m	AL Aug2014_I	1	5	5	82,792,076
	Oct 2014 13 m	AL Oct2014_I	0	1	1	70,579,806
	Dec 2014 13.4 m	AL Dec2014_I	39	57	59	140,544,592
Ellis Fjord	Oct 2014 45 m	EF_45m	14	49	48	322,272,730
Taynaya Bay	Nov 2014 11 m	TB_11m	6 (0.22–20 µm-filter)			91,287,184

5.2.3.3 Subpopulation estimations

The percentage of *Chlorobium* population containing a genomic region of interest (such as a variable coverage region or a specific gene, gene cluster, or gene operon) was calculated as the relative coverage of the region of interest — from the mean read depth of the region of interest and the overall mean read depth of the reference genome (AL_ref MAG or EF_ref MAG) in each metagenome. The mean read depths of regions of interest and reference genomes were calculated by:

$$Mean\ read\ depth_{(Reg/Gen)} = \frac{\sum_{(Reg/Gen)} Base\ read\ depth}{Total\ bases_{(Reg/Gen)}}$$

where Reg is region of interest and Gen is reference genome. The numerator indicates the sum of the read depths of the bases in a region of interest or reference genome, calculated in each metagenome. The denominator indicates the total number of bases in the region of interest or reference genome. The base read depths were taken from the base coverage files generated for each metagenome (sections 5.2.3.1 and 5.2.3.2).

The approximate percentage of *Chlorobium* population (also referred to as the abundance of a *Chlorobium* subpopulation) in a metagenome that contained the region of interest was calculated by:

$$Subpopulation_{(Reg)} = \frac{Mean\ read\ depth_{(Reg)}}{Mean\ read\ depth_{(Gen)}} \times 100$$

where Reg is region of interest and Gen is reference genome. The numerator indicates the mean read depth of the region of interest in a metagenome and the denominator refers to the mean read depth of the reference genome in a metagenome.

5.2.4 Analysis of *Chlorobium* endemicity to the Vestfold Hills

5.2.4.1 Comparative analysis of *Chlorobium* MAGs and C-phaeov genome

The contigs of Ace Lake, Ellis Fjord, and Taynaya Bay *Chlorobium* MAGs with ≥99% genome completeness were aligned to the C-phaeov genome using the methods described in Chapter 4 section 4.2.1. The methods used for ANI and AAI analyses are also described in Chapter 4 section 4.2.1. The overall functional potential of C-phaeov and *Chlorobium* MAGs from Ace Lake (AL_ref MAG), Ellis Fjord (EF_ref MAG), and Taynaya Bay (*Chlorobium* MAG from TB_11m metagenome) were compared using COG number data generated by JGI's IMG system. The COG numbers were categorised under COG categories using COG reference data taken from JGI's IMG system (https://img.jgi.doe.gov/; accessed on 21 December 2020). The *Chlorobium*

genes with COG number assignments that belonged to more than one COG category were assigned to multiple COG categories.

5.2.4.2 Comparison of *Chlorobium* markers with marker sequences in IMG databases

To assess *Chlorobium* endemism to the Vestfold Hills, the *16S rRNA* gene and BclA protein markers from *Chlorobium* MAGs were used. The *Chlorobium 16S rRNA* gene was aligned to the *16S rRNA* genes from public-assembled metagenomes (accessed on 14 Mar, 2021) and public isolates (accessed on 30 Mar, 2021) on JGI's IMG system, using IMG RNA BLAST (blastn) with e-value 10^{-5}. The *Chlorobium* protein sequence was aligned to isolate protein database (including proteins from isolate genomes, MAGs, and single-amplified genomes; accessed on 14 Mar, 2021) on JGI's IMG system, using IMG RNA BLAST (blastp) with e-value 10^{-5}.

5.2.5 Analysis of potential *Chlorobium* viruses in Ellis Fjord and Taynaya Bay metagenomes

For the analysis of potential viruses associated with Ellis Fjord and Taynaya Bay *Chlorobium*, the spacer and repeat data in the metagenome CRISPR files were used, along with the data in the Antarctic virus catalogue and spacer database (see Chapter 3 section 3.2.6 for description of these files and databases). The Antarctic virus catalogue and spacer database did not include data from the Taynaya Bay metagenomes, since these metagenomes were not available at the time the databases were created. Therefore, the Taynaya Bay viral contigs were identified from matches to the Antarctic virus catalogue and not from the rigorous processing of Taynaya Bay metagenomes through the virus identification pipeline (Páez-Espino et al, 2016). To identify viral contigs in Taynaya Bay metagenomes, all assembled contigs were aligned to the Antarctic virus catalogue using blastn module of BLAST+ v2.9.0, with e-value $\leq 10^{-3}$ and $\geq$97% alignment identity. From the output, only the metagenome contigs with 100% identity across the whole length of either the query contig or the reference viral contig were considered as Taynaya Bay viral contigs.

To identify *Chlorobium*-associated CRISPR spacers in Ellis Fjord and Taynaya Bay metagenomes, the contig IDs in the *Chlorobium* OTU refined bin were compared to the contig IDs in the metagenome CRISPR files. These spacer sequences were then aligned to all contigs in the Antarctic virus catalogue using the 'megablast' option of BLAST+

v2.9.0, with e-value ≤10^{-3} and ≥90% alignment identity, to identify viral contigs potentially associated with Ellis Fjord and Taynaya Bay *Chlorobium*. Only viral contigs with ≥97% identity to *Chlorobium*-associated spacers were considered for further analysis. The data in the Antarctic virus catalogue were used to assign cluster or singleton designations to the potential *Chlorobium* viral contigs. The virus-host relation was assessed using the method described in Chapter 3 section 3.2.6.1. The spacer hits to the viral contigs associated with Ace Lake *Chlorobium* (Appendix H: Table H1) were reassessed to identify any host contigs belonging to Ellis Fjord *Chlorobium*.

5.2.6 Analysis of *Chlorobium* defence system genes and CRISPR spacers

The defence systems genes in *Chlorobium* MAGs were assessed using the method described in Chapter 4 section 4.2.4. The CRISPR spacer arrays identified in the Ace Lake *Chlorobium* MAGs were investigated to identify any seasonal pattern of spacer acquisition. The CRISPR spacer and repeat sequences obtained from Ace Lake, Ellis Fjord, and Taynaya Bay *Chlorobium* were compared to assess any sequence similarities.

5.2.7 Phylogeny assessment

The phylogenetic analysis of *Chlorobium* was performed using the *16S rRNA* genes and BclA proteins from Ace Lake, Ellis Fjord, and Taynaya Bay *Chlorobium* as well as various members of *Chlorobiaceae* family (Table 5.2). The *16S rRNA* gene sequences were aligned using ClustalW algorithm and BclA protein sequences were aligned using Neighbor Joining cluster method of MUSCLE algorithm in MEGA X v10.1.7 software. The alignments were used for generating maximum likelihood trees in MEGA X v10.1.7 with default parameters and 1,000 bootstrap values.

Table 5.2 *Chlorobiaceae* family members used for phylogenetic analysis of Ace Lake, Ellis Fjord, and Taynaya Bay *Chlorobium*. * The accession IDs of the *16S rRNA* genes and BclA proteins or the species genomes are provided. † The table includes *16S rRNA* genes and BclA proteins from Ace Lake (AL), Ellis Fjord (EF), and Taynaya Bay (TB) *Chlorobium* MAGs.

Organism	*16S rRNA* gene		BclA protein	
	Accession ID*	Length (in bp)	Accession ID*	Length (in aa)
Chlorobaculum limnaeum	NZ_CP017305.1	1505	WP_069808958.1	366
Chlorobaculum macestae	NR_116056.1	1395	-	-
Chlorobaculum parvum	NC_011027.1	1507	WP_012502817.1	365

Chlorobaculum tepidum	NR_044685.2	1450	WP_010933165.1	366
Chlorobaculum thiosulfatiphilum	NR_029321.1	1388	WP_139457377.1	366
Chlorobium chlorochromatii	NC_007514.1	1506	WP_011362353.1	366
Chlorobium chlorovibrioides	Y10649.1	1466	-	-
Chlorobium ferrooxidans	Y18253.1	1804	WP_006366194.1	366
Chlorobium gokarna	AJ888464.1	1287	-	-
Chlorobium limicola	NC_010803.1	1504	WP_012466619.1	366
Chlorobium luteolum	NC_007512.1	1504	WP_011358231.1	366
Chlorobium phaeobacteroides	NC_010831.1	1507	WP_012474280.1	367
Chlorobium phaeovibrioides	NC_009337.1	1506	WP_011890560.1	366
Chloroherpeton thalassium	NC_011026.1	1501	WP_012499263.1	370
Pelodictyon phaeoclathratiforme	NC_011060.1	1502	WP_012507834.1	366
Prosthecochloris aestuarii	NC_011059.1	1506	WP_012506146.1	367
Prosthecochloris indica	NR_132595.1	1393	-	-
Prosthecochloris marina	-	-	WP_110023260.1	367
Prosthecochloris vibrioformis	M62791.1	1507	WP_068866593.1	367
AL *Chlorobium*†	IMG taxon ID: 3300023061 Gene ID: Ga0222700_1000006154	1505	IMG taxon ID: 3300023061 Gene ID: Ga0222700_1000003178	366
EF *Chlorobium*†	IMG taxon ID: 3300031631 Gene ID: Ga0307987_100000446	1505	IMG taxon ID: 3300031631 Gene ID: Ga0307987_1000002178	366
TB *Chlorobium*†	IMG taxon ID: 3300039187 Gene ID: Ga0400661_000002_151875_153376	1502	IMG taxon ID: 3300039187 Gene ID: Ga0400661_000007_23035_24135	366

5.3 Results

5.3.1 Overview of Ace Lake, Ellis Fjord, and Taynaya Bay *Chlorobium* MAGs and C-phaeov genome

The IMG taxonomic classification of Ace Lake, Ellis Fjord, and Taynaya Bay *Chlorobium* MAGs indicated that C-phaeov was their closest related species. A total of 50 Ace Lake *Chlorobium* MAGs (~82 Mb), seven Ellis Fjord *Chlorobium* MAGs (~11 Mb), and four Taynaya Bay *Chlorobium* MAGs (~7 Mb) were used for various analyses (Appendix A: Table A2). Of these, 31 Ace Lake, five Ellis Fjord, and all Taynaya Bay MAGs had ≥99% genome completeness. The *16S rRNA* marker gene was present in 43 Ace Lake, six Ellis Fjord, and all Taynaya Bay *Chlorobium* MAGs, whereas the BclA marker gene was present in 47 Ace Lake and all Ellis Fjord and Taynaya Bay *Chlorobium* MAGs. A total of 72,884 genes on 1,231 *Chlorobium* MAG contigs were used for analyses. These genes and contigs belonged to Ace Lake, Ellis Fjord, and Taynaya Bay *Chlorobium* MAGs with ≥99% genome completeness as well as an Ace Lake *Chlorobium* MAG with 98% genome completeness and a variant BclA protein (described below in section 5.3.4.1). The *Chlorobium* MAGs used for various analyses represented high- and medium-quality draft genomes with ≥98% genome completeness and <2% bin contamination.

C-phaeov was previously named as *Prosthecochloris vibrioformis* DSM 265 or *C. vibrioforme* f. *thiosulfatophilum* DSM 265, but was reclassified as *C. phaeovibrioides* DSM 265 based on *16S rRNA* gene and BclA protein phylogenies (Imhoff, 2003). The complete genome of C-phaeov was sequenced and assembled by JGI IMG as part of a project led by Bryant DA (Bryant and Frigaard, 2006). This genome is available in NCBI (RefSeq ID: NC_009337.1). C-phaeov genome was 1,966,858 bp long with 53% GC content and contained 1,835 annotated genes, of which 1,764 were protein coding genes.

5.3.2 Analysis of genomic variation in Ace Lake *Chlorobium*

The genomic variation in Ace Lake *Chlorobium* from different seasons was assessed through (i) comparative analysis of Ace Lake *Chlorobium* MAGs (described below in section 5.3.2.1) and (ii) FR of Ace Lake metagenomic reads to AL_ref MAG (>99% genome completeness *Chlorobium* MAG from Ace Lake Dec 2014_Lower 2_0.1 µm-filter metagenome with highest total base pair count; section 5.2.2), which allowed for analysis of variations such as LCRs and SNPs (described below in section 5.3.2.2).

5.3.2.1 Analysis of sequence variations between Ace Lake *Chlorobium* MAGs

The alignment of all Ace Lake *Chlorobium* MAGs against AL_ref MAG showed five genomic regions with SNPs, three of which contained defence genes coding for type I restriction enzyme M or R subunit. The fourth (~94 kb in length) and fifth (~9 kb in length) regions contained genes probably involved in cell wall biosynthesis and modification (glycosyltransferase, GDP mannose 4,6-dehydratase, GDP-L-fucose synthase, undecaprenyl diphosphate synthase, phosphatidylinositol alpha-1,6-mannosyltransferase) and substrate transport (polysaccharide transport family flippase, O-antigen/teichoic acid export membrane protein, outer membrane protein insertion porin family, ABC-type multidrug transport system fused ATPase/permease subunit). These two regions also contained various genes involved in metabolic functions as well as genes of unknown function. As these MAGs probably represented snapshots of subpopulations of *Chlorobium* in a metagenome, the SNPs observed during their comparative analysis might not represent fixed mutations in the *Chlorobium* populations from different metagenomes. Therefore, the output of FR of Ace Lake metagenomic reads to AL_ref MAG was used to assess whether these SNPs were fixed mutations (described below in section 5.3.2.2).

5.3.2.2 FR analysis of *Chlorobium* AL_ref MAG in Ace Lake metagenomes

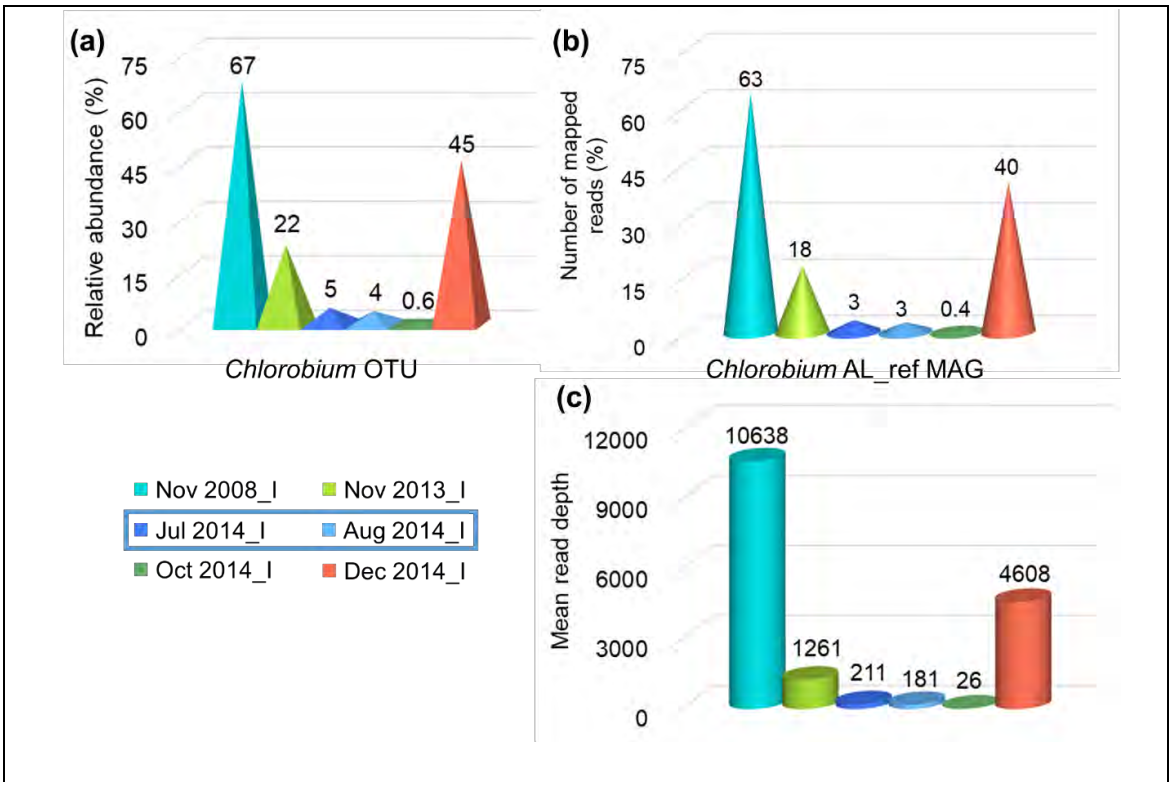
The coverage pattern of AL_ref MAG in the Ace Lake merged metagenomes was similar to the relative abundance pattern of the *Chlorobium* OTU in the merged metagenomes, showing high coverage in summer and spring, except Oct 2014 spring, and low coverage in winter (Figures 5.2a, b, c). For the analysis of SNPs in *Chlorobium* from different time periods, only the mutations that were present in at least 90% of the metagenomic reads aligned to the reference base (i.e., 0.9 variable frequency threshold) were considered as fixed mutations. Notably, only one mutation was identified in an AAA family ATPase gene involved in various metabolic functions, with an A → G transition in the Ace Lake metagenomic reads matching position 5,152 on C26 contig of AL_ref MAG (Table 5.3). The mutant AAA family ATPase gene from the Ace Lake metagenomes coded for a 519 aa length protein containing a non-synonymous mutation at position 500 (threonine → alanine). However, a comparison of this mutated AAA family ATPase protein with the proteins in the RefSeq database showed that all reference proteins contained alanine at this position, and none contained threonine (which was observed in AAA family ATPase of AL_ref MAG). A closer inspection of

the Ace Lake merged metagenome reads that aligned to position 5,152 on C26 contig of AL_ref MAG (within AAA family ATPase gene) showed that more than 99% of the reads contained guanine at this position, unlike AL_ref MAG that contained adenine. Together, these results indicated that the *Chlorobium* from different time periods did not carry a mutation in their AAA family ATPase gene, and rather the AL_ref MAG had a low variable frequency SNP in its gene. This was also evident from the FR analysis of AL_ref MAG in the metagenome from which it was generated (Ace Lake Dec 2014_Lower 2_0.1 μm-filter), which showed that all reads matching position 5,152 on C26 contig of AL_ref MAG contained guanine, except one read pair that contained adenine. These observations also highlighted the idea that these MAGs represented snapshots of *Chlorobium* subpopulations in a metagenome and needed to be carefully analysed for meaningful results.

The alignment of metagenomic reads to AL_ref MAG showed presence of a few LCRs, with read depths less than the mean read depth of AL_ref MAG in the merged metagenomes (Figure 5.2d). The annotated genes on these LCRs were mainly involved in cell wall modification (glycosyltransferases), cell defence (DEAD/DEAH box helicase, R-M proteins, BrnA antitoxin), substrate transport (iron, cobalt, vitamin B12 transporters), DNA repair, protein modification (chaperones), and various metabolic functions (Table 5.4). A few genes of unknown function and genes associated with mobile elements (transposases) were also present in the LCRs. Among the metabolic genes in the LCRs, a cluster of eight single copy genes involved in the anaerobic pathway for cobalamin biosynthesis (*cbiD*, *cbiJ*, *cbiL*, *cbiK*, *cysG*, and bifunctional *cbiFG*, *cbiET*, *cbiHC*) were identified. A single copy gene involved in cobinamide salvaging (*cbiZ*) was also identified in the LCRs. Another cluster of nine genes in the LCRs represented the N-ATPase operon (*atpD, atpC, atpQ, atpR, atpB, atpE, atpF, atpA, atpG*), which codes for ATPase subunits involved in ATP-dependent outflux of Na^+ or H^+ ions. A gene cluster containing one cobaltochelatase (*cobN*) gene and additional copies of three magnesium chelatase (*bchD*, *bchH*, *bchI*) genes was also present in the LCRs.

Overall, the genomic sequences of Ace Lake *Chlorobium* from different time periods contained no mutations. However, subpopulations of Ace Lake *Chlorobium* that might represent phylotypes and/or ecotypes were identified in metagenomes from all time

periods, and the abundances of these *Chlorobium* subpopulations varied with season (Table 5.4).



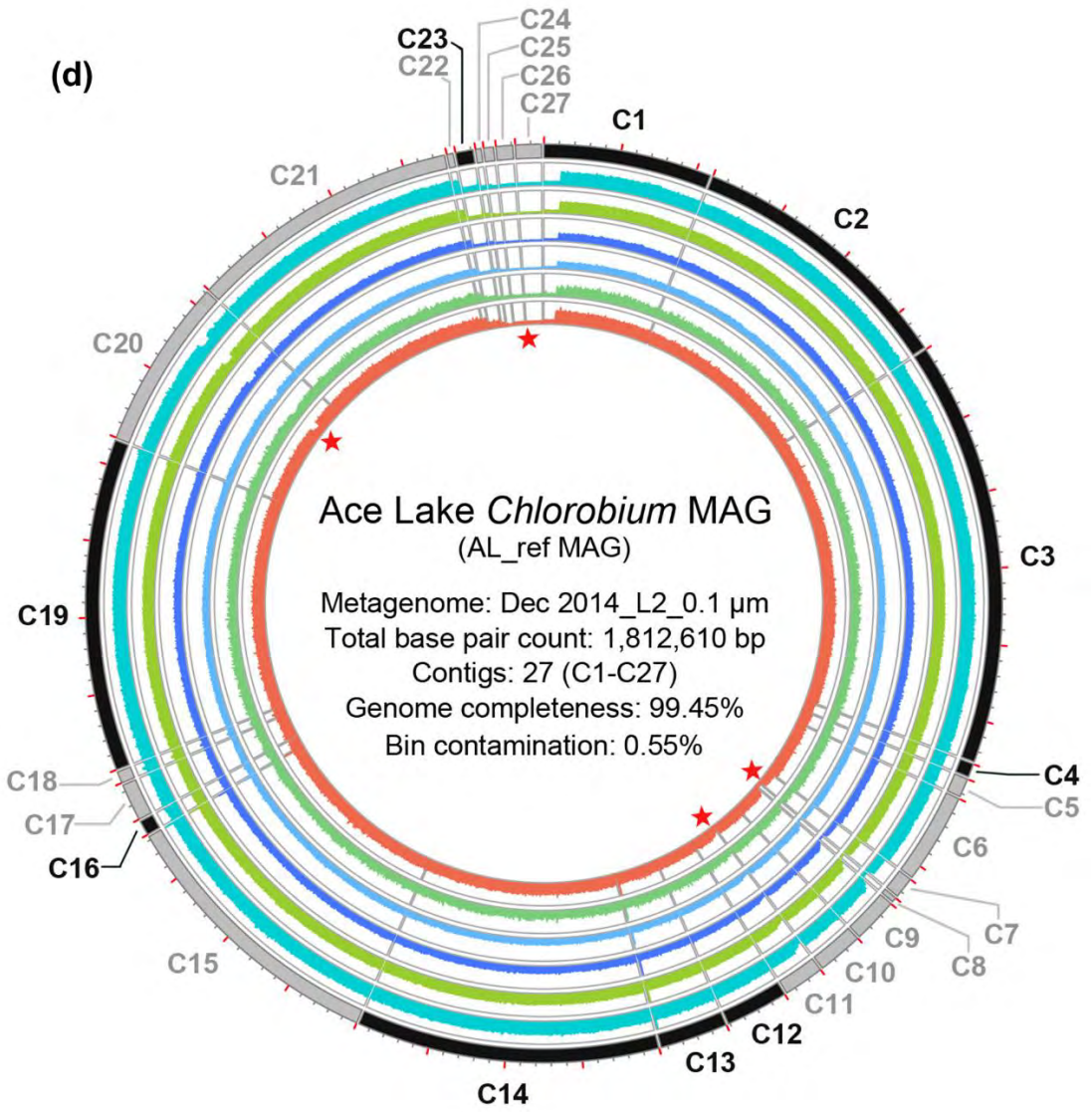


Figure 5.2 *Chlorobium* abundance, coverage distribution, and genomic variation in Ace Lake Interface metagenomes from different seasons. (a) The bar-chart shows the relative abundance distribution of *Chlorobium* OTU (coloured pyramids) in merged metagenomes from Ace Lake Interface (I) and different seasons (summer, Dec; winter, Jul, Aug; spring, Oct, Nov). *Chlorobium* OTU relative abundances in merged metagenomes were calculated from the absolute abundances of *Chlorobium* contigs in 3–20, 0.8–3, and 0.1–0.8 µm-filter metagenomes relative to the total abundance of all contigs in the three metagenomes (formula described in Chapter 3 section 3.2.1). In the colour legend, the merged metagenomes from Jul and Aug 2014 are shown in a blue box to highlight their winter origin. **(b and c)** The bar-charts show the total number of reads mapped to (**b**, coloured cones) and mean read depth of (**c**, coloured cylinders) AL_ref MAG in the Ace Lake Interface merged metagenomes from different time periods. The y-axis in **(b)** indicates the total number of reads that aligned to the AL_ref MAG, whereas the y-axis in **(c)** denotes the mean of read depths of all nucleotide bases in AL_ref MAG. Read depth values were calculated from the data in the base coverage files generated using Samtools v1.10.

(d) The circos plot depicts the coverage distribution of AL_ref MAG in Ace Lake merged metagenomes (coloured rings). The outermost ring depicts the backbone of AL_ref MAG showing Contig 1–27 (C1–C27; Table 5.3). The MAG contigs were reordered in Mauve v2.4.0 using C-phaeov as the reference genome, but only C1–C21 had any matches to C-phaeov; C22–C27 were added to one end of the ordered contigs. The grey contigs represent forward strand MAG sequences, whereas black contigs represent complementary strand MAG sequences. The contig lengths varied between 3 to 277 kb length (Table 5.3), and the x-axis scales for all contigs and rings are drawn around the circumference of the outermost ring, with major ticks (red) at the beginning and every 50 kb length and minor ticks (dark grey) at every 10 kb length of each contig. Read depth values are plotted on linear scale y-axes. As the main purpose of the figure was to highlight the distribution of variable coverage regions, the y-axes scales of circos rings vary. Read depth values greater than the y-axis limits were truncated to the y-axis maximum values. Low coverage regions are marked by red stars inside the innermost ring of the circos plot. Merged metagenomes and their y-axis scale ranges, outer to inner ring: Nov 2008_I (■, 0–20000); Nov 2013_I (■, 0–3000); Jul 2014_I (■, 0–800); Aug 2014_I (■, 0–800); Oct 2014_I (■, 0–100); Dec 2014_I (■, 0–10000).

Table 5.3 *Chlorobium* AL_ref MAG contigs. [A] The contig numbers (C1–C27) correspond to the contigs shown in Figure 5.2d. [B] The contig IDs refer to the scaffold IDs provided by JGI's IMG system. [C] The read depths of the contigs in the Dec 2014_Lower 2_0.1 µm Ace Lake metagenome, from which AL_ref MAG was generated, were measured by JGI's IMG system.

Contig number[A]	Contig ID[B]	Length (bp)	GC content	Read depth[C]
C1	Ga0222700_1000010	109,790	0.54	45
C2	Ga0222700_1000006	177,725	0.53	49
C3	Ga0222700_1000002	276,836	0.53	48
C4	Ga0222700_1000399	8,847	0.5	44
C5	Ga0222700_1000205	12,282	0.52	48
C6	Ga0222700_1000014	58,907	0.51	47
C7	Ga0222700_1000158	13,766	0.52	33
C8	Ga0222700_1003121	3,206	0.51	12
C9	Ga0222700_1000041	29,424	0.51	42
C10	Ga0222700_1000040	29,669	0.5	62
C11	Ga0222700_1000052	26,363	0.5	45
C12	Ga0222700_1000023	41,078	0.5	47
C13	Ga0222700_1000017	44,350	0.5	47

C14	Ga0222700_1000004	197,466	0.52	47
C15	Ga0222700_1000005	182,208	0.51	47
C16	Ga0222700_1000327	9,737	0.52	46
C17	Ga0222700_1000059	24,299	0.52	46
C18	Ga0222700_1000552	7,608	0.49	49
C19	Ga0222700_1000003	216,688	0.54	46
C20	Ga0222700_1000009	111,159	0.53	46
C21	Ga0222700_1000007	177,475	0.54	48
C22	Ga0222700_1001909	4,138	0.49	45
C23	Ga0222700_1000237	11,418	0.52	23
C24	Ga0222700_1002289	3,767	0.49	10
C25	Ga0222700_1000764	6,459	0.47	11
C26	Ga0222700_1000260	10,955	0.52	13
C27	Ga0222700_1000107	16,990	0.52	13

Table 5.4 Genes annotated on LCRs of AL_ref MAG. [A] The AL_ref MAG contigs mentioned in the first column are described in Table 5.3. [B] The approximate starting positions and lengths of the LCRs on the AL_ref MAG contigs are provided in the second column (the low coverage regions are labelled as red stars in Figure 5.2d). [C] The seasons mentioned in the third column refer to seasons in which the Ace Lake Interface samples were collected — summer (**S**), Dec 2014; winter (**W**), Jul 2014 and Aug 2014; spring (**Sp**), Nov 2008, Nov 2013, and Oct 2014 (Table 5.1). The percentages shown are average of relative coverage values from metagenomes from a season calculated across the region specified in column two (section 5.2.3.3). [D] The genes shown in the table were annotated by JGI's IMG system. The regions are arranged from top to bottom in the order of their occurrence along the lengths of AL_ref MAG contigs.

AL_ref MAG contig[A]	Starting position and length of LCR[B]	Seasons and % *Chlorobium* population in which observed[C]	AL_ref MAG genes annotated in the LCR[D]
C1	1 bp (11 kb length)	**S:** 32% **W:** 26–28% **Sp:** 27–31%	Hypothetical protein Restriction system protein PH (Pleckstrin Homology) domain-containing protein

			PD-(D/E)XK nuclease superfamily protein ATP-dependent exoDNAse (exonuclease V) beta subunit/superfamily I DNA/RNA helicase 4 Hypothetical proteins Uncharacterized protein (DUF4415 family)
C7	Whole contig (14 kb length)	**S:** 70% **W:** 61–62% **Sp:** 60–68%	Iron complex outermembrane receptor protein Iron complex transport system substrate-binding protein 5-Methyltetrahydropteroyltriglutamate-homocysteine methyltransferase Ribonucleoside-triphosphate reductase Pyruvate formate lyase activating enzyme Iron complex transport system permease protein Iron complex transport system ATP-binding protein Iron complex transport system substrate-binding protein Type I restriction enzyme R subunit
C8	Whole contig (3 kb length)	**S:** 25% **W:** 10% **Sp:** 11–22%	Threonine dehydrogenase-like Zn-dependent dehydrogenase Anthranilate phosphoribosyltransferase Hypothetical protein
C9	1 bp (2 kb length)	**S:** 27% **W:** 31–36% **Sp:** 28–34%	Hypothetical protein Type I restriction enzyme M protein
C11	1 bp (7 kb length)	**S:** 70% **W:** 58–63% **Sp:** 62–68%	DNA repair protein RadC F-type H+-transporting ATPase subunit gamma F-type H+-transporting ATPase subunit alpha F-type H+-transporting ATPase subunit b

			F-type H+-transporting ATPase subunit c F-type H+-transporting ATPase subunit a F1-F0 ATPase (N-ATPase) AtpR subunit ATP synthase protein I F-type H+-transporting ATPase subunit epsilon F-type H+-transporting ATPase subunit beta
C20	~79.5 kb (~9 kb length)	**S:** 65% **W:** 65–70% **Sp:** 67–77%	Phosphatidylinositol alpha-1,6-mannosyltransferase 4 Glycosyltransferases involved in cell wall biosynthesis 3 Hypothetical proteins Ubiquinone/menaquinone biosynthesis C-methylase UbiE
C23	Whole contig (11 kb length)	**S:** 59% **W:** 33–34% **Sp:** 33–44%	Cobalt-precorrin-5B (C1)-methyltransferase Cobalt-precorrin-5B (C1)-methyltransferase Precorrin-4 methylase/cobalamin biosynthesis protein CbiG Precorrin-6Y C5,15-methyltransferase (decarboxylating) Precorrin-3B methylase/precorrin isomerase Precorrin-2/cobalt-factor-2 C20-methyltransferase Sirohydrochlorin cobaltochelatase Uroporphyrin-III C-methyltransferase cobalt/nickel transport system ATP-binding protein Cobalt/nickel transport system permease protein Cobalt/nickel transport protein Cobalt/nickel transport system permease protein

C24	Whole contig (4 kb length)	**S:** 37% **W:** 34–37% **Sp:** 28–32%	Protease secretion system outer membrane protein Protease secretion system membrane fusion protein ATP-binding cassette subfamily C exporter for protease/lipase
C25	Whole contig (6 kb length)	**S:** 14% **W:** 16% **Sp:** 15–20%	Superfamily I DNA and/or RNA helicase IS5 family transposase Hypothetical protein Acyl-ACP thioesterase DDE family transposase Nitrite reductase/ring-hydroxylating ferredoxin subunit 3 Hypothetical proteins
C26	Whole contig (11 kb length)	**S:** 26% **W:** 9–10% **Sp:** 11–23%	Hypothetical protein Molecular chaperone GrpE Molecular chaperone DnaK (HSP70) SpoVK/Ycf46/Vps4 family AAA+-type ATPase Formylglycine-generating enzyme required for sulfatase activity Hypothetical protein Iron complex transport system substrate-binding protein Iron complex transport system permease protein Iron complex transport system ATP-binding protein Adenosylcobinamide amidohydrolase
C27	Whole contig (17 kb length)	**S:** 24% **W:** 9% **Sp:** 10–21%	Hypothetical protein Predicted amidohydrolase Sugar phosphate isomerase/epimerase Uncharacterized protein Cobaltochelatase CobN Iron complex outermembrane receptor protein

	Magnesium chelatase subunit D Magnesium chelatase subunit I Cobaltochelatase CobN Iron complex outermembrane receptor protein/hemoglobin/transferrin/lactoferrin receptor protein/vitamin B12 transporter

5.3.3 *Chlorobium* relative abundance in Ace Lake, Ellis Fjord, and Taynaya Bay

Among the Ellis Fjord and Taynaya Bay metagenomes, the relative abundance of *Chlorobium* OTU was highest in 45 m and 11 m depth metagenomes, respectively (Figure 5.3a). The oxic-anoxic interfaces of the two meromictic systems lie around these depths, respectively (Burke and Burton, 1988a; Gibson, 1999). The relative abundances of *Chlorobium* OTUs in the three filter fraction metagenomes (0.1–0.8, 0.8–3, and 3–20 μm) from Ellis Fjord Interface (14–49%) were comparable to *Chlorobium* OTU abundances in some Ace Lake Interface metagenomes from summer (39–84%) and spring (12–81%), except Oct 2014 (<1%) (Figure 5.3a). The *Chlorobium* OTU abundance in Taynaya Bay Interface metagenome (0.22–20 μm-filter) was comparatively low (6%), as low as *Chlorobium* OTU abundances in 0.1–0.8 and 0.8–3 μm-filter Ace Lake Interface metagenomes from winter (5–6%) (Figure 5.3a). Notably, Sterivex cartridges used for water sampling in Taynaya Bay capture smaller amounts of biomass than large format filters used for Ace Lake and Ellis Fjord water sampling. This would have affected the absolute abundance of *Chlorobium* OTU in Taynaya Bay samples (Figure 5.3d). However, the relative abundances of *Chlorobium* OTUs in the metagenomes from all three systems should be comparable, considering that they were normalised to the total abundance of all contigs in each metagenome. For example, the relative abundance of *Chlorobium* OTU in Taynaya Bay 11 m depth metagenome indicated that it contributed roughly 6% of the total metagenomic data generated from the biomass captured using Sterivex cartridges (Figure 5.3a). Similarly, relative abundance of *Chlorobium* OTU in 0.8–3 μm-filter metagenome from Ellis Fjord Interface indicated that it contributed nearly half of the metagenomic data generated from the biomass captured on a large format filter.

The microbial diversity of Ellis Fjord Interface was low (Simpson's index of diversity $1\text{-}\lambda' = 0.7$) in metagenomes from 0.1–0.8 and 0.8–3 μm-filter fractions. This was similar to Ace Lake Interface diversity in summer and spring (except Oct 2014) metagenomes

(1-λ′ < 0.7), when *Chlorobium* abundance was high (Chapter 3 Figures 3.6 and 3.9). On the other hand, the microbial diversity in Taynaya Bay Interface metagenome and 3–20 μm-filter metagenome from Ellis Fjord Interface was high (1-λ′ > 0.9), which was similar to Ace Lake Interface diversity in winter and Oct 2014 metagenomes, when *Chlorobium* population was comparatively low (Chapter 3 Figures 3.6 and 3.9). Apart from *Chlorobium*, the oxic-anoxic interfaces of Ellis Fjord and Taynaya Bay also contained members of *Proteobacteria* (mostly *Deltaproteobacteria* — 7–17%, 5%), *Atribacteria* (<1%, 11%), *Marinimicrobia* (0.1–9%, 1%), *Firmicutes* (1%, 5%), *Bacteroidetes* (5–8%, 9%), *Cloacimonetes* (3–6%, <1%), respectively.

The relative coverage pattern of EF_ref MAG (>99% genome completeness *Chlorobium* MAG from Ellis Fjord 45 m depth 3–20 μm-filter metagenome with highest total base pair count; section 5.2.3.2) was similar to the relative abundance pattern of *Chlorobium* OTU in Ace Lake Interface, Ellis Fjord, and Taynaya Bay metagenomes (Figures 5.3b, c). The mean read depth of *Chlorobium*, which indicated its absolute abundance, was much higher in Ellis Fjord 45 m depth metagenome (8741) than in Ace Lake Interface Dec 2014 metagenome (4539), but its relative abundance was higher in Ace Lake (45%) than in Ellis Fjord (38%) (Figures 5.3b, d). This indicated that *Chlorobium* probably contributed to a higher share of the total microbial population in Ace Lake Dec 2014 than in Ellis Fjord, although its absolute abundance in Ellis Fjord was nearly twice as much as that in Ace Lake Dec 2014. Although the Taynaya Bay and Ellis Fjord metagenomes represented spring samples, the mean read depth of *Chlorobium* in Taynaya Bay was low (258) compared to that in Ace Lake spring (except Oct 2014) and Ellis Fjord (>1000 and 8741, respectively). This difference in the absolute abundance of *Chlorobium* was probably because the Taynaya Bay metagenomes were generated from smaller amounts of biomass captured using Sterivex cartridges, whereas Ace Lake and Ellis Fjord metagenomes were generated from larger amounts of biomass captured on large format filters.

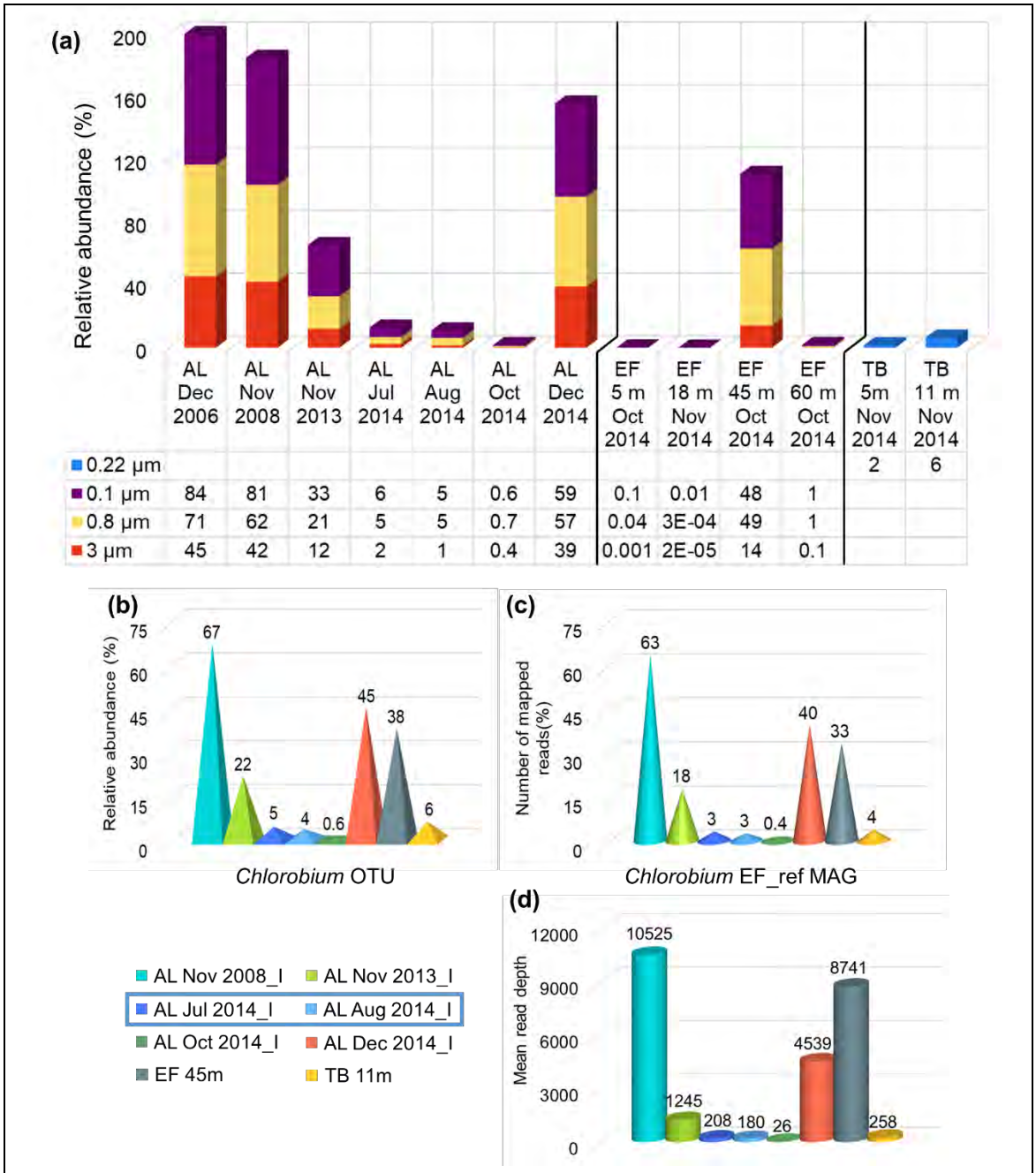


Figure 5.3 *Chlorobium* abundance and coverage distribution in Ace Lake, Ellis Fjord, and Taynaya Bay. (a) The stacked bar chart depicts the relative abundance of *Chlorobium* OTU in Ace Lake (AL) Interface metagenomes from different seasons (summer, Dec; winter, Jul and Aug; spring, Oct and Nov) and filter fractions, in Ellis Fjord (EF) metagenomes from different depths (5m, 18 m, 45 m, and 60 m) and filter fractions, and in Taynaya Bay (TB) metagenomes from different depths (5 m and 11 m). The y-axis indicates the relative abundance of *Chlorobium* OTU in different filter fractions (red: 3–20 µm, 3 µm; yellow: 0.8–3 µm, 0.8 µm; purple: 0.1–0.8 µm, 0.1 µm; blue: 0.22–20 µm, 0.22 µm) from various lake depths and time periods. The data table shows the percentage relative abundance values of *Chlorobium* OTU in metagenomes from each filter fraction, depth, and time period. The data in the bars and table are

arranged from top to bottom in the order of increasing filter size from 0.1 to 3 μm; the 0.22 μm data is shown separately. **(b)** The bar-chart shows the relative abundance distribution of *Chlorobium* OTU (coloured pyramids) in merged metagenomes from Ace Lake (AL) Interface (I) from different seasons and 45 m depth in Ellis Fjord (EF) as well as in the metagenome from 11 m depth in Taynaya Bay (TB). *Chlorobium* OTU relative abundances in merged metagenomes were calculated from the absolute abundances of *Chlorobium* contigs in 3–20, 0.8–3, and 0.1–0.8 μm-filter metagenomes relative to the total abundance of all contigs in the three metagenomes (formula described in Chapter 3 section 3.2.1). In the colour legend, the merged metagenomes from Jul and Aug 2014 are shown in a blue box to highlight their winter origin. **(c** and **d)** The bar-charts show the total number of reads mapped to (**c**, coloured cones) and mean read depth of (**c**, coloured cylinders) EF_ref MAG in the Ace Lake Interface, Ellis Fjord 45 m depth, and Taynaya Bay 11 m depth metagenomes. The y-axis in **(c)** indicates the total number of reads that aligned to the EF_ref MAG (representing relative coverage), whereas the y-axis in **(d)** denotes the mean of read depths of all nucleotide bases in EF_ref MAG (representing absolute abundance). Read depth values were calculated from the data in the base coverage files generated using Samtools v1.10. Ace Lake and Ellis Fjord samples from 0.22 μm Sterivex filters were not available. Similarly, Taynaya Bay samples from the three large format filters (3, 0.8, and 0.1 μm) were not available.

5.3.4 Analysis of genomic variation in Ace Lake, Ellis Fjord, and Taynaya Bay *Chlorobium*

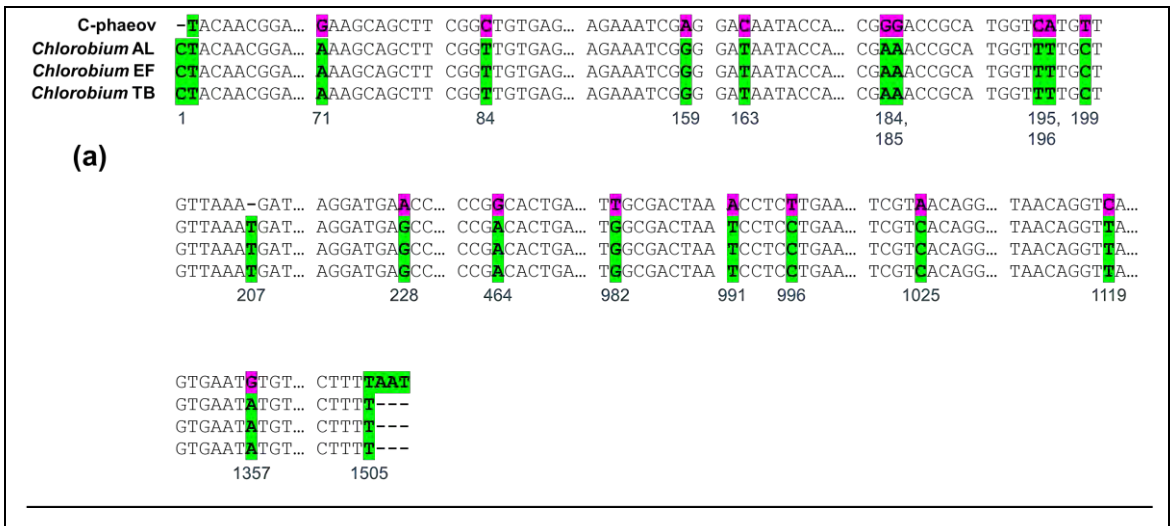
The genomic variation in *Chlorobium* from Ace Lake, Ellis Fjord, and Taynaya Bay and the potential endemicity of this Antarctic *Chlorobium* to the Vestfold Hills were assessed through (i) phylogenetic, AAI, and ANI analyses of the *Chlorobium* from the three systems and C-phaeov (described below in section 5.3.4.1); (ii) FR of Ace Lake, Ellis Fjord, and Taynaya Bay metagenomic reads to EF_ref MAG, which allowed for analysis of variations such as variable coverage regions and SNPs (described below in section 5.3.4.2); and (iii) comparative analysis of *Chlorobium* MAGs from Ace Lake, Ellis Fjord, and Taynaya Bay metagenomes and C-phaeov (described below in section 5.3.4.3).

5.3.4.1 *Chlorobium 16S rRNA* gene identity, BclA protein identity, ANI, AAI, and phylogeny

Similar to the Ace Lake *Chlorobium* OTU, the taxonomies of the *Chlorobium* OTUs identified in Ellis Fjord and Taynaya Bay metagenomes indicated that they were closely

related to C-phaeov. All Ace Lake, Ellis Fjord, and Taynaya Bay *Chlorobium 16S rRNA* genes were identical in their respective aquatic systems. The *16S rRNA* genes from 43 Ace Lake and six Ellis Fjord *Chlorobium* MAGs were identical and were 1,505 bp long. The *16S rRNA* genes from two Taynaya Bay *Chlorobium* MAGs were only 1,502 bp long (Figure 5.4a). However, an analysis of its *16S rRNA* gene sequence and its flanking nucleotides showed that the Taynaya Bay *Chlorobium* contained the complete 1,505 bp *16S rRNA* gene. The FR analysis of the *16S rRNA* gene of EF_ref MAG in Ace Lake, Ellis Fjord, and Taynaya Bay metagenomes also confirmed that the *Chlorobium 16S rRNA* genes from all three systems were identical. The ANI of all *Chlorobium* MAGs to each other was ≥99.9% over ≥92% alignment fraction.

The BclA protein sequences from 46 out of 47 Ace Lake *Chlorobium* MAGs were 366 aa long and were identical, but the BclA protein sequence from the Ace Lake *Chlorobium* MAG generated from Dec 2006_Interface_3 µm-filter metagenome was 389 aa long. An assessment of the DNA sequences of the two BclA proteins from Ace Lake *Chlorobium* showed that the variant *bclA* gene contained a single nucleotide insertion (thymine), which caused a frame-shift mutation. This mutation was present on only one read from Dec 2006 Ace Lake merged metagenome, indicating that the variant *bclA* gene probably resulted from a sequencing error. All BclA protein sequences from Ace Lake, Ellis Fjord, and Taynaya Bay *Chlorobium* appeared to be identical (Figure 5.4b).



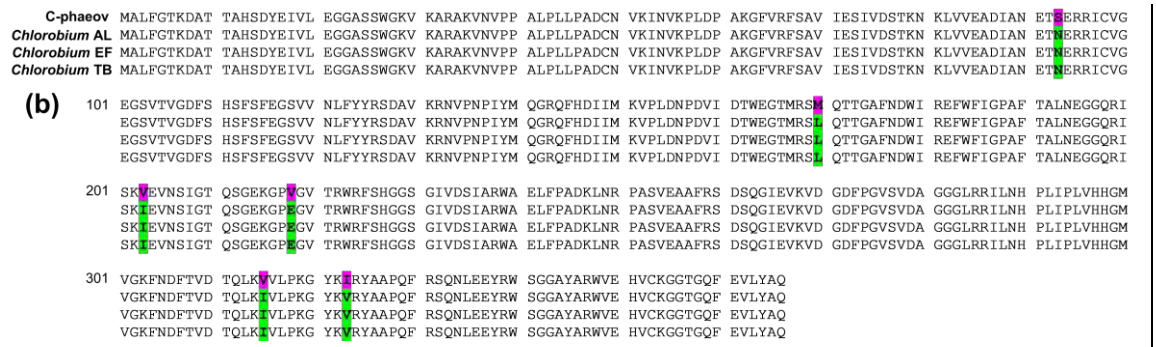
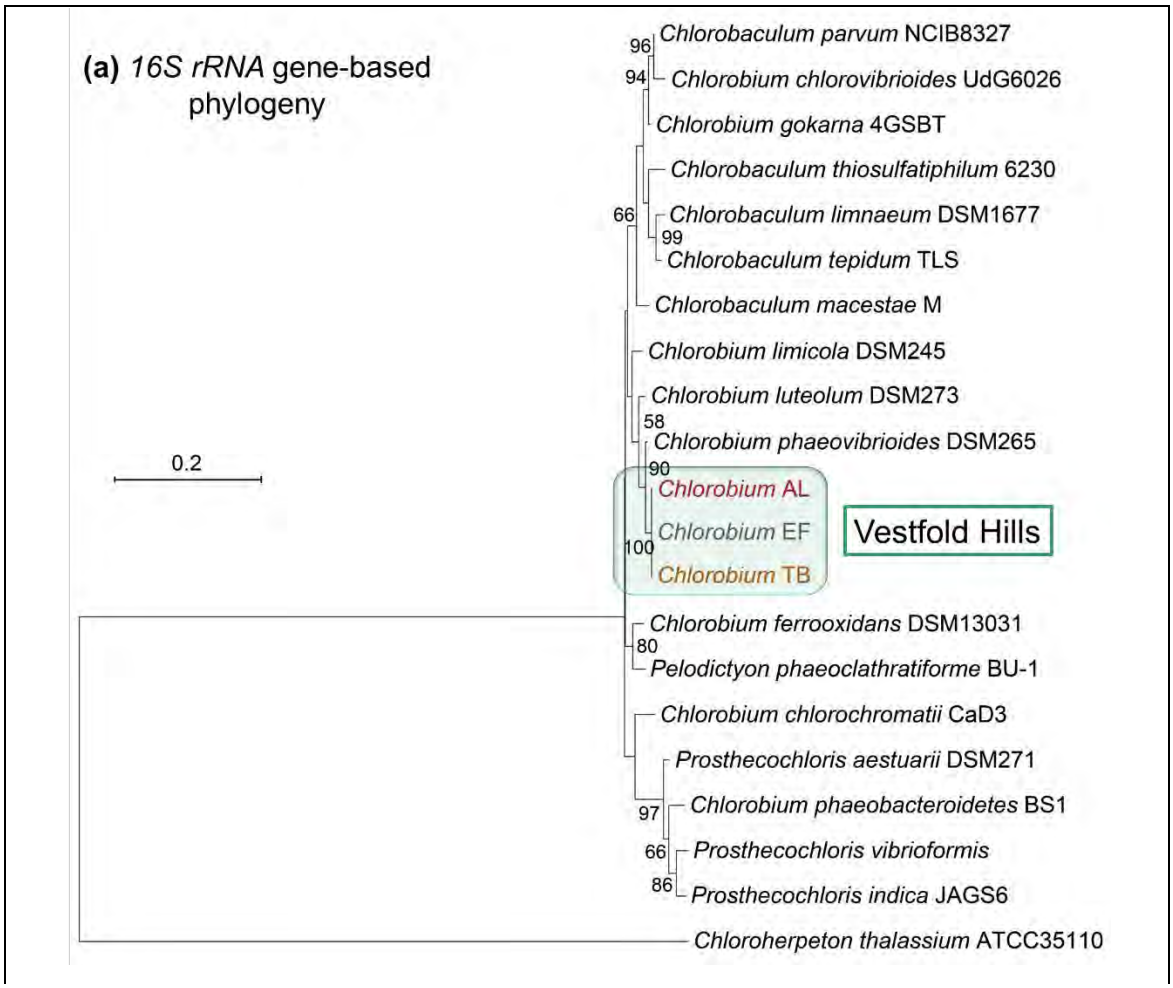


Figure 5.4 Comparison of *Chlorobium* marker genes from Ace Lake, Ellis Fjord, Taynaya Bay, and C-phaeov. The figure shows mismatches (pink) in *Chlorobium 16S rRNA* gene sequences **(a)** and BclA protein sequences **(b)** from Ace Lake (*Chlorobium* AL), Ellis Fjord (*Chlorobium* EF), Taynaya Bay (*Chlorobium* TB), and C-phaeov genome. **(a)** The complete *16S rRNA* gene sequences were 1,505–1,506 bp long, therefore only the regions with mutations are displayed here. Within the *16S rRNA* gene sequence, the dotted regions indicate sequence discontinuity and the numbers below the sequences mark the positions of the mismatches. The dashes at the ends of the sequences and at position 207 indicate sequence gaps. **(b)** The complete BclA protein sequences were 366 aa long and are shown here. The numbers on the left-side (101, 201, 301) indicate the sequence position on the protein.

The *16S rRNA* gene identity, BclA protein identity, ANI, and AAI of the Ace Lake, Ellis Fjord, and Taynaya Bay *Chlorobium* MAGs were also calculated against the C-phaeov genome. All *16S rRNA* genes from the *Chlorobium* MAGs had 99% identity to C-phaeov *16S rRNA* gene, with 17 nucleotide mismatches (Figure 5.4a). The BclA protein sequences from Ace Lake, Ellis Fjord, and Taynaya Bay *Chlorobium* MAGs were 98% similar to C-phaeov BclA protein, with six mismatches (Figure 5.4b). The *Chlorobium* MAGs had 85% ANI over 80–86% alignment fraction, and 89% AAI to C-phaeov genome.

Overall, the *16S rRNA* gene and BclA protein sequences as well as ANI of the *Chlorobium* from Ace Lake, Ellis Fjord, and Taynaya Bay suggested that the same *Chlorobium* species was present in the three Vestfold Hills systems, and that it was distinct from C-phaeov. This was also evident from the *16S rRNA* gene- and BclA protein-based phylogenetic analyses, which showed distinct clustering of Ace Lake, Ellis Fjord, and Taynaya Bay *Chlorobium*, separate from other members of *Chlorobiaceae* family analysed and closest to C-phaeov (Figure 5.5).


(a) 16S rRNA gene-based phylogeny
0.2
Chlorobaculum parvum NCIB8327
Chlorobium chlorovibrioides UdG6026
Chlorobium gokarna 4GSBT
Chlorobaculum thiosulfatiphilum 6230
Chlorobaculum limnaeum DSM1677
Chlorobaculum tepidum TLS
Chlorobaculum macestae M
Chlorobium limicola DSM245
Chlorobium luteolum DSM273
Chlorobium phaeovibrioides DSM265
Chlorobium AL
Chlorobium EF
Chlorobium TB
Vestfold Hills
Chlorobium ferrooxidans DSM13031
Pelodictyon phaeoclathratiforme BU-1
Chlorobium chlorochromatii CaD3
Prosthecochloris aestuarii DSM271
Chlorobium phaeobacteroidetes BS1
Prosthecochloris vibrioformis
Prosthecochloris indica JAGS6
Chloroherpeton thalassium ATCC35110
96
94
66
99
58
90
100
80
97
66
86

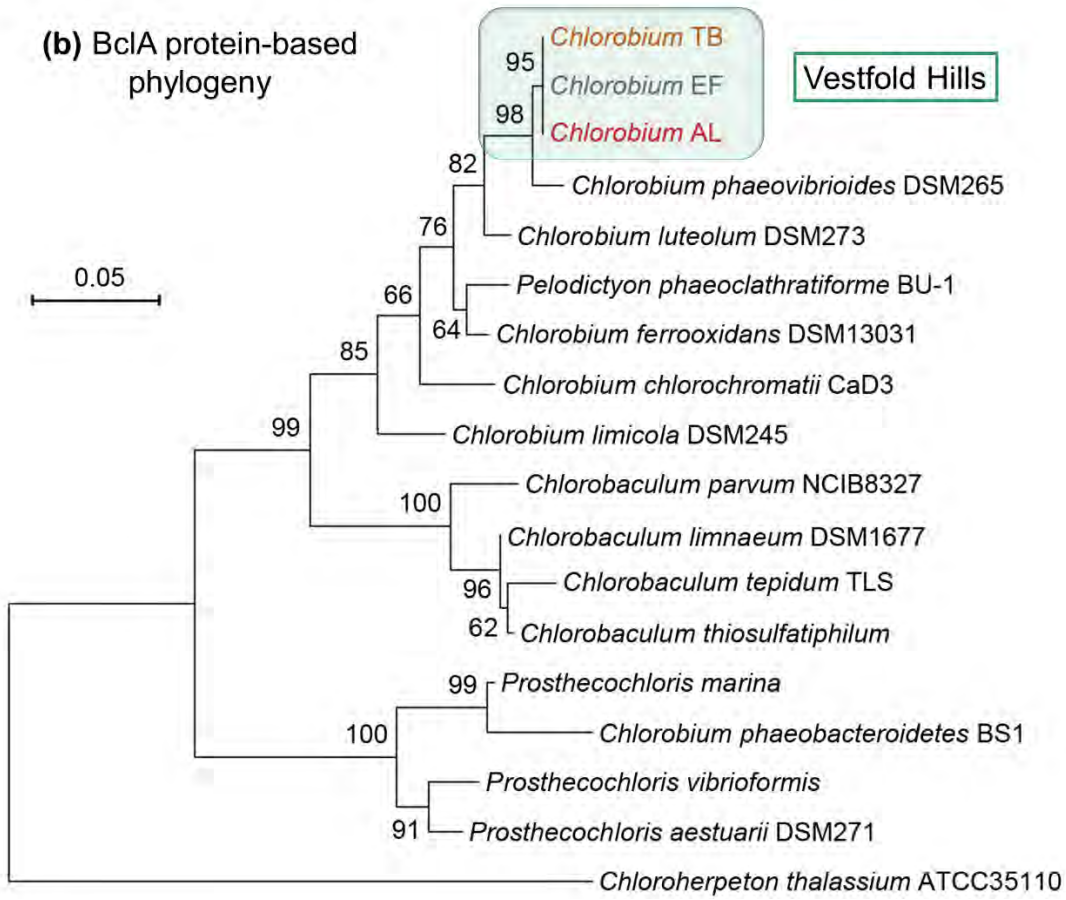


Figure 5.5 BclA protein and *16S rRNA* gene based phylogenetic analyses of *Chlorobium* MAGs. The maximum-likelihood trees show the **(a)** *16S rRNA* gene-based and **(b)** BclA protein-based phylogeny of the *Chlorobiaceae* family. The phylogenetic trees were prepared with MEGA X v10.1.7 using 1,000 bootstrap values. The *Chlorobium* from the Vestfold Hills are highlighted — Ace Lake, red font; Ellis Fjord, grey font; Taynaya Bay, yellow font. The trees are drawn to scale. The scale lengths in respective figures indicate the branch lengths. The numbers next to the branches represent bootstrap values showing the percentage of trees in which the taxa clustered together. Only bootstrap values greater than 50% are shown here.

5.3.4.2 FR analysis of *Chlorobium* EF_ref MAG in Ace Lake, Ellis Fjord, and Taynaya Bay metagenomes

The alignment of metagenomic reads from Ace Lake, Ellis Fjord, and Taynaya Bay to EF_ref MAG showed the presence of a number of variable coverage regions, with read depths higher or lower than the mean read depth of EF_ref MAG (Figure 5.6; Table 5.6). Interestingly, most LCRs identified in AL_ref MAG (Table 5.4) were also represented in the EF_ref MAG LCRs from all three systems, suggesting existence of similar *Chlorobium* subpopulations in Ace Lake, Ellis Fjord, and Taynaya Bay.

However, the abundances of these *Chlorobium* subpopulations varied in the three systems (Table 5.6). Similar to the LCRs in AL_ref MAG, the LCRs in EF_ref MAG contained genes coding for cell wall modification, cell defence, substrate transport, DNA repair, protein modification as well as enzymes involved in anaerobic pathway for cobalamin biosynthesis, cobinamide salvage, Na^+ or H^+ ion efflux, and cobalt/magnesium chelatases. Additional EF_ref MAG LCRs that were not present in AL_ref MAG mainly contained genes of unknown function as well as a few genes potentially involved in cell wall modification and some general function genes. Some of these LCRs in EF_ref MAG had very low read depth (<1%) in all three systems (Table 5.6).

For SNP analysis of *Chlorobium* from Ace Lake, Ellis Fjord, and Taynaya Bay, only the mutations that were present in at least 90% of the metagenomic reads aligned to the reference base (i.e., 0.9 variable frequency threshold) were considered. Of the 1,807 genes in EF_ref MAG, 68 had SNPs in metagenomes from Ace Lake only, 2 had SNPs in Taynaya Bay metagenome only, and 19 had SNPs in metagenomes from both Ace Lake and Taynaya Bay. Most of these mutations occurred in genes involved in cellular and metabolic functions, but a few were present in genes involved in cell wall modification, substrate transport, and some membrane proteins. Notably, no EF_ref MAG SNPs were observed in Ellis Fjord metagenome; this was similar to what was observed in AL_ref MAG, which showed no mutations in Ace Lake metagenomes. Nearly all of the mutations, except three — in a hypothetical gene, a gene for precorrin-3B methylase/precorrin isomerase and a gene for a TonB-dependent protein (on contigs C2, C10 and C28, respectively; Table 5.6), were present in non-variable coverage regions of EF_ref MAG, which indicated that all *Chlorobium* from Ace Lake and Taynaya Bay contained most of these mutations.

Together, the EF_ref MAG variable coverage regions and SNPs indicated that Ace Lake, Ellis Fjord, and Taynaya Bay had similar *Chlorobium* subpopulations of phylotypes and ecotypes, however, the genomic sequences of *Chlorobium* from Ellis Fjord and Taynaya Bay were more similar to each other than to Ace Lake *Chlorobium* (Table 5.6).

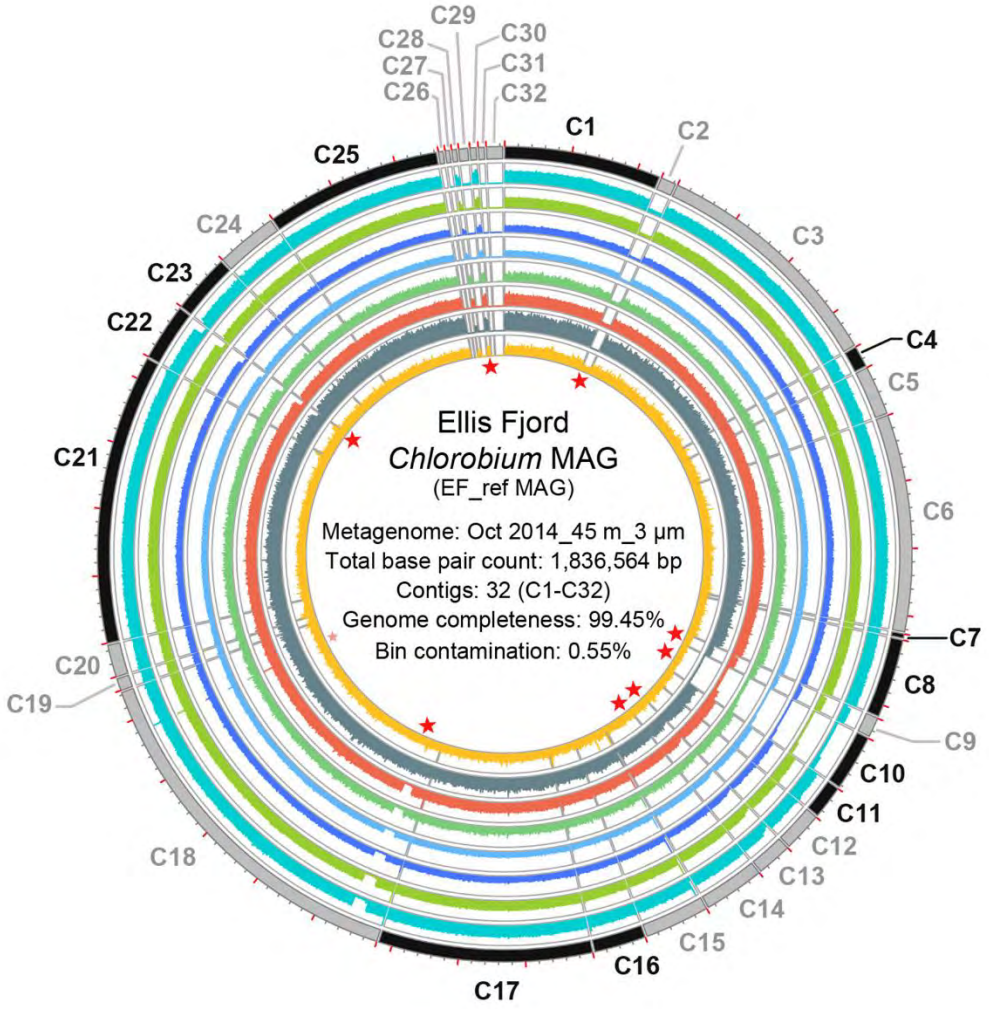


Figure 5.6 Coverage pattern of EF_ref MAG in Ace Lake, Ellis Fjord, and Taynaya Bay metagenomes. The circos plot depicts the coverage distribution of EF_ref MAG in Ace Lake, Ellis Fjord, and Taynaya Bay metagenomes (coloured rings). The outermost ring depicts the backbone of EF_ref MAG showing the Contigs 1–32 (C1–C32; Table 5.5). The MAG contigs were reordered in Mauve v2.4.0 using C-phaeov as the reference genome, but only C1–C25 had any matches to C-phaeov; C26–C32 were added to one end of the ordered contigs. The grey contigs represent forward strand MAG sequences, whereas black contigs represent complementary strand MAG sequences. The contig lengths varied between 3 to 272 kb length (Table 5.5), and the x-axis scales for all contigs and rings are drawn around the circumference of the outermost ring, with major ticks (red) at the beginning and every 50 kb length and minor ticks (dark grey) at every 10 kb length of each contig. Read depth values are plotted on linear scale y-axes. As the main purpose of the figure was to highlight the distribution of variable coverage regions, the y-axes scales of circos rings vary. Read depth values greater than the y-axis limits were truncated to the y-axis maximum values. Variable coverage regions are marked by red stars inside the innermost ring of the circos plot. Merged metagenomes and their y-axis scale ranges, outer to inner ring: AL Nov 2008_I (■, 0–20000); AL Nov 2013_I (■, 0–3000);

AL Jul 2014_I (■, 0–800); AL Aug 2014_I (■, 0–800); AL Oct 2014_I (■, 0–100); AL Dec 2014_I (■, 0–10000); EF 45m (■, 0–15000); TB 11m (■, 0–800).

Table 5.5 *Chlorobium* EF_ref MAG contigs. [A] The contig numbers (C1–C32) correspond to the contigs shown in Figure 5.6. [B] The contig IDs refer to the scaffold IDs provided by JGI's IMG system. [C] The read depths of the contigs in the 45m depth 3–20 μm-filter Ellis Fjord metagenome, from which EF_ref MAG was generated, were measured by JGI's IMG system.

Contig number[A]	Contig ID[B]	Length (bp)	GC content	Read depth[C]
C1	Ga0307987_1000015	116,997	0.54	1229
C2	Ga0307987_1001829	11,473	0.52	20
C3	Ga0307987_1000004	178,460	0.53	1150
C4	Ga0307987_1001038	15,792	0.53	1101
C5	Ga0307987_1000209	36,702	0.52	1086
C6	Ga0307987_1000006	162,002	0.53	1071
C7	Ga0307987_1015042	3,102	0.52	1055
C8	Ga0307987_1000070	58,906	0.51	922
C9	Ga0307987_1001178	14,683	0.5	463
C10	Ga0307987_1000158	41,780	0.52	76
C11	Ga0307987_1000445	24,876	0.51	889
C12	Ga0307987_1000306	29,343	0.5	883
C13	Ga0307987_1000397	26,253	0.5	844
C14	Ga0307987_1000132	45,348	0.5	871
C15	Ga0307987_1000116	48,411	0.5	860
C16	Ga0307987_1000188	38,654	0.52	941
C17	Ga0307987_1000007	159,945	0.52	977
C18	Ga0307987_1000001	271,782	0.52	984
C19	Ga0307987_1002717	9,042	0.52	1029
C20	Ga0307987_1000481	24,045	0.52	1009
C21	Ga0307987_1000002	216,430	0.54	1073
C22	Ga0307987_1000149	42,560	0.52	1030
C23	Ga0307987_1000137	44,410	0.54	1110
C24	Ga0307987_1000128	45,939	0.55	1176
C25	Ga0307987_1000010	131,454	0.54	1213
C26	Ga0307987_1012998	3,422	0.51	16
C27	Ga0307987_1013512	3,330	0.52	14

C28	Ga0307987_1012082	3,606	0.51	485
C29	Ga0307987_1004356	6,757	0.49	80
C30	Ga0307987_1008200	4,637	0.49	850
C31	Ga0307987_1008940	4,380	0.51	13
C32	Ga0307987_1001683	12,043	0.52	22

Table 5.6 Genes annotated on variable coverage regions of EF_ref MAG. [A] The EF_ref MAG contigs mentioned in the first column are described in Table 5.5. [B] The approximate starting positions and lengths of the variable coverage regions on the EF_ref MAG contigs are provided in the second column (the variable coverage regions are labelled as red stars in Figure 5.6). The LCRs are shown with a light blue background colour, whereas high coverage regions are shown with a light orange background colour. [C] The percentages shown are average of coverage values from metagenomes from Ace Lake (AL), Ellis Fjord (EF), and Taynaya Bay (TB) calculated across the regions specified in column two (section 5.2.3.3). [D] The genes shown in the table were annotated by JGI's IMG system. The regions are arranged from top to bottom in the order of their occurrence along the lengths of EF_ref MAG contigs.

EF_ref MAG contig[A]	Starting position and length of variable coverage region[B]	Metagenomes and % *Chlorobium* population in which observed[C]	AL_ref MAG genes annotated in the variable coverage region[D]
C2	Whole contig (11 kb length)	**AL:** 25–32% **EF:** 3% **TB:** 69%	Hypothetical protein Uncharacterized protein (DUF4415 family) 3 Hypothetical proteins ATP-dependent exoDNAse (exonuclease V) beta subunit/superfamily I DNA/RNA helicase PD-(D/E)XK nuclease superfamily protein Membrane protein YdbT with pleckstrin-like domain Restriction system protein Hypothetical protein
C9	Whole contig (15 kb length)	**AL:** 60–70% **EF:** 44%	Iron complex outermembrane receptor protein

		TB: 79%	Iron complex transport system substrate-binding protein 5-Methyltetrahydropteroyltriglutamate-homocysteine methyltransferase Ribonucleoside-triphosphate reductase Pyruvate formate lyase activating enzyme Iron complex transport system permease protein Iron complex transport system ATP-binding protein Iron complex transport system substrate-binding protein Type I restriction enzyme R subunit
C10	Whole contig 1 bp (31 kb length)	**AL:** 9–26% **EF:** 7% **TB:** 78%	Iron complex outermembrane receptor protein Cobaltochelatase CobN Magnesium chelatase subunit I Magnesium chelatase subunit D Iron complex outermembrane receptor protein Cobaltochelatase CobN Uncharacterized protein Sugar phosphate isomerase/epimerase Predicted amidohydrolase Adenosylcobinamide amidohydrolase Iron complex transport system ATP-binding protein Iron complex transport system permease protein Iron complex transport system substrate-binding protein Hypothetical protein Formylglycine-generating enzyme required for sulfatase activity SpoVK/Ycf46/Vps4 family AAA+-type ATPase

			Molecular chaperone DnaK (HSP70) Molecular chaperone GrpE 2 Hypothetical proteins Anthranilate phosphoribosyltransferase Threonine dehydrogenase-like Zn-dependent dehydrogenase
	31 kb (11 kb length)	**AL:** 29–59% **EF:** 8% **TB:** 72%	Hypothetical protein Cobalt-precorrin-5B (C1)-methyltransferase Cobalt-precorrin-5B (C1)-methyltransferase Precorrin-4 methylase/cobalamin biosynthesis protein CbiG Precorrin-6Y C5,15-methyltransferase (decarboxylating) Precorrin-3B methylase/precorrin isomerase Precorrin-2/cobalt-factor-2 C20-methyltransferase Sirohydrochlorin cobaltochelatase Uroporphyrin-III C-methyltransferase Cobalt/nickel transport system ATP-binding protein Cobalt/nickel transport system permease protein
C13	1 bp (7 kb length)	**AL:** 59–71% **EF:** 69% **TB:** 91%	DNA repair protein RadC F-type H+-transporting ATPase subunit gamma F-type H+-transporting ATPase subunit alpha F-type H+-transporting ATPase subunit b F-type H+-transporting ATPase subunit c F-type H+-transporting ATPase subunit a F1-F0 ATPase (N-ATPase) AtpR subunit ATP synthase protein I

			F-type H+-transporting ATPase subunit epsilon F-type H+-transporting ATPase subunit beta
C14	3.6 kb (722 bp length)	**AL:** 24–42% **EF:** 77% **TB:** 36%	DNA-binding response OmpR family regulator
C18	17 kb (11 kb length)	**AL:** 15–20% **EF:** 94% **TB:** >100%	IS5 family transposase Hypothetical protein Acyl-ACP thioesterase DDE family transposase Nitrite reductase/ring-hydroxylating ferredoxin subunit Hypothetical protein Uncharacterized protein YPO0396 Uncharacterized protein DUF4194 Uncharacterized protein DUF3375
	~239 kb (269 bp length)	**AL:** >100% **EF:** >100% **TB:** 93%	Hypothetical protein
C22	~36 kb (6 kb length)	**AL:** 25–34% **EF:** 66% **TB:** >100%	ATP-binding cassette subfamily C exporter for protease/lipase ATP-binding cassette subfamily C exporter for protease/lipase Protease secretion system membrane fusion protein Protease secretion system outer membrane protein Transposase InsO family protein
C26	Whole contig (3 kb length)	**AL:** <1% **EF:** <1% **TB:** 0%	Hypothetical protein FAD/FMN-containing dehydrogenase/Fe-S oxidoreductase Dihydroxy-acid dehydratase
C27	Whole contig (3 kb length)	**AL:** ≤2% **EF:** <1% **TB:** <1%	Sialate O-acetylesterase Polygalacturonase

C28	Whole contig (4 kb length)	**AL:** 56–64% **EF:** 37% **TB:** 64%	Hypothetical protein Hemoglobin/transferrin/lactoferrin receptor protein Hypothetical protein
C29	Whole contig (7 kb length)	**AL:** 8–12% **EF:** 8% **TB:** 1%	3 Hypothetical proteins Predicted dehydrogenase/threonine dehydrogenase-like Zn-dependent dehydrogenase Heparinase superfamily protein Hypothetical protein UDP-N-acetyl-D-mannosaminuronic acid dehydrogenase
C31	Whole contig (4 kb length)	**AL:** <1% **EF:** <1% **TB:** <1%	3 Hypothetical proteins
C32	Whole contig (12 kb length)	**AL:** ≤1% **EF:** <1% **TB:** <1%	Alpha-L-fucosidase 4 Hypothetical proteins Heparinase II/III-like protein Tol biopolymer transport system component/Tol biopolymer transport system component

5.3.4.3 Comparative analysis of C-phaeov and *Chlorobium* MAGs from Ace Lake, Ellis Fjord, and Taynaya Bay

The alignment of 32 Ace Lake, five Ellis Fjord, and two Taynaya Bay *Chlorobium* MAGs against the C-phaeov genome showed overall low identity nucleotide matches (<90%), with multiple alignment gaps and variable sequence regions (Figure 5.7). Many of the C-phaeov genes in the alignment gap regions were associated with various transposases and hypothetical proteins. However, some of them coded for metabolic proteins involved in thiosulfate oxidation (*sox* gene cluster containing *soxA*, *soxB*, *soxX*, *soxY*, *soxZ*), assimilatory sulfate reduction (*cysC*, *cysD*, *cysN*), and pilus assembly, none of which were present in the *Chlorobium* MAGs from the three systems.

Although the sequence alignment pattern of all *Chlorobium* MAGs to the C-phaeov genome was similar, a few regions varied between their genomes (Figure 5.7). A 1.6 kb long, low identity (80%) region starting at ~124 kb length of C-phaeov genome was

observed in only three Ace Lake and one Ellis Fjord *Chlorobium* MAGs analysed (Figure 5.7). The region contained genes associated with membrane proteins and transporters (EmrAB-TolC complex) and a transcriptional regulator. A 5.4 kb long, high identity (98%) region starting at ~192 kb length of C-phaeov genome was present in only two Ace Lake *Chlorobium* MAGs and contained *16S rRNA*, *23S rRNA*, and *5S rRNA* genes (Figure 5.7). However, a closer inspection of this region in different *Chlorobium* MAGs showed that the sequence was identical in all MAGs. The differences in alignment identity were because parts of the region were on different contigs in the two MAGs with high identity matches, but on a single contig in the MAGs with low identity matches. Another region starting at ~716 kb length of C-phaeov genome was present in 20 Ace Lake and one Taynaya Bay *Chlorobium* MAGs analysed and was ~9 kb long with low identity (80%) matches to C-phaeov (Figure 5.7). This region coincided with the ~9 kb length variable sequence region of AL_ref MAG that contained genes associated polysaccharide transporters and cell wall biosynthesis and modification (section 5.3.2.1).

The analysis of *Chlorobium* MAG contigs that had no alignment to the C-phaeov genome showed that the Vestfold Hills *Chlorobium* contained genes associated with the anaerobic pathway for cobalamin biosynthesis, cobalt transporters, putative vitamin B12 transporters, cobalt/magnesium chelatases, and N-ATPases, which were absent from C-phaeov. Interestingly, these genes had low coverage in Ace Lake, Ellis Fjord, and Taynaya Bay *Chlorobium*, suggesting that they were present in only a subpopulation of *Chlorobium* from the Vestfold Hills (section 5.3.2.2). Multiple genes coding for glycosyltransferases involved in cell wall biosynthesis were also annotated on the MAG contigs that did not match the C-phaeov genome. Notably, the *Chlorobium* MAGs contained genes for a subtype I-E CRISPR-Cas system (*cas3*, *casA*, *casB*, *casE*, *casC*, *casD*, *cas1*, *cas2*), unlike C-phaeov that contained genes for a subtype I-C CRISPR-Cas system (*cas3''*, *cas3'*, *cas5*, *cas8c*, *cas7*, *cas4*, *cas1*, *cas2*).

Overall, the comparison of Ace Lake, Ellis Fjord, and Taynaya Bay *Chlorobium* with C-phaeov (a non-Antarctic species) showed that the Vestfold Hills *Chlorobium* was distinct from C-phaeov, not only in terms of its genomic sequence but also its functional potential and viral defence capacity.

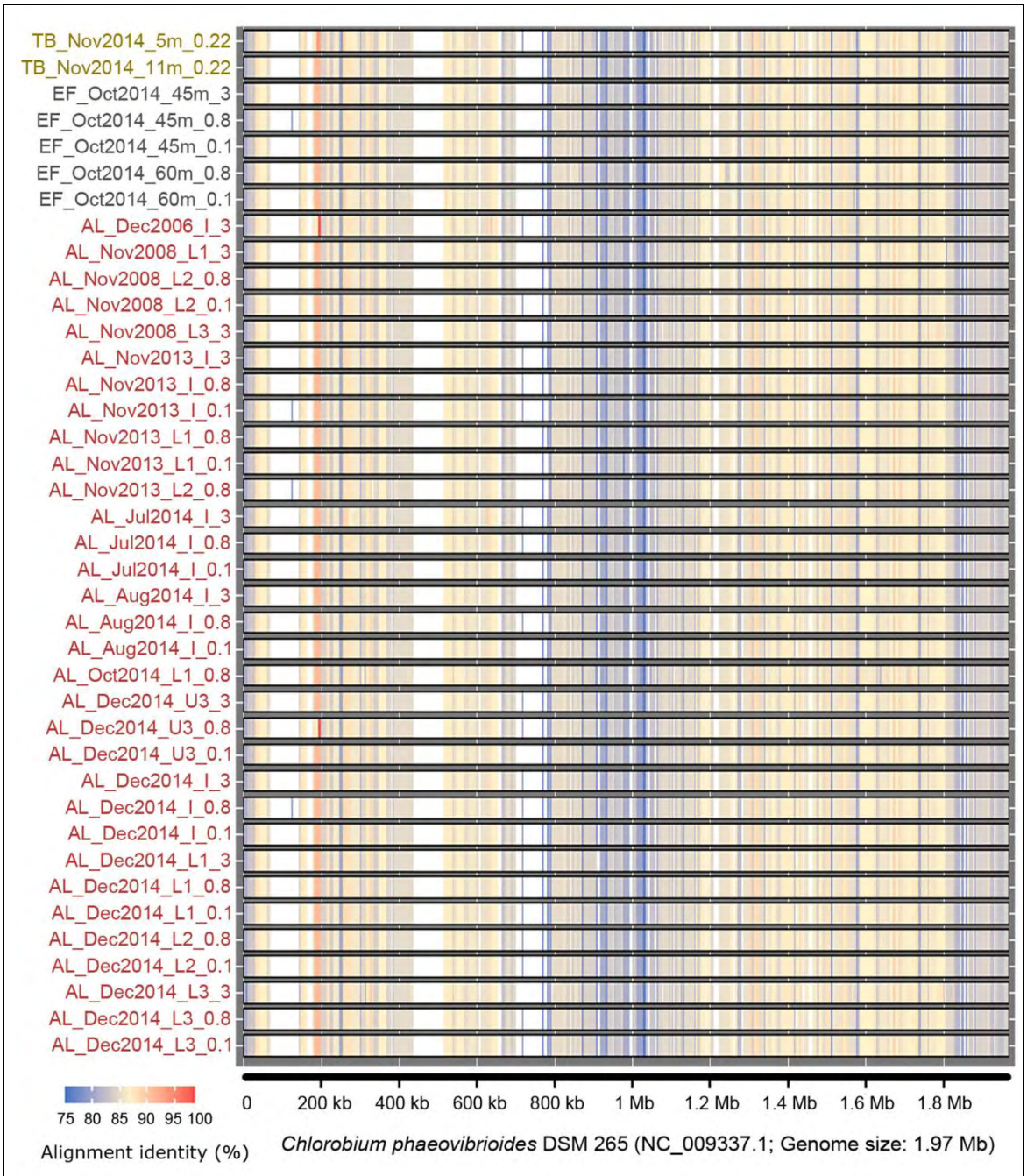


Figure 5.7 Sequence comparison of *Chlorobium* MAGs with C-phaeov genome. The figure shows the alignment of 32 Ace Lake (red font), five Ellis Fjord (grey font), and two Taynaya Bay (dark yellow font) *Chlorobium* MAGs to C-phaeov genome (x-axis, black line). The y-axis shows the metagenomes (sample collection site and time, lake depth, and filter fraction) from which the MAGs were generated. In each system (Ace Lake, Ellis Fjord, Taynaya Bay), the metagenomes are arranged from top to bottom in the order of sample collection time period from 2006 to 2014 and lake depth. The Ace Lake *Chlorobium* MAG from 'Dec 2006_Interface_3.0' contained a variant BclA protein sequence (section 5.3.4.1) and its genome completeness was 98%. All other MAGs shown here had ≥99% genome completeness. The white regions in the alignment bands indicate that the MAGs had no matches to those regions of

the reference genome. The gradient bar indicates the percentage alignment identity. Filter fractions: 3, 3–20 μm; 0.8, 0.8–3 μm; 0.1, 0.1–0.8 μm; 0.22, 0.22–20 μm. Ace Lake depths: U3, Upper 3; I, Interface; L1, Lower 1; L2, Lower 2; L3, Lower 3.

5.3.5 Analysis of defence system genes, potential viruses, and CRISPR spacers

5.3.5.1 Defence genes in *Chlorobium* MAGs

The genes annotated in the *Chlorobium* MAGs were manually parsed to identify the defence genes (Table 5.7). The *Chlorobium* MAGs contained the methyltransferase and restriction enzyme genes of a type I R-M system as well as two type IV restriction endonuclease genes. A cluster of two type III restriction enzyme gene fragments were also annotated in the *Chlorobium* MAGs but they could not be verified as restriction enzymes by manual reannotation of their gene function. A subtype I-E CRISPR-Cas defence system was present in the Ace Lake *Chlorobium* MAGs, containing the core *cas* genes *casA* (or *cse1*) and *casB* (or *cse2*) (Table 5.7). The defence system subtype classification was based on a recently published CRISPR-Cas system classification (Makarova et al, 2020). The genes in the CRISPR-Cas defence cassette were arranged in the order — *cas3*, *casA*, *casB*, *casE*, *casC*, *casD*, *cas1*, *cas2*, followed by a CRISPR spacer array. No genes associated with BREX or DISARM defence systems could be identified in the *Chlorobium* MAGs, however, T-A system genes coding for a ParDE type II T-A system, a RelF family antitoxin, a BrnA family antitoxin, and an AbiEi antitoxin were present in the *Chlorobium* MAGs (Table 5.7). As T-A system proteins containing HEPN domain can be potentially associated with ABI mechanism (Koonin et al, 2017), all genes coding for predicted and uncharacterised HEPN domain containing proteins in the *Chlorobium* MAGs were manually reannotated. Although a HEPN domain-containing gene was identified in *Chlorobium* MAGs, its reannotation did not reveal potential involvement in the ABI mechanism. Together, these findings indicated that the Vestfold Hills *Chlorobium* had intracellular defence systems such as CRISPR-Cas system, type I and type IV R-M systems, and AbiE T-A system that it could use for defence against viruses (Table 5.7).

Table 5.7 Defence genes annotated in *Chlorobium* MAGs. [A] The initial annotations of the MAG genes were performed by JGI's IMG system. [B] The gene functions were verified against reference proteins in the UniProtKB/Swiss-Prot database. The proteins with low alignment or

no hits to UniProtKB/Swiss-Prot database proteins were realigned to reference proteins in the UniProtKB database or RefSeq protein database.

Defence system	Subsystem type	Gene annotation[A]	Gene function and protein sequence identity (%)[B]
R-M system	**Type IV** restriction endonuclease	Restriction system protein	39%; Mrr restriction system protein *Escherichia coli*
		Restriction system protein	38% Mrr restriction system protein *Escherichia coli*
	Type I R-M system	Type I restriction enzyme M protein	38% Putative type I restriction enzyme MpnORFDP M protein *Mycoplasma pneumoniae*
		Type I restriction enzyme R subunit	34% Type-1 restriction enzyme R protein *Staphylococcus saprophyticus* subsp. *Saprophyticus*
CRISPR-Cas system	**Subtype I-E** CRISPR-Cas system	CRISPR-associated protein Cas2	28% CRISPR-associated endoribonuclease Cas2 *Escherichia coli*
		CRISPR-associated protein Cas1	79% CRISPR-associated endonuclease Cas1 *Chlorobaculum tepidum*
		CRISPR system Cascade subunit CasD	29% CRISPR system Cascade subunit CasD *Escherichia coli*
		CRISPR system Cascade subunit CasC	34% CRISPR system Cascade subunit CasC *Escherichia coli*
		CRISPR system Cascade subunit CasE	24% CRISPR system Cascade subunit CasE *Escherichia coli*
		CRISPR system Cascade subunit CasB	31% CRISPR-associated protein Cse2 *Thermus thermophilus*
		CRISPR system Cascade subunit CasA	61% CRISPR-associated protein, Cse1 family *Prosthecochloris aestuarii* (UniProtKB)

		CRISPR-associated endonuclease/helicase Cas3	31% CRISPR-associated nuclease/helicase Cas3 *Streptococcus thermophilus*
BREX system	Not found	-	-
DISARM system	Not found	-	-
T-A system	**ParDE type II** T-A system	Antitoxin ParD1/3/4	35% Antitoxin ParD *Mycobacterium bovis*
		Toxin ParE1/3/4	28% Toxin ParE3 *Caulobacter vibrioides*
	Antitoxin module of a **RelFG type II** T-A system	PHD/YefM family antitoxin component YafN of YafNO toxin-antitoxin module	30% Antitoxin RelF *Mycobacterium tuberculosis*
	Antitoxin module of a **BrnTA type II** T-A system	Uncharacterized protein (DUF4415 family)	94% BrnA antitoxin family protein *Chlorobium limicola* (RefSeq)
	Antitoxin module of an **AbiE type IV** T-A system	Transcriptional regulator with AbiEi antitoxin domain of type IV toxin-antitoxin system	53% Type IV toxin-antitoxin system AbiEi family antitoxin domain-containing protein *Chlorobium phaeobacteroides* (RefSeq)

5.3.5.2 Analysis of CRISPR spacers from Ace Lake, Ellis Fjord, and Taynaya Bay *Chlorobium*

A total of 258 CRISPR spacer and 28 CRISPR repeat sequences were collected from *Chlorobium* MAGs and OTUs from 18 Ace Lake, three Ellis Fjord, and three Taynaya Bay metagenomes (Table 5.8; Appendix H: Table H3). The spacer sequences were numbered from Spc1 to Spc258, whereas the repeat sequences were numbered from Rpt1 to Rpt28. In three other Ace Lake *Chlorobium* MAGs, the presence of CRISPR spacers (a total of 8–14 CRISPR spacers) was determined during the analysis of the multiple sequence alignment of all *Chlorobium* MAGs, but the sequences of the spacers could not be determined (Table 5.8). On average, the spacer sequences were 33 bp long

and the repeat sequences were 28 bp long. The comparison of spacers and repeats identified in *Chlorobium* from Ace Lake, Ellis Fjord, and Taynaya Bay showed that a large number of them were exclusive to a system, but some were common to *Chlorobium* from two or all three systems (Figure 5.8). For example, Rpt3 was present in *Chlorobium* from all three systems, as were 14 spacer sequences — Spc99–103 and Spc108–116 (Figure 5.8).

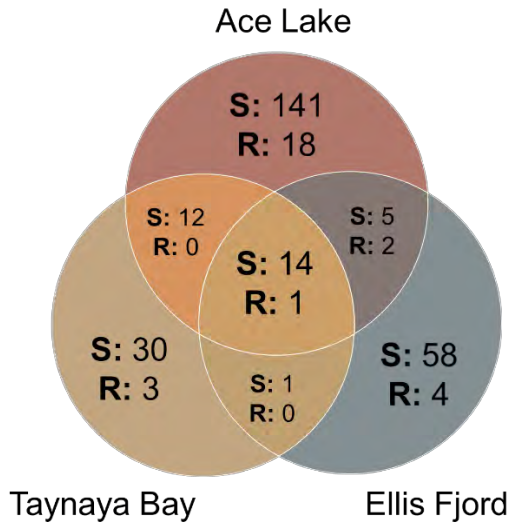


Figure 5.8 Comparison of spacers and repeats identified in *Chlorobium* from Ace Lake, Ellis Fjord, and Taynaya Bay. The Venn diagram depicts the number of distinct spacers (S) and repeats (R) that were exclusive to or common between *Chlorobium* from Ace Lake (red), Ellis Fjord (grey), and Taynaya Bay (yellow). The data in the overlapping zones indicate the number of distinct spacers and repeats that were present in *Chlorobium* from two or all three systems. For example, five spacers and two repeats were identified in *Chlorobium* from Ace Lake and Ellis Fjord, but not in Taynaya Bay *Chlorobium*.

In Ace Lake, some spacer and repeat sequences were found only in *Chlorobium* from specific time periods irrespective of season, e.g., Rpt7–9 and Spc49–51 were present only in Jul 2014 winter metagenome and Rpt16–21 were present in Dec 2006 summer, but not in Dec 2014 summer metagenomes. On the other hand, some of the spacer and repeat sequences were present in *Chlorobium* from different time periods, such as Rpt1, 2, 3, 5 and Spc9–22, Spc38–41 were identified in both summer and spring (Table 5.8). A higher number of spacers (>10) were generally found in Ace Lake *Chlorobium* from Lower 2 and 3 zone metagenomes, which probably represented dead *Chlorobium* cells settling to the bottom of the lake, as it is unlikely that *Chlorobium* would grow at such low depths with no sunlight (Table 5.8). Moreover, most of the CRISPR arrays were

present at the ends of *Chlorobium* contigs, which might be attributed to the issues usually encountered during sequencing and assembly of regions containing repeats, in this case CRISPR repeats. This could lead to the truncation of the CRISPR arrays in contigs containing the CRISPR *cas* genes, and there might be additional *Chlorobium*-associated spacers in the metagenomic data from Ace Lake, Ellis Fjord, and Taynaya Bay that could not be included in the *Chlorobium* MAGs and OTUs.

Overall, the spacer data from Ace Lake *Chlorobium* did not show a clear seasonal pattern of spacer acquisition (Table 5.8). Moreover, the comparison of spacer data from Ace Lake, Ellis Fjord, and Taynaya Bay *Chlorobium* showed some common spacers, which might indicate the existence of similar *Chlorobium* virus populations in the three systems (Figure 5.8).

Table 5.8 Spacers and repeats identified in CRISPR arrays of Ace Lake, Ellis Fjord, and Taynaya Bay *Chlorobium*. [A] The *Chlorobium* CRISPR spacers and repeats were identified in metagenomes from Ace Lake, Ellis Fjord, and Taynaya Bay, with the background colours indicating the seasons — summer (red), winter (blue), and spring (green). [B] The 28 CRISPR repeats and 258 CRISPR spacers identified in the *Chlorobium* MAGs were numbered as Rpt1 to Rpt28 and Spc1 to Spc258, respectively (Appendix H: Table H3). The spacers and repeats identified in multiple MAGs are highlighted in bold. * The presence of CRISPR spacers in some Ace Lake *Chlorobium* MAGs was discerned from the multiple sequence alignment of all *Chlorobium* MAGs to AL_ref MAG, but the sequences of the spacers could not be determined. † Some Taynaya Bay MAGs (with suffix '-nbfc') were generated from metagenomes assembled directly from filtered reads and not error-corrected reads (see section 5.2.1 for methods used for Taynaya Bay metagenome assembly). Ace Lake depths: U3, Upper 3; I, Interface; L1, Lower 1; L2, Lower 2; L3, Lower 3.

System	Metagenome[A]	Total spacers identified	CRISPR repeats[B]	CRISPR spacers[B]
Ace Lake	Dec 2006_I_3 µm	2	**Rpt1**	Spc164–Spc165
	Dec 2006_L1_0.8 µm	2	Rpt16–Rpt18	Spc166–Spc167
	Dec 2006_L2_0.8 µm	2	Rpt19–Rpt21	Spc168–Spc169
	Nov 2008_L2_0.8 µm	22	**Rpt1**, **Rpt2**	Spc1–Spc8, **Spc9–Spc22**
	Nov 2008_L3_3 µm	15	**Rpt3**, Rpt4	Spc23–Spc37
	Nov 2008_L3_0.8 µm*	4-6	-	-

	Nov 2013_I_0.1 µm	7	**Rpt1**	**Spc9–Spc15**
	Nov 2013_L1_0.1 µm	7	**Rpt1**, **Rpt5**	**Spc38–Spc41**, Spc42–Spc44
	Nov 2013_L2_0.8 µm	2	Rpt6	Spc45, Spc46
	Nov 2013_L3_0.8 µm	2	**Rpt3**	Spc47, **Spc48**
	Jul 2014_I_0.1 µm	3	Rpt7–Rpt9	Spc49–Spc51
	Oct 2014_I_0.8 µm*	2-4	-	-
	Oct 2014_L1_0.8 µm*	2-4	-	-
	Dec 2014_U3_0.1 µm	5	**Rpt1**	**Spc38–Spc41**, **Spc52**
	Dec 2014_I_0.1 µm	7	**Rpt1**	**Spc9–Spc15**
	Dec 2014_L1_3 µm	7	**Rpt1**, **Rpt2**	**Spc16–Spc22**
	Dec 2014_L1_0.8 µm	4	Rpt10–Rpt12	Spc53–Spc56
	Dec 2014_L2_3 µm	60	**Rpt3**	Spc57–Spc82, **Spc83**, Spc84, **Spc85-92**, Spc93, **Spc94**, **Spc95**, Spc96, Spc97, **Spc98–116**
	Dec 2014_L2_0.8 µm	27	**Rpt1**	Spc117–Spc143
	Dec 2014_L2_0.1 µm	11	**Rpt1**	**Spc38–Spc41**, **Spc52**, Spc142–Spc149
	Dec 2014_L3_0.8 µm	14	**Rpt5**, Rpt13, **Rpt14**, Rpt15	Spc150–Spc163
Ellis Fjord	Oct 2014_60 m_3 µm	3	Rpt22–Rpt23	Spc170–Spc172
	Oct 2014_60 m_0.8 µm	56	**Rpt1**, **Rpt14**, Rpt27–Rpt28	**Spc48**, Spc179–Spc232
	Oct 2014_60 m_0.1 µm	20	**Rpt3**	**Spc99–Spc104**, Spc173, **Spc105–Spc116**, **Spc174**
Taynaya Bay	Nov 2014_5 m	52	**Rpt3**	Spc233–Spc253, **Spc83**, **Spc85–Spc89**, Spc254–Spc256, **Spc90–Spc92**, **Spc94–**

				Spc95, **Spc98**, Spc257, **Spc99–Spc103**, **Spc108–Spc116**, Spc258
	Nov 2014_5 m-nbfc†	2	**Rpt3**	**Spc116**, **Spc174**
	Nov 2014_11 m-nbfc†	4	Rpt24–Rpt26	Spc175–Spc178

5.3.5.3 Potential viruses associated with Ellis Fjord and Taynaya Bay *Chlorobium*

The CRISPR spacer sequences present in Ellis Fjord and Taynaya Bay *Chlorobium* were used to identify their potential viruses; similar to the analysis performed to identify viruses potentially associated with Ace Lake *Chlorobium* (Chapter 3 section 3.3.5.6). A total of 79 and 58 CRISPR spacers were found in *Chlorobium* from Ellis Fjord and Taynaya Bay, respectively (Appendix H: Table H3). The comparison of Ellis Fjord *Chlorobium* contigs containing the spacer sequences with the spacer database led to the identification of eight viral contigs with 97% similarity to one of its spacer sequences Spc230. These viral contigs were from Ace Lake metagenomes and belonged to cl_248, which was shown to be a potential virus of Ace Lake *Chlorobium* (Chapter 3 section 3.3.5.6). However, none of the viral contigs in the Antarctic virus catalogue that were identified from Ellis Fjord metagenomes had any matches to Ellis Fjord *Chlorobium* spacers. The Antarctic virus catalogue contained only viral contigs of length ≥5 kb, and among the assembled contigs from Ellis Fjord 45 m depth, only 374 contigs were identified as viral contigs and included in the catalogue. Therefore, it is likely that the potential *Chlorobium* virus contigs from Ellis Fjord were not in the Antarctic virus catalogue.

As the Taynaya Bay metagenomes were not a part of the Antarctic virus catalogue or spacer database, a slightly different approach was applied to identify potential viruses of Taynaya Bay *Chlorobium*. Of the 58 Taynaya Bay *Chlorobium* spacers aligned against the Antarctic virus catalogue, nine spacers (Spc236, Spc238, Spc241, Spc243–Spc245, Spc249, Spc251, Spc252; Appendix H: Table H3) had matches to 23 viral contigs with ≥97% identity. Among these 23 viral contigs, 18 were from Ace Lake metagenomes and belonged to cl_1024 (14), sg_10581 (1), sg_14551 (1), sg_14796 (1), and sg_14959 (1). Notably, cl_1024 was identified as a potential *Chlorobium* virus in Ace Lake (Chapter 3 section 3.3.5.6). The remaining five viral contigs were from hypersaline Antarctic

systems such as Deep Lake and Rauer 13 Lake and belonged to cl_9176 (1), sg_1370 (1), sg_1648 (1), sg_1649 (1), and sg_1677 (1). However, none of the 23 viral contigs with matches to Taynaya Bay *Chlorobium* spacers were associated with Taynaya Bay viral contigs. As the Taynaya Bay viral contigs were identified from matches to the Antarctic virus catalogue, the Taynaya Bay viral contig data was probably incomplete. Therefore, the lack of matches between the *Chlorobium* spacers and viral contigs from Taynaya Bay could not be interpreted as absence of potential *Chlorobium* viruses in Taynaya Bay metagenomes.

The potential hosts of the viral contigs with matches to Taynaya Bay *Chlorobium* spacers were verified using the data in the spacer database. The potential hosts of the viral contigs belonging to clusters and singletons other than cl_1024 (as this was covered in Chapter 3) and with 100% identity matches to host spacers were assessed. The data showed that the viral contigs had a broad range of hosts, with most host contigs belonging to *Gammaproteobacteria* class and *Chlorobi* phylum (including *Chlorobium* OTU) and a few host contigs belonging to *Actinobacteria*, *Firmicutes*, *Betaproteobacteria*, *Deltaproteobacteria*, and *Verrucomicrobia* (Table 5.9). This was similar to what was observed for Ace Lake *Chlorobium* viruses (cl_1024, cl_248, sg_14554), which had a broad host range including *Gammaproteobacteria*. Altogether, the potential *Chlorobium* viruses from Ace Lake, Ellis Fjord, and Taynaya Bay belonged to similar viral clusters such as cl_1024 and cl_248, and these viruses were not specific to *Chlorobium* (Table 5.9; Chapter 3 Table 3.6).

Table 5.9 Host analysis of viral cluster and singletons with matches to Taynaya Bay *Chlorobium* spacers. The table includes data from Taynaya Bay *Chlorobium* spacer matches to the viral contigs. [A] The viral contig cluster (cl_9176) and singletons (sg_10581, sg_1370, sg_14551, sg_14796, sg_14959, sg_1648, sg_1649, sg_1677) shown in the table had matches to the spacers of Taynaya Bay *Chlorobium*. [B] The phylum/class of the host contigs are shown in the first column. The numbers in the brackets indicate the number of host contigs containing spacers that had 100% identity matches to at least one of the viral contigs. The abbreviations in the table represent taxonomies of the potential host contigs, including the Vestfold Hills *Chlorobium* (represented by CPv here), with red-highlighted taxonomies indicating <100% identity spacer matches to viral contigs; all other matches had 100% identity. Host contig taxonomies: AJ, *Alcanivorax jadensis*; CPb, *C. phaeobacteroides*; CPv, *C. phaeovibrioides*; D, *Desulfurivibrio* sp.; KP, *Klebsiella pneumoniae*; L, *Lactobacillus* sp.; LM, *Legionella massiliensis*; M, *Marinobacter* sp.; MA, *Marinobacter antarcticus*; ME, *Marinobacter* sp.

ELB17; P, *Polaromonas* sp.; PP, *Pseudomonas putida*; PS, *Pseudomonas stutzeri*; S, *Streptomyces* sp.; T, *Thauera* sp.; U, Unclassified; V, *Verrucomicrobium* sp. 3C; VC, *Vibrio cholerae*.

Host phylum/class (number of host contigs) [B]	Viral cluster and singletons[A]								
	cl_9176	sg_10581	sg_1370	sg_14551	sg_14796	sg_14959	sg_1648	sg_1649	sg_1677
Chlorobi (21)	CPv	CPb, CPv	CPv	CPb, CPv	CPb, CPv	CPb, CPv	CPv	CPv	CPv
Actinobacteria (1)					S				
Firmicutes (1)	L								
Betaproteobacteria (4)			P		T	T			
Deltaproteobacteria (1)			D						
Gammaproteobacteria (102)	M, KP, VC	M, KP, AJ	MA, M, ME, KP, LM, AJ, PP, PS, VC	MA, M, ME, KP, AJ	MA, M, KP, AJ	MA, M, KP, AJ, VC		KP	M
Verrucomicrobia (1)		V	V	V					
Unclassified (8)			U						

5.4 Discussion

5.4.1 Genomic variation in Ace Lake *Chlorobium* — potential phylotypes and ecotypes

The Ace Lake *Chlorobium* MAGs from different lake depths and seasons represented a single *Chlorobium* species, as was indicated by their IMG taxonomy and was evident from their identical *16S rRNA* genes and BclA proteins and ≥99.9% ANI. The presence of a single species of *Chlorobium* in Ace Lake has been previously reported based on metagenomic data from 2006 (Lauro et al, 2011). The genomic variation, in the form of

LCRs, identified in the *Chlorobium* MAGs suggested the presence of subpopulations that might represent phylotypes and ecotypes of Ace Lake *Chlorobium*. Some of the LCRs, especially those containing genes for metabolic functions and substrate transport, showed seasonal variation, with higher coverage (relative abundance) in summer compared to that in winter (Table 5.4). It is possible that some of the *Chlorobium* ecotypes were more prevalent in summer than in winter.

5.4.1.1 Variations potentially associated with cold adaptation

The low coverage of genes probably involved in cell wall modification indicated the existence of *Chlorobium* subpopulations that might differ in their cell surface structure. Multiple glycosyltransferase genes were identified in a LCR of AL_ref MAG, along with a phosphatidylinositol alpha-1,6-mannosyltransferase (a single copy gene), which also belonged to the glycosyltransferase family (Table 5.4). The phosphatidylinositol alpha-1,6-mannosyltransferase catalyses the transfer of mannose to phosphatidylinositol, and in *Mycobacterium tuberculosis*, this enzyme is essential for the production of structural components of cell wall (Boldrin et al, 2014). However, phosphatidylinositol is absent from the members of *Chlorobiaceae* family (Imhoff and Bias-Imhoff, 1995; Imhoff, 2014). Moreover, Ace Lake *Chlorobium* did not appear to have the capacity to synthesise phosphatidylinositol. Therefore, the function of phosphatidylinositol alpha-1,6-mannosyltransferase in Ace Lake *Chlorobium* was unclear, although it was present in more than 75% of the *Chlorobium* populations from each time period.

The glycosyltransferase genes present in a LCR of AL_ref MAG were auto-annotated as being involved in cell wall biosynthesis. Some of the Ace Lake *Chlorobium* genes involved in cell wall biosynthesis and modification, including glycosyltransferases, were shown to be distinct from the genes present in other members of *Chlorobiaceae* family, and it has been speculated that the presence of these genes in Ace Lake *Chlorobium* might be a means of cold adaptation (Ng et al, 2010). This is similar to the findings in *M. burtonii*, an archaeon found in the anoxic zone of Ace Lake. The genes involved in cell wall and membrane biosynthesis, mainly including glycosyltransferases, are overrepresented in *M. burtonii* genome and were speculated to be important for cold adaptation (Allen et al, 2009). At low temperatures, *M. burtonii* produces more extracellular polymeric substances, including polysaccharides, than at high temperatures (Reid et al, 2006). Multiple low coverage glycosyltransferase genes

(some containing SNPs) involved in cell wall biosynthesis were also present in Ace Lake *Synechococcus*, and were speculated to be involved in cell defence and immunity (Chapter 4 section 4.4.1.2). It is possible that the Ace Lake *Chlorobium* subpopulations containing these cell wall modification genes have a different cell wall structure that might help with adaptation to cold environment and/or cell immunity.

Apart from the genes involved in cell wall modification, a gene coding for DEAD/DEAH box helicase family protein was also present in a LCR of AL_ref MAG. This gene was identified from the manual reannotation of a gene initially annotated as a hypothetical protein. DEAD/DEAH-box helicases are generally involved in RNA-associated processes, and might contribute toward cell innate immunity and viral interactions as well as cell adaptation and response to stress, such as oxidative stress (Redder et al, 2015; Perčulija and Ouyang, 2019). Some of these RNA helicases like CsdA (*deaD*) and CrhC (*rhlE*) have also been speculated to be involved in cold adaptation in bacteria and archaea, such as *Escherichia coli*, *Anabaena* sp. strain PCC 7120, and *M. burtonii* (Jones et al, 1996; Chamot et al, 1999; Lim et al, 2000; Williams et al, 2011). The CsdA helicase has been speculated to be involved in destabilization of stable secondary structures in mRNA at low temperature, allowing for its translation to protein (Jones et al, 1996). The DEAD/DEAH box helicase family gene in AL_ref MAG LCR was truncated at the end, indicating incomplete assembly, and contained only the putative catalytic HKD family nuclease domain fused with a DEAD/DEAH box helicase domain. It was present in at least 25% of the *Chlorobium* populations from each time period. However, two other RNA helicase genes (*deaD*, *rhlE*) were present in all *Chlorobium* from each time period; both these genes have been speculated to be involved in cold adaptation (Jones et al, 1996; Chamot et al, 1999; Lim et al, 2000; Williams et al, 2011). The function of the low coverage DEAD/DEAH box helicase family gene was unclear, but it is likely that it was involved in cold adaptation.

5.4.1.2 Variations potentially associated with cell defence and immunity

Among the genes identified in the LCRs of AL_ref MAG, a few coded for genes involved in various bacterial defence systems indicating presence of *Chlorobium* subpopulations with varying capacity for cell defence and immunity. The defence genes in the LCRs coded for a type IV restriction endonuclease, type I restriction enzyme R and M subunits, and a BrnA antitoxin (Table 5.4). No SNPs were observed in any of these low coverage genes. The *brnA* gene in the LCRs was identified from the manual

reannotation of a gene initially annotated as an uncharacterized DUF4415 family protein. The *brnA* gene was present in ≤32% of the *Chlorobium* populations from each time period, however, its corresponding *brnT* toxin gene could not be identified among the annotated AL_ref MAG genes. In *Pseudomonas putida*, the *brnT* toxin gene is often disrupted, truncated, or lost from the *brnTA* operon, probably to reduce its toxic effects on the cell (Rosendahl et al, 2020).

The type I restriction enzymes (M and R subunits) identified in the LCRs were present in 45–65% of the *Chlorobium* populations from each time period. Two other type I restriction enzyme M subunits (both <100 aa long) were present in the non-LCR of AL_ref MAG, but their manual annotation showed that they contained only a portion of the N-terminal domain of a HsdM methyltransferase. This indicated that only a subpopulation of *Chlorobium* contained a type I R-M system. The type IV restriction endonuclease gene in the LCRs was manually annotated from a gene initially characterised as restriction system protein and was present in 25–35% of the *Chlorobium* populations from each time period. Another type IV restriction endonuclease gene was present in the non-LCRs of AL_ref MAG, indicating its presence in all *Chlorobium* from each time period. Type IV restriction enzymes target and restrict modified (usually methylated) DNA, and probably evolved in response to phage with the capacity to modify their DNA to evade host R-M systems (Loenen and Raleigh, 2014). Overall, the *Chlorobium* subpopulations containing these additional defence genes might have a better cell immunity and capacity for cell defence. See below section 5.4.3 for discussion on the defence capabilities of *Chlorobium*.

5.4.1.3 *Chlorobium* subpopulations containing specific substrate transporters

Various substrate transport genes were present in the LCRs of AL_ref MAG indicating that only subpopulations of Ace Lake *Chlorobium* had the capacity to transport certain substrates (Table 5.4). These genes coded for ABC transporters involved in the import of iron, cobalt, and vitamin B12 and export of proteases as well as N-ATPases involved in sodium ion export (Table 5.10). Four other genes coding for subunits of an iron ABC transporter and one gene coding for a cobalt/nickel transport system permease subunit CbiM were present in non-LCR of AL_ref MAG, indicating that all *Chlorobium* contained these genes. For analysis in this section, the substrate transport genes that colocalised on AL_ref MAG contigs were grouped together into Clusters 1–8 (Table 5.10). Some of these gene clusters probably represented gene operons, such as Clusters

6, 7, and 8 represented a cobalt transporter gene operon, a protease export system gene operon, and a N-ATPase gene operon, respectively.

Iron transporters

Iron is an essential trace element and an important component of the photosynthetic reaction centre of GSB. In bacteria, iron uptake can involve import of inorganic Fe^{3+}/Fe^{2+} ions or organic iron complexes like siderophores and hemes, using slightly different processes (Hogle et al, 2016). While inorganic iron is directly imported through the inner membrane by ABC transporters containing substrate-binding, permease, and ATP-binding subunits, the organic forms first need to be imported through the outer membrane by a TonB-dependent transporter (Hogle et al, 2016). Among the substrate transport genes in the LCRs of AL_ref MAG, the genes in Clusters 1 and 2 (Table 5.10) were probably associated with the uptake of both inorganic and organic forms of iron, considering that one of the clusters contained part of a TonB-dependent transporter gene; the two clusters were only three genes apart on the same contig. On average, Clusters 1 and 2 were present in 61% and 65% of the *Chlorobium* populations from all time periods, respectively, indicating that more than half the population had an added capacity to acquire iron from its surrounding environment (Table 5.10). The concentration of iron in Ace Lake increases with depth, being highest in the bottom-most anoxic waters (~3 μM), where its concentration is nearly four-times the concentration in the upper oxic zone (0.7 μM) (Masuda et al, 1988; Rankin et al, 1999). Based on these concentration values, the iron levels at the Ace Lake oxycline would be around 1.2–1.3 μM, which is only one-third of the maximum iron concentration in the lake (Figure 5.9a). Therefore, the presence of additional iron transporters that allow uptake of both inorganic and organic forms of iron would probably be advantageous to *Chlorobium*. The abundance of the *Chlorobium* subpopulation containing these additional iron transporter genes was slightly higher in summer than in winter, which might suggest that this subpopulation was more competitive in summer (Table 5.10).

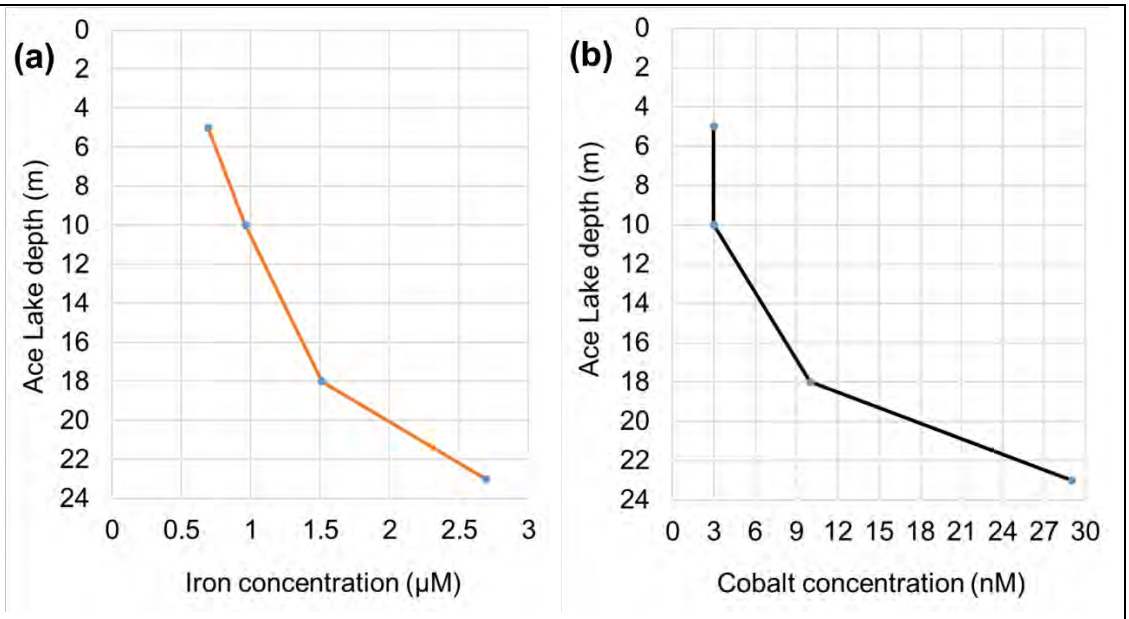


Figure 5.9 Iron and cobalt concentrations in Ace Lake. The iron and cobalt concentrations in the figure were taken from the data published by Rankin et al (1999) and originally measured by Masuda et al (1988). The Ace Lake oxycline samples were from depths ranging from 12.7–14.5 m, which would put the iron and cobalt concentrations at the Ace Lake oxycline between 1.15–1.25 μM and 5–7 nM, respectively.

Vitamin B12 transporters

Vitamin B12 (cobalamin) acts as a cofactor for various metabolic functions, and although most bacteria contain cobamide-dependent enzymes, not all are capable of synthesizing it (Shelton et al, 2019). Such bacteria need to acquire cobalamin from their surrounding environment. Similar to siderophore and heme, the uptake of vitamin B12 first requires transport through the outer membrane using the outer membrane transporter BtuB, which is a TonB-dependent transporter (Pieńko and Trylska, 2020). The transport of vitamin B12 through the inner membrane requires ABC transporters such as BtuCDF or energy-coupling factor (ECF) CbrT (Cadieux et al, 2002; Santos et al, 2018). Some of the LCRs of AL_ref MAG contained genes that coded for a BtuB outer membrane transporter and a BtuC inner membrane permease protein along with a substrate- and an ATP-binding subunit potentially involved in vitamin B12 transport. These potential vitamin B12 transport genes belonged to Clusters 3–5, of which Clusters 4 and 5 were only three genes apart on the same contig (Table 5.10). On average, Clusters 3, 4, and 5 were present in 16%, 18%, and 12% of the *Chlorobium* populations from all time periods, respectively, indicating that a very small *Chlorobium* population had the capacity to acquire vitamin B12 from its surrounding environment

(Table 5.10). Interestingly, the abundance of the *Chlorobium* subpopulation containing vitamin B12 transport genes was nearly double or triple in summer than in winter, indicating their prevalence in summer (Table 5.10). Some *Chlorobium* subpopulations also had genes associated with cobalamin biosynthesis, whereas some had a cobinamide salvaging gene colocalised with the vitamin B12 transport genes. See below section 5.4.1.4 for further discussion.

Cobalt transporters

Cobalt is an essential micronutrient and is a major component of the corrin rings of some coenzymes such as cobalamin and its derivatives. Similar to inorganic iron uptake, the uptake of cobalt is performed by high affinity ABC transporters in bacteria such as the CbiMNQO transport system, which is an ECF-type ABC transporter (Cheng et al, 2011). This transport system consists of an ATP-binding protein (CbiO), a transmembrane protein (CbiQ), a substrate-binding permease protein (CbiM), and an additional small transmembrane protein (CbiN), which together work to import cobalt and nickel ions, preferably copper (Rodionov et al, 2006; Cheng et al, 2011; Kirsch and Eitinger, 2014). In Ace Lake *Chlorobium*, the *cbiMNQO* operon potentially involved in cobalt transport was identified in LCR, but the *cbiM* gene was truncated (Cluster 6 in Table 5.10). This gene was placed at one end of the contig and was missing the beginning half of its sequence, which probably resulted from incomplete assembly rather than gene disruption. A longer sequence of the *cbiM* gene was located at the end of another contig, in a non-LCR of AL_ref MAG, indicating that all *Chlorobium* from each time period had this gene. The *cbiMNQO* operon was colocalised with some genes involved in the anaerobic pathway for cobalamin biosynthesis in *Chlorobium*, which suggested that this transport system was probably used for cobalt uptake and not nickel uptake (section 5.4.1.4). On average, Cluster 6 was present in 47% of the *Chlorobium* populations from all time periods, indicating that nearly half of the population had the capacity for cobalt uptake from surrounding lake waters (Table 5.10). The concentration of cobalt in Ace Lake Upper oxic zone (3 nM) is almost one order of magnitude lower than that in the lowest anoxic zone (29 nM) of the lake, but 150-times the cobalt levels in sea water (Masuda et al, 1988; Rankin et al, 1999). Based on these concentration values, the cobalt concentration at the Ace Lake oxycline would be around 5–7 nM, which is much lower than the maximum concentration of cobalt in the lake (Figure 5.9b). Therefore, the presence of cobalt transporters would probably be advantageous to

Chlorobium, especially to the subpopulation also capable of cobalamin biosynthesis (section 5.4.1.4). Notably, the abundance of the *Chlorobium* subpopulation containing cobalt transporter genes was nearly double in summer than in winter, except for the truncated *cbiM* gene that had similar abundance in summer and winter (Table 5.10).

Protease transporters

Of the three Ace Lake *Chlorobium* proteins coded by the protease export system operon, two matched AprD and AprE proteins of the AprDEF protease export system in *Pseudomonas aeruginosa* (Cluster 7 in Table 5.10). The third protein was related to a TolC outer membrane protein and represented part of the AprF protein sequence. On average, Cluster 7 was present in 33% of the *Chlorobium* populations from all time periods and its abundance was similar in summer and winter (Table 5.10). In *P. aeruginosa*, the protease export system is used to secrete an alkaline protease (AprA), which breaks down laminins and is involved in bacterial virulence (Heck et al, 1986; Laarman et al, 2012). However, in Ace Lake *Chlorobium*, it was unclear which protease was associated with the export system.

N-ATPases

N-ATPases are sodium-transporting adenosine triphosphatases that are similar to F_0F_1-ATPases (F-ATPases) present in bacterial plasma membranes. They are coded by a highly conserved operon (*atpDCQRBEFAG*), which contains genes similar to those in the F-ATPase operon (*atpIBEFHAGDC*) (Dibrova et al, 2010; Schulz et al, 2017). However, the two ATPases are distinct: (i) N-ATPases do not contain the *atpH* and *atpI* genes, which code for F-ATPase delta and I subunits, respectively; (ii) N-ATPases contain *atpR* and *atpQ* genes, which are not present in F-ATPases; and (iii) unlike F-ATPases that use a proton (H^+) gradient for ATP production, N-ATPases utilise ATP to actively transport Na^+ or H^+ ions out of the bacterial cell (Von Ballmoos et al, 2008; Dibrova et al, 2010; Schulz et al, 2017). Many of the microbes that contain N-ATPases are from saline environments, and the N-ATPase operon is always accompanied by the F-ATPase operon in these organisms (Dibrova et al, 2010). This is similar to what was observed in *Chlorobium* as well as *Synechococcus* from Ace Lake, both of which contained a F-ATPase operon apart from the N-ATPase operon in their genomes. N-ATPases are capable of translocating both Na^+ and H^+ ions and the sequence composition of their ATPase subunit c (coded by *atpE*) decides which ion will be

translocated (Von Ballmoos et al, 2008; Dibrova et al, 2010; Schulz et al, 2017). As the ATPase subunit c in Ace Lake *Chlorobium* had the two Na^+-binding glutamate residues in both its C-and N-terminal helices, it is likely that the *Chlorobium* N-ATPases are involved in Na^+ ion export, which would be beneficial to these bacteria living in a saline environment. On average, the N-ATPase operon was present in 68% of the *Chlorobium* populations from all time periods, indicating that more than half of the population had the capacity to actively export Na^+ ions (Cluster 8 in Table 5.10). Moreover, the abundance of the *Chlorobium* subpopulation containing the N-ATPase operon was slightly higher in summer (76%) than in winter (61–66%).

Table 5.10 Ace Lake *Chlorobium* low coverage genes associated with substrate transport.
[A]The percentages indicate the average of relative coverages of the gene clusters in all metagenomes (section 5.2.3.3). The contig number of the AL_ref MAG contigs on which the genes were identified are also provided (Table 5.3). [B] The seasons mentioned in the second column refer to seasons from which the Ace Lake Interface samples were collected — summer (S), Dec 2014; winter (W), Jul 2014 and Aug 2014; spring (Sp), Nov 2008, Nov 2013, and Oct 2014 (Table 5.1). The percentages shown are average of coverage values from metagenomes from a season calculated across the gene length (section 5.2.3.3). [C] The initial annotation of AL_ref MAG genes was performed by JGI's IMG system. [D] The gene functions were verified against reference proteins in the UniProtKB/Swiss-Prot database. The proteins with low alignment or no hits to the UniProtKB/Swiss-Prot database proteins were realigned to the reference proteins in the UniProtKB or RefSeq protein databases.

Cluster number and coverage (AL_ref MAG contig)[A]	Seasons and % *Chlorobium* subpopulation in which observed[B]	Gene annotation[C]	Gene function and protein sequence identity[D]
Cluster 1 61% (C7)	**S:** 48% **W:** 44–49% **Sp:** 45–47%; 70% in Oct 2014	Iron complex outer membrane receptor protein/hemoglobin/transferrin /lactoferrin receptor protein/vitamin B12 transporter	86% TonB-dependent receptor *Chlorobium limicola* (RefSeq)
	S: 74% **W:** 67–69%	Iron complex transport system substrate-binding protein	68% ABC transporter substrate-binding protein (metal-binding TroA-like

	Sp: 67–72%; 83% in Oct 2014		domain) *Chlorobium limicola* (RefSeq)
Cluster 2 65% (C7)	**S:** 78% **W:** 63–64% **Sp:** 63–68%	Iron complex transport system permease protein	65% Iron ABC transporter permease *Prosthecochloris aestuarii* (RefSeq)
	S: 72% **W:** 60–62% **Sp:** 44–71%	Iron complex transport system ATP-binding protein	42% Uncharacterized ABC transporter ATP-binding protein HI_1272 *Haemophilus influenzae* Rd KW20
	S: 77% **W:** 63–65% **Sp:** 64–71%	Iron complex transport system substrate-binding protein	23% Fe(3+)-citrate-binding protein YfmC *Bacillus subtilis* subsp. *subtilis* str. 168
Cluster 3 16% (C26)	**S:** 27% **W:** 10% **Sp:** 11–24%	Iron complex transport system substrate-binding protein	25% Uncharacterized lipoprotein MJ0878 (containing Fe/B12 periplasmic-binding domain) *Methanocaldococcus jannaschii*
	S: 27% **W:** 10% **Sp:** 11–24%	Iron complex transport system permease protein	35% Vitamin B12 import system permease protein BtuC *Klebsiella pneumoniae*
	S: 23% **W:** 8–10% **Sp:** 10–23%	Iron complex transport system ATP-binding protein	37% Uncharacterized ABC transporter ATP-binding protein MJ0873 (ABC-type cobalamin/Fe-siderophore transporter) *Methanocaldococcus jannaschii*
Cluster 4 18% (C27)	**S:** 25% **W:** 9–10% **Sp:** 11–29%	Iron complex outermembrane receptor protein	24% Vitamin B12 transporter BtuB *Salmonella typhimurium*

Cluster 5 12% (C27)	**S:** 21% **W:** 7–9% **Sp:** 8–19%	Iron complex outermembrane receptor protein/hemoglobin/transferrin /lactoferrin receptor protein/vitamin B12 transporter	52% TonB-dependent receptor *Prosthecochloris* sp. GSB1 (RefSeq)
Cluster 6 47% (C23)	**S:** 57% **W:** 31–32% **Sp:** 30–51%	Cobalt/nickel transport system ATP-binding protein	45% Cobalt import ATP-binding protein CbiO *Rhodobacter capsulatus*
	S: 61% **W:** 33–36% **Sp:** 34–51%	Cobalt/nickel transport system permease protein	31% Cobalt transport protein CbiQ *Rhodobacter capsulatus*
	S: 71% **W:** 45–47% **Sp:** 38–59%	Cobalt/nickel transport protein	52% Cobalt transport protein CbiN *Nostoc* sp.
	S: 53% **W:** 54–59% **Sp:** 34–53%	Cobalt/nickel transport system permease protein	Small sequence matches to cobalt transporter CbiM
Cluster 7 33% (C24)	**S:** 37% **W:** 31–34% **Sp:** 28–38%	Protease secretion system outer membrane protein	28% Outer membrane protein TolC *Vibrio cholerae*
	S: 38% **W:** 34–38% **Sp:** 30–32%	Protease secretion system membrane fusion protein	32% Alkaline protease secretion protein AprE *Pseudomonas aeruginosa* PAO1
	S: 36% **W:** 35–42% **Sp:** 26–31%	ATP-binding cassette subfamily C exporter for protease/lipase	46% Alkaline protease secretion ATP-binding protein AprD *Pseudomonas aeruginosa* PAO1
Cluster 8 68% (C11)	**S:** 58% **W:** 46–55% **Sp:** 52–56%	F-type H+-transporting ATPase subunit gamma	24% ATP synthase gamma chain (AtpG) *Natranaerobius thermophilus*
	S: 60% **W:** 50–52% **Sp:** 55–57%	F-type H+-transporting ATPase subunit alpha	78% ATP synthase subunit alpha 1 (AtpA) *Pelodictyon luteolum* DSM 273

	S: 80% **W:** 63–67% **Sp:** 65–74%	F-type H+-transporting ATPase subunit b	50% ATP synthase subunit b 1 (AtpF) *Pelodictyon luteolum* DSM 273
	S: 78% **W:** 66–69% **Sp:** 65–72%	F-type H+-transporting ATPase subunit c	52% ATP synthase subunit c 1 (AtpE) *Pelobacter carbinolicus* DSM 2380
	S: 77% **W:** 54–58% **Sp:** 53–74%	F-type H+-transporting ATPase subunit a	88% ATP synthase subunit a 1 (AtpB) *Pelodictyon luteolum* DSM 273
	S: 76% **W:** 65–76% **Sp:** 66–78%	F1-F0 ATPase (N-ATPase) AtpR subunit	49% ATP synthase subunit I (AtpR) *Polaribacter* sp. IC073 (RefSeq)
	S: 84% **W:** 64–77% **Sp:** 71–82%	ATP synthase protein I	55% AtpZ/AtpI family protein (AtpQ) *Chlorobaculum parvum* (RefSeq)
	S: 87% **W:** 69–73% **Sp:** 80–87%	F-type H+-transporting ATPase subunit epsilon	84% F0F1 ATP synthase subunit epsilon (AtpC) *Chlorobium* sp. N1 (UniProtKB)
	S: 84% **W:** 70–72% **Sp:** 75–81%	F-type H+-transporting ATPase subunit beta	81% ATP synthase subunit beta 2 (AtpD) *Pelodictyon luteolum* DSM 273

5.4.1.4 *Chlorobium* subpopulations capable of cobalamin biosynthesis and cobinamide salvaging

Cobalamin is an organometallic compound containing a central corrin ring with chelated cobalt that is synthesised by certain bacteria and archaea. It is used as cofactor in many metabolic reactions. In bacteria and archaea, there are two main pathways for the synthesis of adenosylcobalamin (a biologically active form of cobalamin) — the aerobic and anaerobic pathways (Figure 5.10). The main difference between the aerobic and anaerobic pathways for cobalamin biosynthesis, apart from the oxygen requirement, is the timing of cobalt insertion — in the anaerobic pathway cobalt is added to the corrin ring much earlier than in the aerobic pathway (Roth et al, 1993; Roessner et al, 2002; Frank et al, 2005; Heldt et al, 2005). The two pathways also differ in their use of

substrates and genes involved in the production of the lower axial ligand (5,6-dimethylbenzimidazole, DMB) of adenosylcobalamin — the anaerobic pathway uses 5-amino-1-(5-phospho-β-D-ribosyl)imidazole as a substrate and the *bzaABCDE* operon, whereas the aerobic pathway uses riboflavin as a substrate and the *bluB* gene (Taga et al, 2007; Hazra et al, 2015). Cyanobacteria, including many members of *Synechococcus* genus, do not have the ability to produce DMB, and so instead of adenosylcobalamin they synthesise pseudocobalamin, which contains adenine in place of DMB as its lower axial ligand (Helliwell et al, 2016; Heal et al, 2017). Apart from the ability to synthesise adenosylcobalamin, bacteria and archaea can salvage cobinamides, precursors of adenosylcobalamin, and convert them to intermediates of the cobalamin biosynthesis pathways (Figure 5.10).

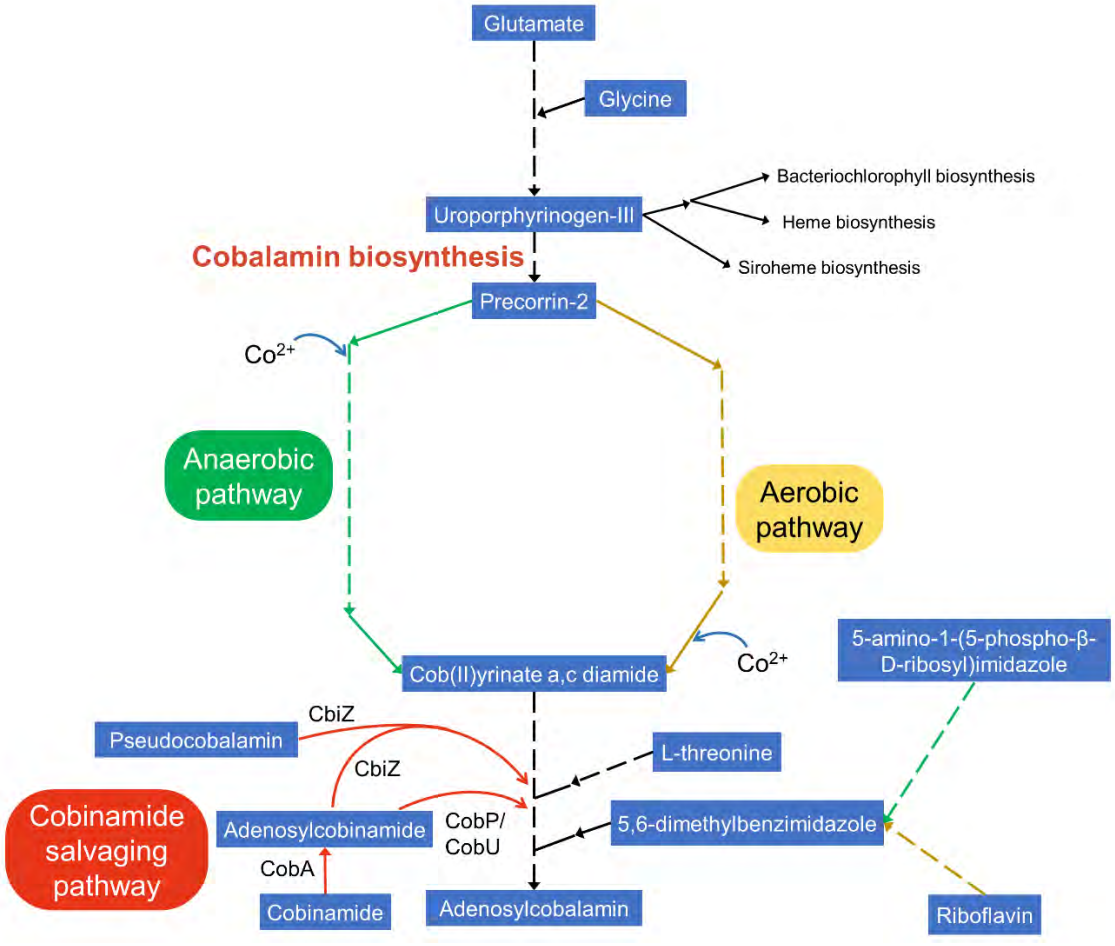


Figure 5.10 Cobalamin biosynthesis and cobinamide salvaging pathways. The schematic shows the various pathways involved in the biosynthesis of adenosylcobalamin and cobinamide salvaging. Cobalamin biosynthesis begins with glutamate, whereas glycine is the starting substrate for bacteriochlorophyll and heme biosynthesis. The three biosynthesis pathways part after formation of uroporphyrinogen-III. The aerobic (yellow branch) and anaerobic (green

branch) pathways of cobalamin biosynthesis share common reactions before the formation of precorrin-2 and after the formation of cob(II)yrinate a,c-diamide. The two pathways differ in the timing of cobalt insertion into the corrin ring and the source of 5,6-dimethylbenzimidazole, the lower axial ligand of adenosylcobalamin. The cobinamide salvaging pathway (red branch) involves the conversion of cobinamide to intermediates of the cobalamin biosynthesis pathway using CobP/CobU and/or CbiZ. Of these two enzymes, CbiZ is capable of remodelling pseudocobalamin into intermediates of cobalamin biosynthesis pathway (Gray and Escalante-Semerena, 2009). The dashed arrows connecting the intermediate substrates indicate multi-step processes. The pathway information for this schematic was taken from BioCyc online service (https://biocyc.org/) as well as the data published by Taga et al (2007), Gray et al (2008), Gray and Escalante-Semerena (2009), and Hazra et al (2015). CobA, corrinoid adenosyltransferase; CobP/CobU, adenosylcobinamide kinase/adenosylcobinamide-phosphate guanylyltransferase; CbiZ, adenosylcobinamide amidohydrolase.

Adenosylcobalamin production through anaerobic pathway

A comparative genomics study of bacterial potential for cobalamin biosynthesis and utilisation has shown that most bacteria rely on cobalamin, but cannot synthesise it (Shelton et al, 2019). The *Chlorobi* members analysed in the study, including C-phaeov, did not have genes for cobalamin biosynthesis, but many *Synechococcus* species had the ability to produce cobalamin via the anaerobic pathway (Shelton et al, 2019). The genes for anaerobic pathway for cobalamin biosynthesis have also been identified in Ace Lake microbes such as the archaea *M. burtonii* (Allen et al, 2009). Interestingly, the Ace Lake *Chlorobium* contained genes associated with the anaerobic pathway for cobalamin biosynthesis, whereas the Ace Lake *Synechococcus* contained cobalamin biosynthesis genes associated with the aerobic pathway. However, in *Chlorobium*, the cobalamin biosynthesis genes exclusive to the anaerobic pathway (green branch between precorrin-2 to cob(II)yrinate a,c-diamide in Figure 5.10) were located in a LCR of AL_ref MAG (Table 5.11). On average, the cobalamin biosynthesis operon was present in 38% of the *Chlorobium* populations from all time periods, indicating that less than half of the population had the capacity to synthesize adenosylcobalamin. Moreover, the abundance of the *Chlorobium* subpopulation carrying this operon was almost twice in summer than in winter, indicating their prevalence in summer (Table 5.11).

An analysis of the annotated genes on *Chlorobium* MAGs showed that Ace Lake *Chlorobium* did not contain the genes required for DMB production (*bzaABCDE* or *bluB*) and the final step in adenosylcobalamin production (*cobC*). However, it did

contain the DMB activation and utilisation genes (*cobT*, *cobS*), indicating its capacity to use DMB for cobalamin biosynthesis. Some organisms have the ability to remodel exogenous DMB to produce cobalamin (Anderson et al, 2008; Helliwell et al, 2016). It is likely that Ace Lake *Chlorobium* also acquires DMB from its surrounding environment and uses it to produce adenosylcobalamin rather than pseudocobalamin. The absence of *cobC* gene has been reported in many cobalamin-producing bacteria, including all *Actinobacteria* and some *Alphaproteobacteria*, and it is considered to be replaced by hypothetical protein-coding genes *chlZ* or *cbiXY*, respectively (Rodionov et al, 2003). It is worth noting that the Ace Lake *Chlorobium* MAGs are draft genomes (99.5% genome completeness), and it is possible that *bzaABCDE*, *bluB*, and/or *cobC* genes were not identified in Ace Lake *Chlorobium* because they were part of the 0.5% of the genome that could not be assembled, rather than being absent from the genomes.

The Ace Lake *Chlorobium* also contained a colocalised cluster of genes coding for cobalt/magnesium chelatases in the LCRs (Table 5.11). One of these genes coded for the cobaltochelatase subunit N, but the genes coding for subunits S and T were not identified in *Chlorobium*. The other three chelatase genes coded for putative magnesium chelatase subunits BchH, BchI, BchD. Cobaltochelatase subunits NST form a complex that catalyses the insertion of cobalt into hydrogenobyrinic acid a,c-diamide in the aerobic pathway for cobalamin biosynthesis (Crouzet et al, 1991; Debussche et al, 1992). On the other hand, magnesium chelatase subunits HID are involved in the magnesium insertion step of bacteriochlorophyll biosynthesis. Existing homology between cobaltochelatase subunits N, S, and T and magnesium chelatase subunits H, I, and D, respectively, has been shown (Petersen et al, 1998; Willows et al, 2001). It has also been speculated that certain bacteria that do not contain genes for cobaltochelatase subunits S and T instead use magnesium chelatase subunits I and D to form the cobaltochelatase complex (Rodionov et al, 2003). Interestingly, these cobalt/magnesium chelatase genes were colocalised with potential vitamin B12 transport genes (Clusters 4 and 5 in Table 5.10), which might indicate their relevance to cobalamin biosynthesis. A similar gene cluster configuration, with cobalt/magnesium chelatases placed next to Ton-B dependent receptor protein for vitamin B12, were identified in the genome of *Cb. tepidum*, and it was speculated that these chelatases might be involved in incorporating cobalt into exogenously-acquired vitamin B12 (Eisen et al, 2002).

The genes coding for cobalamin biosynthesis have been previously reported to be colocalised with the cobalt transporter operon *cbiMNQO* (Rodionov et al, 2003; Rodionov et al, 2006). In accordance with this, the Ace Lake *Chlorobium* genes involved in anaerobic pathway for cobalamin biosynthesis (Table 5.11) were identified next to the genes coding for a cobalt transporter (Cluster 6 in Table 5.10). As discussed earlier, the presence of cobalt transporters could help *Chlorobium* in acquiring cobalt from its surrounding waters in the Ace Lake oxycline (section 5.4.1.3). As cobalt is the major component of the central corrin ring of cobalamin and considering that cobalt transporter genes colocalised with cobalamin biosynthesis genes, it is likely that the imported cobalt is specifically used to synthesise cobalamin.

Cobinamide and pseudocobalamin salvaging by Chlorobium and its potential interaction with Synechococcus

Apart from cobalamin biosynthesis, a small population of *Chlorobium* (on average 15% from all time periods) had an added capacity to salvage cobinamide, a precursor of adenosylcobalamin. Generally, bacteria use CobA and CobP/CobU enzymes for salvaging cobinamide, whereas archaea use CobA and CbiZ (Woodson et al, 2003; Woodson and Escalante-Semerena, 2004; Gray et al, 2008; Gray and Escalante-Semerena, 2009). However, some bacteria, including a few *Chlorobium* species, also use CbiZ for salvaging cobinamide and pseudocobalamin (Gray et al, 2008; Gray and Escalante-Semerena, 2009). The Ace Lake *Chlorobium* had genes for all three enzymes, but only *cbiZ* was located in a LCR of AL_ref MAG. These findings suggested that probably all Ace Lake *Chlorobium* had the ability to salvage cobinamide, but a subpopulation had an added capacity to convert cobinamide as well as pseudocobalamin to intermediates of cobalamin biosynthesis pathway. Moreover, the abundance of the *Chlorobium* subpopulation containing the *cbiZ* gene varied with season, summer abundance being twice the winter abundance (Table 5.11).

Many bacterial *cbiZ* genes, including those from some *Chlorobium* species, are usually colocalised with the genes coding for vitamin B12 transporters (Gray et al, 2008). Similar to this report, the *cbiZ* gene of Ace Lake *Chlorobium* was identified next to the genes probably involved in vitamin B12 transport (Cluster 3 in Table 5.10). Studies on *Rhodobacter sphaeroides*, a purple bacterium usually found in freshwater environments, suggested that the bacterium probably uses the vitamin B12 transporters to import pseudocobalamin produced by cyanobacteria in its environment and uses CbiZ to

salvage it (Watanabe et al, 1999; Miyamoto et al, 2006; Watanabe et al, 2006; Watanabe et al, 2007; Gray et al, 2008; Gray and Escalante-Semerena, 2009). As Ace Lake contained a large population of *Synechococcus* and the *Chlorobium cbiZ* gene was colocalised with potential vitamin B12 transporters, it is possible that a similar interaction between these two key players occurs in the lake — where *Synechococcus*-produced pseudocobalamin is salvaged by *Chlorobium* for cobalamin biosynthesis.

Potential benefits of adenosylcobalamin production and cobinamide salvaging

Overall, the genes involved in cobalt uptake (*cbiMNQO*), adenosylcobalamin production (anaerobic pathway genes), uptake of cobalamin precursors (vitamin B12 transporter genes), cobalt/magnesium chelatases (*cobN*, *bchHID*), and cobinamide and pseudocobalamin salvaging (*cbiZ*) had low coverage in Ace Lake *Chlorobium*. Of these, the genes for cobalamin biosynthesis were colocalised with the cobalt transporter operon, highlighting the potential association between uptake of cobalt and its use for cobalamin production in *Chlorobium*. Moreover, the clustering of potential vitamin B12 transporter genes and the gene for cobinamide and pseudocobalamin salvaging (*cbiZ*) indicated that *Chlorobium* might be capable of salvaging exogenous cobinamides and pseudocobalamin to form cobalamin; *Synechococcus* being the most probable source of pseudocobalamin. Similarly, the presence of a potential vitamin B12 transporter gene next to the cobalt/magnesium chelatase genes might indicate their role in inserting cobalt into vitamin B12 acquired from the surrounding environment.

In cultivated isolates of two *Chlorobium* species, it has been shown that cobalamin deficiency can lead to reduced bacteriochlorophyll content and affect chlorosome formation (Fuhrmann et al, 1993). Furthermore, treating cultivated vitamin B12-deficient microbes with added vitamin B12 increases their bacteriochlorophyll content (Sato et al, 1981; Fuhrmann et al, 1993). Considering its capacity to synthesise cobalamin and salvage cobalamin precursors and the presence of genes associated with cobalt transporters and potential vitamin B12 transporters, it is possible that the Ace Lake *Chlorobium* relies on cobalamin production for maintaining bacteriochlorophyll production and chlorosome formation, both of which would affect its photosynthetic capacity. The ability to improve its bacteriochlorophyll content could help the Ace Lake *Chlorobium* in recovering from the effects of being in the dark for prolonged periods in winter and to reach high abundance levels in summer. This is supported by the higher summer abundance (twice the winter abundance) of the *Chlorobium* subpopulations

containing genes for cobalamin biosynthesis, cobinamide and pseudocobalamin salvaging, cobalt transporter, and/or vitamin B12 transporters. It is likely that in the presence of sufficient light in summer, the *Chlorobium* capable of producing cobalamin are able to reach a higher abundance due to their improved photosynthetic capacity.

Table 5.11 Ace Lake *Chlorobium* low coverage genes associated with cobalamin biosynthesis and cobinamide and pseudocobalamin salvaging. [A] The seasons mentioned in the first column refer to seasons from which the Ace Lake Interface samples were collected — summer (S), Dec 2014; winter (W), Jul 2014 and Aug 2014; spring (Sp), Nov 2008, Nov 2013, and Oct 2014 (Table 5.1). The percentages shown are average of relative coverages of genes or operons in metagenomes from each season (section 5.2.3.3). The contig numbers of the AL_ref MAG contigs on which the genes or operons were identified are also provided (Table 5.3). [B] The initial annotation of the genes was performed by JGI's IMG system. [C] The gene functions were verified against reference proteins in the UniProtKB/Swiss-Prot database. Some of the genes coded for bifunctional proteins, which is indicated by their gene names, i.e., *cbiFG*, *cbiET*, and *cbiHC*. The cobalamin biosynthesis operon was identified on C23 contig of AL_ref MAG alongside the cobalt transporter operon, whereas the cobinamide salvaging gene was colocalised with potential cobalamin transporter operon on C26 contig. The cobalt chelatase gene and magnesium chelatase operon were also clustered with two cobalamin transporter genes on C27 contig, with one transporter gene between them and one flanking the magnesium chelatase operon.

Gene or operon; Season: *Chlorobium* subpopulation (AL_ref MAG contig)[A]	Gene	Gene annotation[B]	Gene function and protein sequence identity[C]
Cobalamin biosynthesis (anaerobic pathway) genes; **S:** 58% **W:** 28–29% **Sp:** 29–44% (C23)	*cbiD*	Cobalt-precorrin-5B (C1)-methyltransferase	51% Cobalt-precorrin-5B C(1)-methyltransferase *Prosthecochloris aestuarii*
	cbiJ	Cobalt-precorrin-5B (C1)-methyltransferase	30% Cobalt-precorrin-6A reductase *Methanothermobacter thermautotrophicus*
	cbiFG	Precorrin-4 methylase/cobalamin biosynthesis protein CbiG	49% Cobalt-precorrin-4 C(11)-methyltransferase CbiF *Methanocaldococcus jannaschii*

			31% Cobalt-precorrin-5A hydrolase CbiG *Salmonella typhimurium*
	cbiET	Precorrin-6Y C5,15-methyltransferase (decarboxylating)	32% Cobalamin biosynthesis bifunctional protein CbiET *Bacillus megaterium*
	cbiHC	Precorrin-3B methylase/precorrin isomerase	49% Cobalt-factor III methyltransferase CbiHC *Bacillus megaterium*
	cbiL	Precorrin-2/cobalt-factor-2 C20-methyltransferase	28% Precorrin-2 C(20)-methyltransferase *Pseudomonas aeruginosa*
	cbiK	Sirohydrochlorin cobaltochelatase	26% Sirohydrochlorin cobaltochelatase CbiKP *Desulfovibrio vulgaris*
	cysG	Uroporphyrin-III C-methyltransferase	44% Uroporphyrinogen-III C-methyltransferase *Bacillus megaterium*
Cobinamide and pseudocobalamin salvage gene; **S:** 25% **W:** 8–10% **Sp:** 10–20% (C26)	*cbiZ*	Adenosylcobinamide amidohydrolase	33% Uncharacterized protein MJ1613 (containing CbiZ domain) *Methanocaldococcus jannaschii* DSM 2661
Cobalt chelatase gene; **S:** 25% **W:** 9% **Sp:** 11–22% (C27)	*cobN*	Cobaltochelatase CobN	34% Aerobic cobaltochelatase subunit CobN *Sinorhizobium* sp.
Magnesium chelatase genes; **S:** 26% **W:** 9% **Sp:** 10–22%	*bchD*	Magnesium chelatase subunit D	28% Magnesium-chelatase subunit D *Rhodobacter capsulatus* SB 1003
	bchI	Magnesium chelatase subunit I	55% Magnesium-chelatase subunit I homolog *Synechocystis* sp. PCC 6803

(C27)	*bchH*	Cobaltochelatase CobN	28% Magnesium-chelatase subunit H *Rhodobacter capsulatus*

5.4.2 *Chlorobium* endemism in the Vestfold Hills

The taxonomic and abundance analyses of Ace Lake, Ellis Fjord, and Taynaya Bay metagenomes showed that all three systems contained *Chlorobium* closely related to C-phaeov (Figure 5.3). The presence of members of *Chlorobiaceae* family in all three systems has been shown before (Burke and Burton, 1988a; Ng et al, 2010; Lauro et al, 2011). Interestingly, despite the distance between Ace Lake, Ellis Fjord, and Taynaya Bay, the three systems harboured the same species of *Chlorobium*, as indicated by their identical *16S rRNA* genes and BclA proteins, and ≥99.9% ANI (Figures 5.4 and 5.5). Moreover, C-phaeov was the closest related GSB to the Vestfold Hills *Chlorobium* but the two species were distinct, which was evident from their 99% *16S rRNA* gene identity, 98% BclA protein identity, 85% ANI, 89% AAI, and some differences in their metabolic function and defence capacities (Figures 5.4, 5.5, and 5.7; section 5.3.4.3).

5.4.2.1 *Chlorobium* phylotypes and ecotypes in Ace Lake, Ellis Fjord, and Taynaya Bay

The FR analysis of EF_ref MAG showed that some *Chlorobium* gene clusters and operons had low coverage in metagenomes from Ace Lake, Ellis Fjord, and Taynaya Bay, indicating the presence of similar *Chlorobium* phylotypes and ecotypes in all three systems (the *Chlorobium* phylotypes and ecotypes are described in section 5.4.1). These gene clusters and operons included genes involved in cell defence, substrate transport, cell wall modification, and some metabolic functions, and their coverages were different in all three systems (Figure 5.11a). This suggested that although Ace Lake, Ellis Fjord, and Taynaya Bay contained the same species of *Chlorobium* and probably similar *Chlorobium* subpopulations, the abundances of the *Chlorobium* phylotypes and ecotypes varied in the three systems (Figure 5.11a). A cluster analysis of the *Chlorobium* subpopulations from the three systems was performed based on the relative coverages of the genes identified in LCRs of EF_ref MAG. The clustering showed that the *Chlorobium* subpopulations from Ellis Fjord and Ace Lake had a more similar abundance pattern than those from Taynaya Bay (Figure 5.11b). Notably, the metagenomic data for Ace Lake and Ellis Fjord was from biomass collected on large format filters (biomass size range 0.1–20 μm), whereas Taynaya Bay data was from

biomass collected on Sterivex cartridges (biomass size range 0.22–20 µm), which might have contributed toward the distinct abundance pattern of Taynaya Bay *Chlorobium* subpopulations. However, the relative coverages of genes or operons used for cluster analysis should be comparable, as they were normalised to the mean read depths of EF_ref MAG in each metagenome.

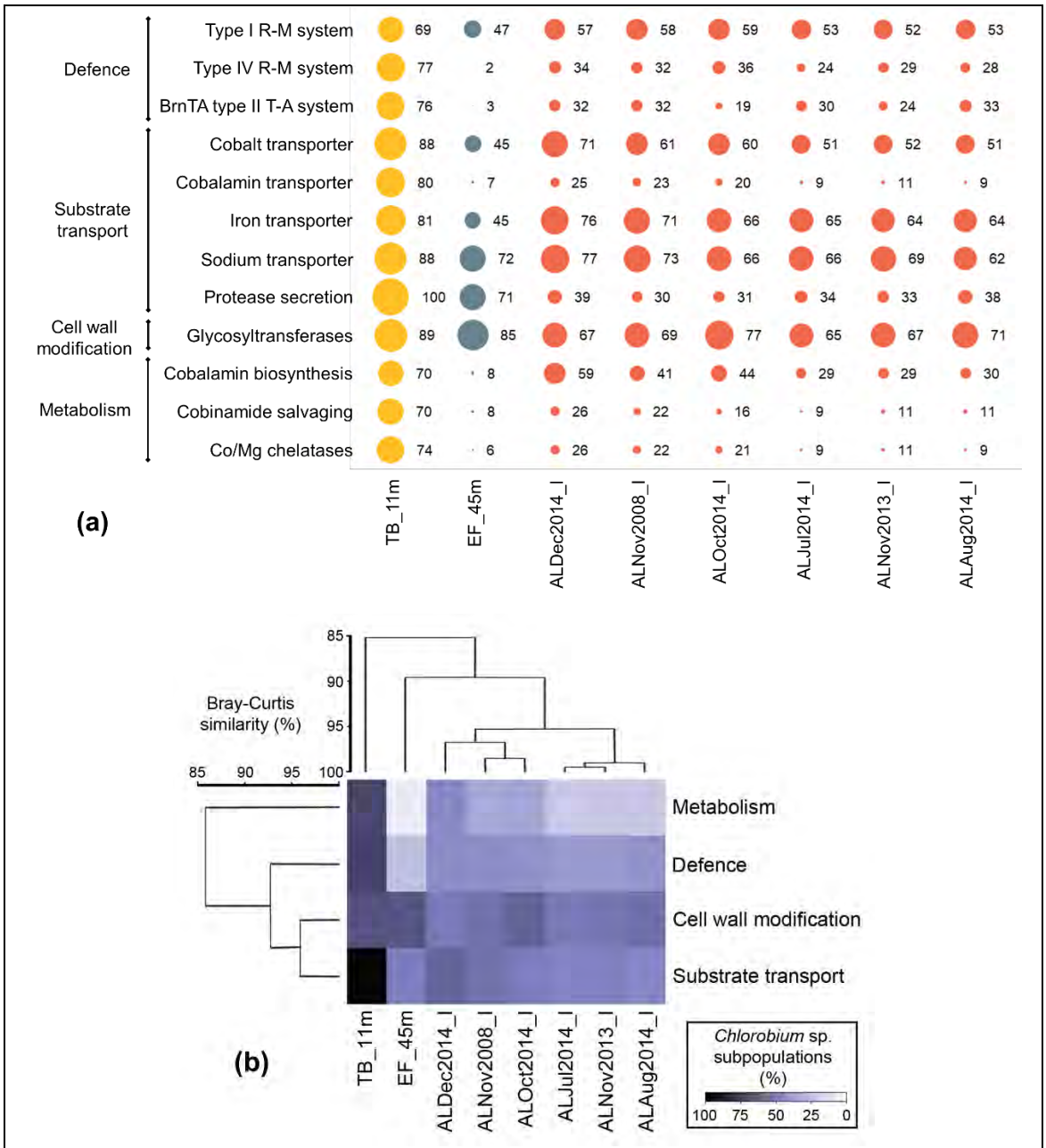


Figure 5.11 Abundance and clustering of *Chlorobium* subpopulations containing low coverage genes of EF_ref MAG. The scatter plot **(a)** and heat map **(b)** show the approximate abundances of *Chlorobium* subpopulations carrying the low coverage EF_ref MAG genes associated with metabolism, cell defence, cell wall modification, and substrate transport. **(a)** The widths of the bubbles in the scatter plot indicate the relative coverages of the genes or operons in Ace Lake, Ellis Fjord, and Taynaya Bay metagenomes. The y-axis indicates the functional

categories and the pathways or reactions in which these *Chlorobium* genes were involved. The x-axis indicates the metagenomes from Ace Lake Interface (red; AL), Ellis Fjord (grey; EF), and Taynaya Bay (yellow; TB), arranged in the order in which they clustered. **(b)** The clustering of samples (x-axis; AL, EF, TB metagenomes) and variables (y-axis; gene functions) was based on the relative coverages of the EF_ref MAG genes and operons in metagenomes from the three systems. The Bray-Curtis similarity as well as sample and variable clustering using UPGMA method were performed in Primer v7. The relative coverages of the genes, which also represented the *Chlorobium* subpopulation abundances, were calculated against EF_ref MAG mean read depth using the method described in section 5.2.3.3. The genes involved in various functions were associated with: Metabolism — cobalamin biosynthesis, cobinamide and pseudocobalamin salvaging, and cobalt/magnesium chelatases; Defence — type I R-M subunits M and R, type IV R-M enzyme, and BrnA antitoxin protein; Cell wall modification — glycosyltransferases, phosphatidylinositol alpha-1,6-mannosyltransferase, and UDP-N-acetyl-D-mannosaminuronic acid dehydrogenase; Substrate transport — cobalt transporter, iron complex transporter, cobalamin transporter, Na^+ ion transporter, and alkaline protease transporter. Co/Mg chelatases, cobalt/magnesium chelatases: I, Interface.

The *Chlorobium* subpopulations containing genes for cobalamin biosynthesis, cobinamide and pseudocobalamin salvaging, cobalt/magnesium chelatases, cobalt transporter, and vitamin B12 transporters were prevalent in Ace Lake, Ellis Fjord, and Taynaya Bay (Figure 5.11a). It is possible that these genes (except *cbiZ*) play a similar role in *Chlorobium* from all three systems, i.e., improve the photosynthetic capacity of *Chlorobium* by supporting its bacteriochlorophyll production and chlorosome formation. The gene involved in cobinamide and pseudocobalamin salvaging (*cbiZ*) might be limited to cobinamide salvaging in Ellis Fjord and Taynaya Bay, considering that the relative abundance of *Cyanobacteria* was <1% in the metagenomes from these two systems. The genes potentially involved in cell wall modification and Na^+ ion export were also present in *Chlorobium* from all three systems (Figure 5.11a). As discussed earlier, the Ace Lake *Chlorobium* genes for cell wall modification and sodium export might be associated with adaptation to cold environment and maintenance of cell homeostasis, respectively (sections 5.4.1.1 and 5.4.1.3). It is likely that these genes play a similar role in Ellis Fjord and Taynaya Bay *Chlorobium*, as both these systems are cold and have high salinity, nearly the same as that in the bottom-most waters of Ace Lake — 3.5% in Ellis Fjord 45 m depth, 3.8% in Taynaya Bay 11 m depth, 3.4–4.2% in Ace Lake 24 m depth.

Among the low coverage *Chlorobium* genes coding for cell defence proteins, the Type I R-M system genes were present in a similar number of *Chlorobium* from the three systems (Figure 5.11a). However, the Type IV R-M system and T-A system genes were present in a very small *Chlorobium* population in Ellis Fjord, but relatively larger *Chlorobium* populations in Ace Lake and Taynaya Bay (Figure 5.11a). Interestingly, only one viral cluster (cl_248) was identified as a potential *Chlorobium* virus in Ellis Fjord, compared to three and 10 *Chlorobium* viral clusters and singletons in Ace Lake and Taynaya Bay, respectively. The coverage of *Chlorobium* genes involved in substrate transport also varied in Ace Lake, Ellis Fjord, and Taynaya Bay. Generally, the relative coverages of the genes identified in the LCRs of EF_ref MAG that were associated with *Chlorobium* phylotypes and ecotypes were >70% in Taynaya Bay, but on average 42% and 33% in Ace Lake and Ellis Fjord, respectively. This indicated that these *Chlorobium* subpopulations contributed to a major portion of the *Chlorobium* population in Taynaya Bay, but not so much in Ace Lake and Ellis Fjord (Figure 5.11a).

The FR analysis of *Chlorobium* EF_ref MAG in Ace Lake, Ellis Fjord, and Taynaya Bay metagenomes also revealed SNPs in many *Chlorobium* genes present on non-variable coverage regions of EF_ref MAG and involved in metabolic and cellular functions, indicating that the *Chlorobium* in these three aquatic systems might represent different phylotypes. Interestingly, mutations in *Chlorobium* EF_ref MAG genes were more prevalent in Ace Lake than in Taynaya Bay, suggesting that the genomic sequence of Ellis Fjord and Taynaya Bay *Chlorobium* might be more similar to each other than to Ace Lake *Chlorobium*. Ellis Fjord and Taynaya Bay are connected to the Southern Ocean by narrow water channels, unlike Ace Lake, which is isolated from the ocean. This biogeographic partitioning could be relevant to these genomic distinctions between Ace Lake *Chlorobium* and the *Chlorobium* from Ellis Fjord and Taynaya Bay.

5.4.2.2 The endemicity of the Vestfold Hills *Chlorobium*

The initial IMG taxonomic assignments of the Ace Lake, Ellis Fjord, and Taynaya Bay *Chlorobium* MAGs (referred to as the Vestfold Hills *Chlorobium* here) showed that they were closely related to C-phaeov. However, the comparative genomic analyses of these *Chlorobium* MAGs with C-phaeov genome showed that the Vestfold Hills *Chlorobium* belonged to a different species than C-phaeov (Figures 5.4 and 5.5). An analysis of their functional potential showed some variations in their functional capacities (Figure 5.12). Compared to C-phaeov, the Vestfold Hills *Chlorobium* had

higher capacity for cell wall synthesis and modification ('M'), protein modification ('O'), translation ('J'), transcription ('K'), replication ('L'), energy production ('C'), transport and metabolism of amino acids ('E'), coenzymes ('H'), and inorganic ions ('P'), and contained more poorly characterised genes ('R', 'S') (Figure 5.12). On the other hand, C-phaeov contained genes associated with extracellular structures ('W') and cytoskeleton ('Z'), which were not identified in the Vestfold Hills *Chlorobium*, but this lack of genes might be due to the incomplete assembly of the Vestfold Hills *Chlorobium* MAGs.

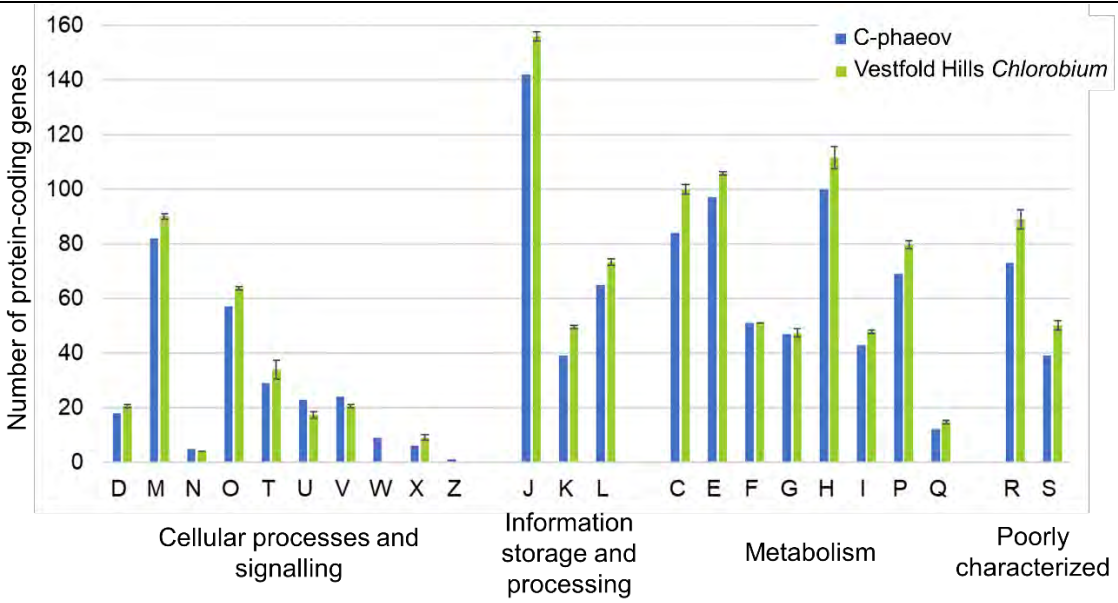


Figure 5.12 Comparison of functional potential of the Vestfold Hills *Chlorobium* and C-phaeov. The bar chart shows the number of protein-coding genes (y-axis) from the Vestfold Hills *Chlorobium* (green bars) and C-phaeov (blue bars) that were classified to various COG categories (x-axis). The data for the Vestfold Hills *Chlorobium* was calculated from the COG categorisation of protein-coding genes in AL_ref MAG (from Ace Lake), EF_ref MAG (from Ellis Fjord), and a Taynaya Bay *Chlorobium* MAG generated from 11 m depth metagenome. Therefore, the green bars represent the average number of *Chlorobium* genes from the three Vestfold Hills systems that were classified to each COG category and the error bars indicate the standard deviation. The COG categories do not include categories 'Y', 'A', and 'B', as none of the protein-coding genes from C-phaeov or the Vestfold Hills *Chlorobium* were classified under these categories. COG categories: A, RNA processing and modification; B, Chromatin structure and dynamics; C, Energy production and conversion; D, Cell cycle control, cell division, chromosome partitioning; E, Amino acid transport and metabolism; F, Nucleotide transport and metabolism; G, Carbohydrate transport and metabolism; H, Coenzyme transport and metabolism; I, Lipid transport and metabolism; J, Translation, ribosomal structure and

biogenesis; K, Transcription; L, Replication, recombination and repair; M, Cell wall/membrane/envelope biogenesis; N, Cell motility; O, Posttranslational modification, protein turnover, chaperones; P, Inorganic ion transport and metabolism; Q, Secondary metabolites biosynthesis, transport and catabolism; R, General function prediction only; S, Function unknown; T, Signal transduction mechanisms; U, Intracellular trafficking, secretion, and vesicular transport; V, Defence mechanisms; W, Extracellular structures; X, Mobilome: prophages, transposons; Y, Nuclear structure; Z, Cytoskeleton.

Unlike C-phaeov genome, the Vestfold Hills *Chlorobium* MAGs did not contain *sox* genes, assimilatory sulfate reduction genes, and pilus assembly genes. It has been speculated that C-phaeov acquired its *sox* gene cluster from another member of the *Chlorobiaceae* family through HGT of a mobile element, and GSB originally acquired this cluster from *Proteobacteria* (Gregersen et al, 2011). The *sox* gene operon and assimilatory sulfate reduction genes have been previously reported to be absent in Ace Lake *Chlorobium* (Ng et al, 2010). The absence of assimilatory sulfate reduction genes is a characteristic of most GSB (Frigaard and Bryant, 2008). Therefore, it is likely that the Vestfold Hills *Chlorobium* does not have the capacity to assimilate sulfate or to oxidise thiosulfate. As for viral defence capabilities, C-phaeov contained a subtype I-C CRISPR-Cas system, whereas the Vestfold Hills *Chlorobium* contained a subtype I-E CRISPR-Cas system.

The comparison of C-phaeov genome and the Vestfold Hills *Chlorobium* MAGs also showed that C-phaeov did not contain some of the genes identified on the *Chlorobium* MAGs. Notably, most of these *Chlorobium* MAG genes had low coverage in Ace Lake, Ellis Fjord, and Taynaya Bay and were associated with cell defence (type I and type IV R-M system genes), substrate transport (cobalt, vitamin B12, iron, sodium ion, and protease transporters), cell wall modification (glycosyltransferases), and metabolic functions (cobalamin biosynthesis, cobinamide salvaging, cobalt/magnesium chelatases) (Figures 5.11 and 5.12). Overall, these genes and operons not only contributed toward some of the differences observed within the Vestfold Hills *Chlorobium* population, but also between C-phaeov and the Vestfold Hills *Chlorobium*.

To further assess the endemicity of the Vestfold Hills *Chlorobium*, its marker sequences (*16S rRNA* gene and BclA protein) were compared to the marker sequences in the IMG metagenomic and genomic databases. The Vestfold Hills *Chlorobium 16S rRNA* gene was ≤99% similar to the marker genes from metagenomes and genomes from various

global sites. Similarly, the Vestfold Hills *Chlorobium* BclA protein was <98% similar to the marker proteins from various GSB (except C-phaeov, 98% similarity). Together, these findings indicated that the Vestfold Hills *Chlorobium* was distinct from other GSB and was likely endemic to this Antarctic region.

5.4.3 The Vestfold Hills *Chlorobium* potential for defence against viruses

The analysis of annotated genes in *Chlorobium* MAGs revealed a number of genes potentially associated with cell defence and immunity. These included CRISPR-Cas system genes, R-M system genes, and T-A system genes. The Ace Lake *Chlorobium* contained a complete subtype I-E CRISPR-Cas system (*cas3ABECD12* operon) indicating its capacity to defend against viruses. The presence of a CRISPR-Cas system in Ace Lake *Chlorobium* has been previously reported (Ng et al, 2010; Lauro et al, 2011). An analysis of the annotated genes in the genomes of various GSB available on NCBI showed that the presence of CRISPR-Cas system genes was common in GSB and was not limited to a few subtype systems, and that some species contained genes for multiple subtype systems. For example, C-phaeov, *C. chlorochromatii* CaD3, and *C. luteolum* DSM 273 had subtype I-C; *Cb. tepidum* TLS contained subtype I-C and I-E; *C. phaeobacteroides* DSM 266 had subtype III-A and I-C; *C. phaeobacteroides* BS1 contained subtype III-A and I-E; *Cb. parvum* NCIB 8327 contained subtype III-A; *C. limicola* DSM 245 contained subtype I-B and III-B; and *C. phaeovibrioides* GrTcv12 had subtype I-F. Some of these GSB defence genes have been reported in previous publications (Eisen et al, 2002; Mansor and Macalady, 2016; Boldyreva et al, 2020). Despite this variety of defence systems identified in GSB, only a few GSB viruses have been identified and reported to date, including the ones from Lake Banyoles in Spain, Trout Bog Lake in USA, and Ace Lake in Antarctica (Llorens–Marès et al, 2017; Berg et al, 2020; Panwar et al, 2020).

The Vestfold Hills *Chlorobium* contained genes associated with type I and type IV R-M systems indicating its capacity to neutralise foreign DNA such as phage, including viruses capable of modifying their genome. Notably, the type I R-M genes were present only in a subpopulation of *Chlorobium* (section 5.4.1.2). Five genes associated with T-A systems (*parD*, *parE*, *relF*, *brnA*, *abiEi*) were identified in *Chlorobium*, with *brnA* being located in LCR. However, only the *abiEi* T-A system antitoxin gene was potentially involved in viral disruption through the ABI mechanism. The gene coding

for the toxin component (*abiEii*) of this T-A system was not identified among the annotated genes of *Chlorobium* MAGs, but it is possible that the gene was annotated as hypothetical or uncharacterised protein or was not a part of the MAG assemblies. The AbiE type IV T-A system is involved in the ABI mechanism (Dy et al, 2014). Its toxin component (AbiEii) does not cause immediate cell death on viral infection, but induces cell dormancy, which reverts once the cell is re-exposed to the antitoxin component (AbiEi) (Dy et al, 2014). Considering this, it is likely that *Chlorobium* cells infected with viruses would become dormant under the effect of AbiEii toxin, preventing spread of the virus to surrounding uninfected populations. Once the virus is neutralised, the *Chlorobium* cell would regain its ability to produce AbiEi antitoxin and exit dormant state. Based on the matches to *Chlorobium* spacers, two viral contig clusters (cl_1024, cl_248) and one singleton viral contig (sg_14554) were identified as potential *Chlorobium* viruses, which had positive abundance correlation to *Chlorobium* (Chapter 3 section 3.3.5.6). It is possible that the dormant *Chlorobium* population infected with viruses contributed toward the positive correlation observed between *Chlorobium* and its potential viruses.

The potential Ace Lake, Ellis Fjord, and Taynaya *Chlorobium* viruses (cl_1024, cl_248, cl_9176, sg_1370, sg_1648, sg_1649, sg_1677, sg_10581, sg_14551, sg_14554, sg_14796, sg_14959) had a broad host range. Their hosts included *Chlorobium* and members of *Gammaproteobacteria* as well as some *Actinobacteria*, *Betaproteobacteria*, *Deltaproteobacteria*, *Firmicutes*, *Flavobacteriia*, and *Verrucomicrobia*, indicating that they were generalist viruses. Marine cyanobacteria usually resist generalist viruses at an intracellular-level (Zborowskya and Lindell, 2019). This may also be true of Ace Lake *Chlorobium*, which contained an arsenal of intracellular defence systems (CRISPR-Cas, R-M, T-A systems). Although, some the Vestfold Hills *Chlorobium* had additional genes for glycosyltransferases, substrate transporters, and outer membrane proteins, none of these genes had any mutations that might contribute toward changes in the cell surface structure. This suggested that *Chlorobium* immunity and defence was probably not extracellular and it can be speculated that the Vestfold Hills *Chlorobium* probably does not have any specialist viruses that target it. Therefore, unchecked *Chlorobium* propagation due to lack of specialist *Chlorobium* viruses might be a contributing factor toward its high abundance in Ace Lake in summer and spring (Chapter 3 section 3.3.4).

5.4.4 Biogeographic distribution of viruses associated with the Vestfold Hills *Chlorobium*

A number of viral contig clusters and singletons were associated with *Chlorobium* from Ace Lake (cl_1024, cl_248, sg_14554), Ellis Fjord (cl_248), and Taynaya Bay (cl_1024, cl_9176, sg_1370, sg_1648, sg_1649, sg_1677, sg_10581, sg_14551, sg_14796, sg_14959). Analysis of the Antarctic systems from which these viral contigs originated showed the biogeographic distribution of potential *Chlorobium* viruses (Figure 5.13). The Ace Lake *Chlorobium* spacers had ≥97% identity matches to viral contigs from Ace Lake as well as Deep Lake, Club Lake, Organic Lake, and some Rauer Island lakes (Rauer 2, 3, 5, 6, 11, and 13 lakes). Similarly, the Taynaya Bay *Chlorobium* spacers had matches to Ace Lake as well as Deep Lake and Rauer 13 Lake viral contigs (Figure 5.13). On the other hand, only one spacer sequence from Ellis Fjord *Chlorobium* had matches to some Ace Lake viral contigs. This data indicated that the potential *Chlorobium* viruses were probably widely distributed in the Antarctic systems of the Vestfold Hills and the Rauer Islands (Figure 5.13). Some similarities in the microbial compositions of hypersaline lakes from the Rauer Islands and the Vestfold Hills has been observed, although the hypersaline lakes from the Vestfold Hills are dominated by haloarchaea, whereas the Rauer Island hypersaline lakes are dominated by either bacteria or archaea (Tschitschko et al, 2018). As *Chlorobium* have not been reported in any of these lakes, except Ace Lake, Ellis Fjord, and Taynaya Bay, these findings also highlighted the broad host range of these potential *Chlorobium* viruses. The other potential hosts of these viruses, such as the members of *Gammaproteobacteria*, are prevalent in Organic Lake and a small population was also identified in Deep Lake and some Rauer Island lakes (Bowman et al, 2000a; DeMaere et al, 2013; Yau et al, 2013; Tschitschko et al, 2018 unpublished data).

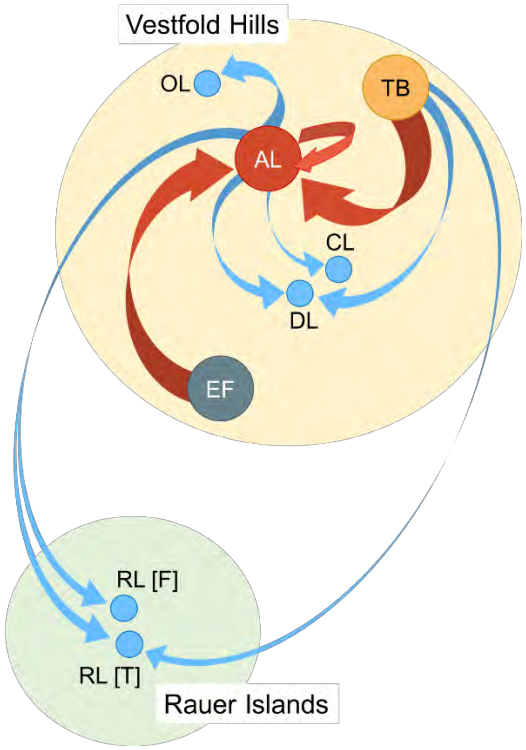


Figure 5.13 Biogeographic distribution of viral contigs with matches to the Vestfold Hills *Chlorobium* spacers. The schematic shows the Antarctic aquatic systems from the Vestfold Hills and the Rauer Islands to which the viral contigs with matches to the Vestfold Hills *Chlorobium* spacers belonged. The *Chlorobium* spacers were from Ace Lake (AL, red circle), Ellis Fjord (EF, grey circle), and Taynaya Bay (TB, yellow circle). The viral contigs were from Ace Lake, Ellis Fjord, Taynaya Bay, Deep Lake (DL), Club Lake (CL), Organic Lake (OL), and Rauer lakes from Filla Island (RL[F]) and Torckler Island (RL[T]). The arrows originate from the system that contained the *Chlorobium* spacers and point toward the system containing the matching viral contigs. The widths of the arrows roughly indicate the number of spacers from a system that matched the viral contigs from another system. For example, most of the spacers from Taynaya Bay had matches to Ace Lake viral contigs and a few matched the viral contigs from Deep Lake and Rauer Island lakes. The red arrows show connections between Ace Lake, Ellis Fjord, and Taynaya Bay, in which *Chlorobium* was identified, whereas blue arrows show connections to all other Antarctic aquatic systems. The systems are arranged approximately in the order of their location in the Vestfold Hills (AL, EF, TB, DL, CL, OL) and the Rauer Islands (RL[F], RL[T]) in Antarctica.

5.5 Conclusion

The continuing presence and high abundance of *Chlorobium* in the Ace Lake oxycline emphasizes its adaptation to the lake environment. The analysis of genomic variation in the Ace Lake *Chlorobium* populations from six different time periods ranging across seven years from 2008 to 2014 showed that the genome of this microbe was very stable, with no mutations in its genomic sequence from different time periods. However, a closer look at its genomic composition revealed subpopulations that represented Ace Lake *Chlorobium* phylotypes and ecotypes and whose abundances varied with season. Some of the additional genes identified in *Chlorobium* subpopulations probably provided them with the added capacity to adapt to cold environment, to actively export sodium ions out of their cells, and to defend against viruses and other foreign DNA (section 5.4.1). *Chlorobium* subpopulations containing additional genes for iron import would also be at an advantage, as iron is an essential trace element and a part of the photosynthetic reaction centre of *Chlorobium* (section 5.4.1.3). Apart from these, the *Chlorobium* subpopulations containing additional genes for cobalt and vitamin B12 transport, cobalamin biosynthesis, cobinamide and pseudocobalamin salvaging, and cobalt/magnesium chelatases probably represented *Chlorobium* ecotypes that had the capacity for *de novo* synthesis of cobalamin as well as cobalamin production from salvaged precursors (section 5.4.1.4). Moreover, the *Chlorobium* subpopulations containing the genes for cobinamide and pseudocobalamin salvaging and vitamin B12 transporter might be capable of interacting with Ace Lake *Synechococcus*, during which the pseudocobalamin potentially produced by the cyanobacteria would be absorbed and salvaged by the *Chlorobium* for cobalamin biosynthesis. *Chlorobium* bacteriochlorophyll content and chlorosome formation rely on the availability of cobalamin (Sato et al, 1981; Fuhrmann et al, 1993). Therefore, it is likely that the cobalamin produced by the Ace Lake *Chlorobium* helps it to recuperate after a long, dark winter and rise to very high abundance in summer.

An analysis of the distribution of *Chlorobium* in Ace Lake, Ellis Fjord, and Taynaya Bay, three meromictic systems in the Vestfold Hills, showed that the same species of *Chlorobium* was prevalent in all three stratified systems. A comparative analysis of the Vestfold Hills *Chlorobium* with its closest related species C-phaeov, showed that the two organisms were distinct species with different genomic compositions (section 5.4.2.2). Furthermore, a comparison of the Vestfold Hills *Chlorobium* markers to GSB markers in the IMG metagenomic and genomic databases showed that this *Chlorobium*

was different from other GSB, highlighting its endemicity to the Vestfold Hills (section 5.4.2.2). Interestingly, all three systems contained the same *Chlorobium* subpopulations, but their abundances varied in the three aquatic systems (Figure 5.11a, b). The relative coverages of the genes that contributed toward *Chlorobium* phylotypes and ecotypes were higher in Taynaya Bay (>70%) than in Ace Lake and Ellis Fjord (on average 42% and 33%, respectively), indicating that these phylotypes and ecotypes composed a major portion of the Taynaya Bay *Chlorobium* population, but not Ace Lake and Ellis Fjord *Chlorobium* populations (Figure 5.11a). The FR of EF_ref MAG (Ellis Fjord *Chlorobium* MAG) showed no mutations in Ellis Fjord metagenomes, few mutations (21 SNPs) in Taynaya Bay metagenomes, and many mutations (87 SNPs) in Ace Lake metagenomes (described in section 5.3.4.2). Notably, most of these mutations (except three) were located in non-LCRs, indicating that each of these mutations was prevalent in all *Chlorobium* subpopulations from the system in which they were identified. Together, these findings indicated that the *Chlorobium* from Ellis Fjord and Taynaya Bay (both of which are linked to the Southern Ocean) were more similar to each other than to the *Chlorobium* from Ace Lake (which is landbound), indicating that the biogeographic partitioning of the three systems might contribute toward this genomic distinction of *Chlorobium*.

The Vestfold Hills *Chlorobium* had three viral clusters and nine viral singletons associated with it, representing potential viral predators (section 5.4.4). These potential *Chlorobium* viruses had a broad host range, and as a defence against these generalist viruses, the Vestfold Hills *Chlorobium* had a number of intracellular defence systems, including subtype I-E CRISPR-Cas system, type I and IV R-M systems, and AbiE type IV T-A system (section 5.4.3). Moreover, the viral predators of the Vestfold Hills *Chlorobium* appeared to be spread across a large region, from the Vestfold Hills to the Rauer Islands in East Antarctica (Figure 5.13).

Overall, the findings in this chapter showed that a single species of *Chlorobium* was prevalent in Ace Lake, Ellis Fjord, and Taynaya Bay and that this species was endemic to the stratified systems in the Vestfold Hills. Moreover, similar phylotypes and ecotypes of these GSB were present in all three systems, but with varying abundances. This Vestfold Hills *Chlorobium* also had the capacity for defence against viruses using intracellular defence systems. The viruses potentially associated with the Vestfold Hills

Chlorobium had a broad host range and were distributed across aquatic systems in the Vestfold Hills as well as the Rauer Islands.

6. Conclusion

This thesis describes the first metagenomics-led analysis of the effects of seasonal variation on Ace Lake microbial population and functional dynamics, using an in-house metagenome analysis pipeline (Cavlab pipeline) and other genomic methods. The Cavlab metagenome analysis pipeline was developed for and tested on IMG-annotated Antarctic metagenomes, and allowed for the analysis of the taxonomic composition and functional potential of Ace Lake microbial community using a time-series of metagenomes. The genomic analyses of the two key species of Ace Lake, namely *Synechococcus* and *Chlorobium*, led to the identification of their phylotypes and ecotypes in the lake. With the availability of metagenomes and MAGs from Ace Lake, Ellis Fjord, and Taynaya Bay, three stratified aquatic systems in the Vestfold Hills, the draft genomes of *Chlorobium* from the three Antarctic systems were compared to each other and the complete genome of a non-Antarctic *Chlorobium* species, to assess the endemicity of this GSB to the Vestfold Hills. In this final chapter, a summary of the main findings of the thesis are discussed along with the prospects for future work.

6.1 The importance of the development of the Cavlab pipeline — an Antarctic metagenome analysis pipeline

Metagenomes are large datasets that represent snapshots of genetic information from an environmental site, and the in-depth analysis of metagenomes can be a time-exhaustive task. Although various software are available for different types of analyses of metagenomes such as taxonomic, abundance and functional potential analyses, none of them can be used for complete metagenomic analysis by themselves. Therefore, it can be useful to have a pipeline that when launched, automatically performs an initial suite of comprehensive analyses on select metagenomes, generating outputs that can be the base for more detailed analyses. For example, a pipeline that automatically performs taxonomic, abundance, and functional potential analyses on a metagenomic dataset, requiring no input from the user once launched, can be time saving and useful compared to running individual analyses on each metagenome, which would require data preparation and verification for each analysis. The Cavlab pipeline was specifically

developed for such analyses of Antarctic metagenomes annotated by JGI's IMG system, by exploiting the IMG folder structure to perform the analyses (Chapter 2 section 2.3.6). The computational methods tested for the analysis of taxonomy, abundance, and functional potential of Antarctic metagenomes showed that not all methods worked with the Antarctic metagenome data (Chapter 2). Therefore, the methods used in Cavlab pipeline were carefully evaluated and selected based on their data reproducibility and robustness (Chapter 2).

In this thesis, the Cavlab pipeline was used for the initial comprehensive analyses of 120 Ace Lake, 12 Ellis Fjord, and four Taynaya Bay metagenomes. The taxonomy and abundance data generated through the pipeline were used for assessing the microbial diversity of the three systems and identifying the most abundant microbes (Chapters 3 and 5). The relative abundances of microbes calculated from the abundance data generated by the pipeline allowed for a direct comparison between the microbial populations of the three systems (Chapter 5 section 5.3.3). Therefore, the use of Cavlab pipeline allowed for analysis and comparison of the different systems.

The Cavlab pipeline comprised of separate code subsections associated with different analyses, such as contig taxonomy and abundance analysis, protein taxonomy and function analysis, COG function analysis, and KEGG function analysis (Chapter 2 Figure 2.13; Appendix C). This allowed for parallel running of these analyses, making the Cavlab pipeline runs time-efficient. Due to this segregation of analyses (through the use of separate code subscripts), the Cavlab pipeline is quite flexible, making it relatively easy to test new analyses, methods, and/or databases as well as to upgrade the pipeline. To test and include a new method or analysis, the user would have to — verify new input files at the beginning of the pipeline, create a code subsection for the new method or analysis, add command to call the code, and call only the code for the method or analysis being tested, so that it could be tested without having to run all other analyses in the pipeline. To update a method or a database associated with an existing analysis, the user would have to — verify new input files (if any) at the beginning of the pipeline and modify the code subsection associated with the analysis.

6.2 Microbial and viral population dynamics of Ace Lake

The isostatic rebound of the Vestfold Hills landmass nearly 10,000 years ago led to the formation of many marine water bodies, including Ace Lake, in this region of East Antarctica (Gibson, 1999; Cavicchioli, 2015; Siegert et al, 2016). Ace Lake is a permanently stratified lake with an Upper oxic zone, an oxic-anoxic Interface, and a Lower anoxic zone — the presence of an ice cover for most part of the year and a strong salinity gradient are responsible for such stable stratification of the lake (Walker, 1974; Burton and Barker, 1979; Burch, 1988; Rankin et al, 1999). However, it has been speculated that since its isolation from the ocean, Ace Lake must have completely mixed at least once, during which most of the sulfur (76%) was lost from the lake (Burton and Barker, 1979; Gibson and Burton, 1996; Rankin et al, 1999). Light penetration in Ace Lake depends on the thickness, age, and quality of the ice cover, presence/absence of a snow cover, and the amount of biomass in the Upper zone and Interface of the lake (Burch, 1988; Kirk, 1994; Rankin, 1998; Rankin et al, 1999). Generally, in summer, light penetrates to ~12 m depth in Ace Lake in ice-free conditions (Hand and Burton, 1981; Burch, 1988; Rankin, 1998; Rankin et al, 1999).

6.2.1 Microbial population

In this thesis, the Cavlab pipeline was used for taxonomic, abundance, and functional potential analyses of Ace Lake metagenomes (Chapter 3). The microbial community of Ace Lake was segregated by depth, with distinct microbes identified in the oxic, anoxic, and interface zones of the lake (Chapter 3 section 3.3.3). This niche segregation of Ace Lake microbes has been reported previously (Rankin et al, 1999; Lauro et al, 2011). The Upper oxic zone of Ace Lake mainly contained phototrophic eukarya (*Micromonas*) and bacteria (*Synechococcus*) along with a variety of algal viruses (*Phycodnaviridae* 1–5). The Interface contained a high abundance population of GSB (*Chlorobium*) and some *Deltaproteobacteria* (*Desulfatiglanales* NaphS2, *Desulfobacterales* S5133MH16, *Desulfobacterium*, *Desulfocapsa*, *Syntrophales* UBA2210). The Lower anoxic zone harboured obligate anaerobes including the Deltaproteobacteria prevalent in the interface zone as well as some members of bacterial candidate phyla (*Atribacteria* 34-128, *Cloacimonetes* JGIOTU-2) and methanogenic archaea (*Methanomicrobiaceae* 1, *Methanothrix*_A). The impact of change in season was evident from the variations in the abundances of the microbes in the Upper zone and Interface of Ace Lake, especially the phototrophs that relied on light for primary production (Chapter 3 section 3.3.4). On the other hand, the abundances of microbes in the Lower zone of Ace Lake showed little

variation with change in season, which was consistent with their reliance on chemolithoautotrophy for energy production.

The method used for contig taxonomic assignment, which was part of the Cavlab pipeline, depended on the protein taxonomies provided by IMG. Therefore, contigs containing unclassified proteins, contigs that did not contain proteins, and contigs that could not be assigned unambiguous taxonomies (i.e., did not fulfill the criteria for contig taxonomy assignment; Chapter 2 section 2.2.2.5) were termed 'unassigned contigs' and represented the 'dark matter' in Antarctic metagenomes. The analysis of 'unassigned contigs' of length ≥1 kb showed that they were mainly associated with uncultured microbes and might represent novel taxa (Chapter 3 section 3.3.6). The genes on these contigs were generally annotated as 'hypothetical proteins' and might represent novel proteins. However, most of the 'unassigned contigs' were <1 kb in length. Generally, small contigs (usually <500 bp) with low coverage could represent spurious sequencing products. Therefore, the lengths and coverages of these small contigs need to be evaluated to assess whether they represent contamination or possibly rare or novel taxa or viruses. Further investigation of these contigs and genes is required to identify their microbial origin and function.

The comparison of 'unassigned contigs' with known databases such as GTDB (using RefineM) or NCBI-nr database (using DIAMOND/MEGAN6 or LAST/MEGAN-LR) might shed some light on their taxonomy, which might help in obtaining a more complete picture of the microbial diversity of Ace Lake. To assess the potential functions of the hypothetical genes on these contigs, their protein sequences could be run through software such as MG-RAST (Meyer et al, 2008) or DeepEC (Ryu et al, 2019). MG-RAST is an open source, online service for metagenome analysis that uses matches to manually curated protein families (SEED FIGfam; Meyer et al, 2009) and subsystems (Overbeek et al, 2005) for functional annotation of predicted proteins. DeepEC is a more recently developed high-throughput annotation approach for high-precision assignment of enzyme commission (EC) numbers to predicted proteins, to assess their potential enzymatic functions. It uses three convolutional neural networks (CNNs), a class of deep neural networks, to predict protein functions, where the first CNN predicts whether the input protein is an enzyme and the second and third CNNs predict EC numbers up to third- and fourth-level, respectively (Meyer et al, 2009). DeepEC protocol also uses homology analysis to predict EC numbers of proteins that

were predicted to be enzymes by the first CNN, but could not be assigned EC numbers by the second and third CNNs (Meyer et al, 2009).

6.2.2 Viral population

The analysis of viral data, including viral contigs representing complete genomes of viruses, revealed a diverse viral population in Ace Lake, especially in the Upper oxic zone (Chapter 3 section 3.3.5). Apart from the five algal viruses probably associated with *Micromonas*, viruses potentially associated with *Synechococcus* and *Chlorobium* were identified in Ace Lake; all three hosts were phototrophs. No significant correlation was observed between the abundances of *Synechococcus* or *Micromonas* and their respective viruses, whereas a positive correlation was observed between *Chlorobium* and its viruses (Chapter 3 section 3.3.5). The findings in this thesis indicated that the availability of light, rather than viral predation, was probably responsible for the seasonal variations observed in the abundances of these phototrophic hosts. The complete genome of a 'huge' phage containing defence genes (*cas* genes) was also identified in Ace Lake, and it was speculated that these defence genes might give it a competitive edge by allowing it to target other viruses that could infect its potential host (Chapter 3 section 3.3.5.2). The overall abundance of the viral population in Ace Lake did not appear to vary with seasonal changes.

Apart from the 'huge phage' complete genome, the complete genome of around 172 viruses were identified in metagenomes from Ace Lake, and their potential hosts need to be determined to assess their impact on the Ace Lake microbiome. The potential hosts of these viruses could be analysed using the spacer database, which contained the matches of host spacers to Antarctic viral contigs (Chapter 3 section 3.2.6). In this thesis, a similar method was applied for the host verification analysis of potential *Chlorobium* viruses (Chapter 3 section 3.2.6.1). The viral contigs were compared with the spacer database to identify host contigs, whose taxonomy was assessed through the method used in Cavlab pipeline (Chapter 3 section 3.2.1).

The taxonomy and abundance analysis of Ace Lake had identified a number of abundant OTUs. An analysis of the potential viruses of the abundant OTUs, other than *Micromonas*, *Synechococcus*, and *Chlorobium* (which have already been analysed; Chapter 3 section 3.2.6), might reveal additional information about the population dynamics of Ace Lake. A method similar to the one used for identifying potential

Chlorobium viruses in Ace Lake, Ellis Fjord, and Taynaya Bay (Chapter 3 section 3.2.6.1 and Chapter 5 section 5.2.5) could be utilised to identify the viruses associated with Ace Lake abundant OTUs that contain CRISPR-Cas system genes.

6.2.3 Seasonal variation in Ace Lake

Ace Lake has been studied extensively for nearly four decades, with emphasis on its physicochemical characteristics as well as microbial diversity and function (Hand, 1980; Hand and Burton, 1981; Burch, 1988; Burke and Burton, 1988; Gibson and Burton, 1996; Rankin et al, 1997; Rankin, 1998; Bell and Laybourn-Parry, 1999; Rankin et al, 1999; Laybourn-Parry et al, 2005; Madan et al, 2005; Powell et al, 2005; Ng et al, 2010; Lauro et al, 2011; Laybourn-Parry and Bell, 2014). Although some seasonal studies have been conducted on Ace Lake data (Burch, 1988; Gibson and Burton, 1996; Bell and Laybourn-Parry, 1999; Rankin et al, 1999), this thesis described the first metagenomics-led analysis of seasonal variation in the microbial community of Ace Lake.

The polar light cycle is distinct from the light cycle experienced in the lower latitudes including high altitude cold areas, with 24 h of sunlight in summer and 24 h of darkness in winter for a few weeks to months, depending on the latitude. This stark contrast in summer/winter light availability is likely to shape the Antarctic microbial communities, alongside important environmental factors such as temperature. Considering this, the effect of change in season on the microbial diversity of Ace Lake was studied using a time-series of 120 metagenomes from the surface and six depths of the lake (described in Chapter 3 sections 3.2.4.1 and 3.2.4.2). Among the high-quality OTUs, it was observed that most phototrophs (including photoheterotrophs and photoautotrophs) and other microbes in Ace Lake responded to seasonal variation, with their abundances varying in summer, winter and spring (Figure 3.10). Notably, some microbes were abundant (≥1% relative abundance) only in specific seasons, e.g., the *Polaribacter* OTU was abundant only in summer, whereas the *Pseudomonas*_E OTU was abundant only in winter (Figure 3.10).

The overall functional potential of Ace Lake microbial community in metagenomes from different seasons was also analysed (Chapter 3 section 3.2.1). This allowed for prediction of seasonal shifts in the functional potential of Ace Lake and the microbes that probably contributed to them (Chapter 3 section 3.3.7). The functional potential of

Ace Lake microbial community in summer *vs* winter has been reported in a publication (Panwar et al, 2020). For further investigation of microbial function in Ace Lake and their contribution to overall biogeochemistry of the lake, additional 'omic' analyses using metatranscriptomes and metaproteomes would be helpful. Metagenomes allow determination of the genetic composition of an environment, and can be used to identify microbial OTUs, calculate their approximate abundances and predict their probable function in the environment. On the other hand, metatranscriptomes and metaproteomes allow for better understanding of microbial community functions as they are based on RNA and protein data, respectively, collected from an environment. These tools can be used to further study the effects of change in season on the Ace Lake microbial community functions and their impact on lake biochemistry.

6.3 Ace Lake *Synechococcus* subpopulations — adaptation to the lake environment and a complex interplay with potential viruses

Synechococcus is the most abundant bacteria in the Upper zone of Ace Lake, in the depths just above the Interface (Rankin et al, 1997; Rankin, 1998; Rankin et al, 1999; Powell et al, 2005; Lauro et al, 2011). Its phenotypic characteristics, phylogeny, distribution in some stratified systems of the Vestfold Hills, and seasonal variation have been studied previously using microscopy and culture-based methods (Rankin et al, 1997; Rankin, 1998; Rankin et al, 1999; Powell et al, 2005). *Synechococcus* was found to be prevalent throughout the Ace Lake (>1% relative abundance in all depths of the lake) and appeared to recuperate from the effects of change in season much faster (10–20% relative abundance in August late winter) than the other phototrophs in the lake (Chapter 3 section 3.3.3). The analysis of *Synechococcus* functional potential revealed that it had the capacity for fermentation, which could have supported its survival and growth in the dark, anoxic waters of Ace Lake and in winter (Chapter 3 section 3.3.4). Similar fermentative ability has been reported in the *Synechococcus* isolated from the dark, anoxic waters of Black Sea (Callieri et al, 2019).

In this thesis, the Ace Lake metagenomes and *Synechococcus* MAGs were used to assess genomic variation in this cyanobacterium to identify potential phylotypes and ecotypes that might be prevalent in the lake (Chapter 4). Moreover, the auto-annotated MAG genes were parsed to identify defence genes, which were manually annotated to

verify their gene function, and the defence capacity of Ace Lake *Synechococcus* was explored (Chapter 4). The *Synechococcus* identified in metagenomes from all Ace Lake depths and time periods belonged to the same species (Chapter 4 section 4.3.2). *Synechococcus* subpopulations representing a potential phylotype with modified capacity for cell defence and immunity and a potential ecotype with the ability to utilise glutamine (in addition to ammonia) for asparagine production were identified in Ace Lake data (Chapter 4 section 4.4.1). Together, these findings highlighted the adaptation of Ace Lake *Synechococcus* to the Upper oxic zone, where a diverse population of viruses existed (Chapter 3 section 3.3.5) and bioavailable nitrogen is a limiting nutrient (Rankin et al, 1999).

Synechococcus contained genes for a variety of cell defence systems that might be used to combat viruses and other foreign DNA (Chapter 4 section 4.4.2). These included intracellular defence systems genes, such as the genes for type I, II, and III R-M system, type I BREX system, and a number of type II T-A systems including MazEF T-A (Chapter 4 section 4.4.2), which is known to be involved in the ABI mechanism for viral infection disruption (Hazan and Engelberg-Kulka, 2004; Engelberg-Kulka et al, 2005). In terms of extracellular defence in *Synechococcus*, the mutations observed in its genes for membrane-associated proteins (Chapter 4 section 4.4.2) might provide it immunity to viruses that invade host cells by attaching to host cell receptors (Avrani et al, 2011; Schmid et al, 2016; Tschitschko et al, 2015; Tschitschko et al, 2018; Zborowskya and Lindell, 2019). Marine cyanobacteria have been shown to resist both generalist (broad host range) and specialist (host-specific) viruses, using intracellular and extracellular defence strategies, respectively (Zborowskya and Lindell, 2019). Moreover, marine cyanobacteria have been reported to have a complex interaction with their viral predators, where part of the host population is resistant to the viruses, which allows for the co-existence of both host and virus populations (Coleman et al, 2006; Avrani et al, 2011; Zborowskya and Lindell, 2019). Together, these findings indicated that a similar, intricate interplay might exist between Ace Lake *Synechococcus* and its viral predators, which might also explain the lack of a linear correlation between the abundances of *Synechococcus* and its potential virus (Chapter 3 section 3.3.5.5).

6.4 Ace Lake *Chlorobium* subpopulations — adaptation and endemicity to the Vestfold Hills

In Ace Lake, Ellis Fjord, and Taynaya Bay, three stratified aquatic systems in the Vestfold Hills, the oxic-anoxic interface had high abundance of a *Chlorobium* (Chapter 5 section 5.3.3). The presence of members of *Chlorobiaceae* family in the oxic-anoxic interfaces of these three systems has been reported previously (Burke and Burton, 1988a; Coolen et al, 2006; Ng et al, 2010; Lauro et al, 2011). The phylogeny and functional potential of the *Chlorobium* in Ace Lake have been studied using *16S rRNA* gene sequencing (Imhoff, 2003; Coolen et al, 2006) and/or metagenomic data (Ng et al, 2010; Lauro et al, 2011). The metagenomics-led seasonal study of Ace Lake showed that the change in season affected the relative abundance of *Chlorobium*, which was high in summer and spring (up to 80% in both seasons), low in winter (up to 6%), but lowest in Oct 2014 spring (<1%) (Chapter 3 section 3.3.4). This highlighted the reliance of *Chlorobium* on the availability of light to perform its primary production and growth.

In this thesis, the Ace Lake metagenomes and *Chlorobium* MAGs were used to assess genomic variation in these GSB to identify potential phylotypes and ecotypes in the lake (Chapter 5). The metagenomes and *Chlorobium* MAGs from Ace Lake, Ellis Fjord, and Taynaya Bay and the genome of *C. phaeovibrioides* DSM 265 (C-phaeov; Chapter 5) were also used to assess the genomic variation in the *Chlorobium* from the three systems and to assess their endemicity to the Vestfold Hills (Chapter 5). Similar to *Synechococcus* MAG genes, the auto-annotated *Chlorobium* MAG genes were parsed to identify defence genes, which were manually annotated to verify their gene function, and the defence capacity of the Vestfold Hills *Chlorobium* was explored (Chapter 5). The analysis of Ace Lake data from the Interface and different time periods (2008–2014) showed no mutations in *Chlorobium* MAGs, indicating that the genomic sequence of the Ace Lake *Chlorobium* was very stable and did not vary over time (Chapter 5 section 5.3.2.2). However, potential *Chlorobium* phylotypes with a modified capacity for cold adaptation, sodium ion export, and/or cell defence as well as potential ecotypes with the capacity to import iron, cobalt, vitamin B12, biosynthesize cobalamin, and/or salvage cobalamin precursors were identified in Ace Lake metagenomes from all time periods (Chapter 5 section 5.4.1). In *Chlorobium* species, it has been shown that cobalamin is linked to the improved production of bacteriochlorophyll and formation of chlorosomes, both of which are associated with the photosynthetic machinery of *Chlorobium* (Sato et al, 1981; Fuhrmann et al, 1993). The analyses in this thesis also showed a potential interaction between Ace Lake *Synechococcus* and the *Chlorobium*

subpopulation containing genes for salvaging cobalamin precursors, such as pseudocobalamin, which is known to be produced by cyanobacteria (Watanabe et al, 1999; Miyamoto et al, 2006; Watanabe et al, 2006; Watanabe et al, 2007). Together, these findings indicated the adaptation of *Chlorobium* to the Ace Lake environment and its ability to rise from very low abundance in winter to very high abundance in summer (Chapter 5 section 5.4.1).

The *Chlorobium* from Ace Lake, Ellis Fjord, and Taynaya Bay belonged to the same species and were closely related to C-phaeov (Chapter 5 section 5.3.4.1). The genomic composition and functional potential of the Vestfold Hills *Chlorobium* was also distinct from that of C-phaeov (Chapter 5 section 5.3.4.3). Moreover, the Vestfold Hills *Chlorobium* markers were different from the GSB markers in metagenomes and genomes from global sites (Chapter 5 section 5.4.2.2). Together, these findings indicated that the *Chlorobium* from the three systems might be endemic to the Vestfold Hills. Notably, similar *Chlorobium* phylotypes and ecotypes were prevalent in Ace Lake, Ellis Fjord, and Taynaya Bay, although their abundances varied in the three systems (Chapter 5 section 5.3.4.2). The *Chlorobium* from Ellis Fjord and Taynaya Bay were found to be more similar to each other than to Ace Lake *Chlorobium*, although Ace Lake and Taynaya Bay were located closer to each other (~2 km apart) than to Ellis Fjord (13–15 km apart). This might be related to the biogeographic partitioning of the three systems in the Vestfold Hills — both Ellis Fjord and Taynaya Bay are connected to the Southern Ocean by narrow water channels, whereas Ace Lake is isolated from the ocean and is landbound.

Considering that C-phaeov was the closest related species to the Vestfold Hills *Chlorobium*, a comparison of the Vestfold Hills *Chlorobium* with other *C. phaeovibrioides* strains isolated from various global sites could be used to further investigate the endemicity of this Antarctic *Chlorobium*. The genomes of five *C. phaeovibrioides* strains (one complete and four draft genomes), apart from the complete genome of C-phaeov, are available on NCBI and all five of them were isolated from stratified lakes in Russia (mostly from around the White Sea). Two medium-quality, 99% genome completeness *C. phaeovibrioides* MAGs (representing draft genomes) generated from metagenomes from Etoliko Lagoon in Greece are also available in the public database of JGI IMG. The *16S rRNA* gene- and BclA protein-based phylogenies, ANI, and AAI of these seven complete or draft genomes of *C. phaeovibrioides* and the

Vestfold Hills *Chlorobium* can be assessed using the methods applied in this thesis (Chapter 5 sections 5.2.4 and 5.2.7). Moreover, the alignment of the Vestfold Hills *Chlorobium* MAGs to these reference *C. phaeovibrioides* strains (from NCBI) and MAGs (from IMG) would help assess the similarities/differences in their genomic compositions and functional potentials; similar to the analysis performed in this thesis (Chapter 5 section 5.2.4). Together, these analyses can be used for further investigation of *Chlorobium* endemicity to the Vestfold Hills and obtain a more thorough understanding of its global presence. FR analysis of reference *C. phaeovibrioides* strains and MAGs using Antarctic metagenomes may or may not be useful, depending on the ANI of the reference genome to the Vestfold Hills *Chlorobium*. For example, the ANI of C-phaeov to the Vestfold Hills *Chlorobium* was <90%. Therefore, during FR analysis of C-phaeov in Ace Lake, Ellis Fjord, and Taynaya Bay metagenomes, the read alignment threshold had to be reduced to <90% in order to recruit sufficient number of reads to C-phaeov. However, this could have led to the recruitment of reads that were not associated with *Chlorobium*. Due to this reason, the FR of C-phaeov to the Antarctic metagenomes was not reported in this thesis.

6.5 The Vestfold Hills *Chlorobium* viruses and their biogeographic distribution in East Antarctica

Although most *Chlorobium* species contain CRISPR-Cas systems and other defence genes (Chapter 5 section 5.4.3), only a few *Chlorobium* viruses have been reported to date (Llorens–Marès et al, 2017; Berg et al, 2020). In this thesis, the viral and CRISPR spacer data from various Antarctic metagenomes were used to identify potential viruses of the Vestfold Hills *Chlorobium* (Chapters 3 and 5). At least three viral clusters and nine viral singletons representing potential *Chlorobium* viruses were identified. These *Chlorobium* viruses had a broad range of hosts, including not only *Chlorobium* but also some members of *Gammaproteobacteria*, *Actinobacteria*, *Betaproteobacteria*, *Deltaproteobacteria*, *Firmicutes*, *Flavobacteriia*, and *Verrucomicrobia*. The Vestfold Hills *Chlorobium* contained a number of genes associated with intracellular defence systems, such as a subtype I-E CRISPR-Cas system, type I and IV R-M systems, and a AbiE type IV T-A system involved in ABI mechanism for viral infection disruption (Chapter 5 section 5.4.3). However, it did not contain any known extracellular defences, such as the ones observed in subpopulations of Ace Lake *Synechococcus*, i.e., mutations

in membrane proteins that might allow for changes in cell surface structure and help the host cell to evade viruses that attach to specific cell surface receptors (Avrani et al, 2011; Schmid et al, 2016; Tschitschko et al, 2015; Tschitschko et al, 2018; Zborowskya and Lindell, 2019). Together, the findings indicated that these *Chlorobium* viruses were probably generalist viruses that were neutralised by *Chlorobium* using intracellular defence systems, and that the Vestfold Hills *Chlorobium* did not have any specialist viruses.

It has been speculated that viruses usually evolve into specialist viruses when they are exposed to a homogenous host population (e.g., composed of a single species) that does not change with time, whereas generalist viruses can evolve from viruses exposed to a heterogenous host population (e.g., composed of multiple species) that fluctuates with time (Elena et al, 2009). The adaptation of a specialist virus to a single host, which would allow it to effectively replicate in the specific host, can have fitness costs on its replication in other potential hosts, whereas a generalist virus usually incurs no fitness costs on replication in different hosts (Elena et al, 2009). In Ace Lake, there appeared to be no evidence of any *Chlorobium*-specific viruses, rather the potential *Chlorobium* viruses identified in the lake had a broad host range (Chapter 3 section 3.3.5.6). Although the Ace Lake *Chlorobium* population was homogenous (Chapter 5 section 5.3.4.1), its relative abundance fluctuated with change in season — Nov 2013 summer (33%), Jul and Aug 2014 winter (<6%), Oct 2014 spring (<1%), and Dec 2014 (59%) (Chapter 3 section 3.3.4). Together, these ideas and observations might indicate that the Ace Lake *Chlorobium* lacks specialist viruses due to seasonal changes in its abundance. This lack of viral predation by *Chlorobium*-specific viruses, in turn, might be contributing toward the very high abundance of *Chlorobium* in the Interface of Ace Lake. Moreover, the drop in *Chlorobium* abundance in winter was inferred to be related to the low availability of light rather than viral predation. Therefore, this seasonal die-off of *Chlorobium* population due to its reliance on light might be beneficial to *Chlorobium*, considering that it restricts the ability of specialist viruses to establish effective lifecycles in this host.

The Vestfold Hills *Chlorobium* viruses were identified in aquatic systems from the Vestfold Hills and the Rauer Islands, indicating their wide-spread distribution in East Antarctica (Chapter 5 section 5.4.4). Interestingly, other than Ace Lake, Ellis Fjord, and Taynaya Bay, none of the other aquatic systems in which these viruses were identified,

namely Deep Lake, Club Lake, Organic Lake, and Rauer 2, 3, 5, 6, 11, and 13 lakes, contained *Chlorobium* (Bowman et al, 2000a; DeMaere et al, 2013; Yau et al, 2013; Tschitschko et al, 2018 unpublished data). However, these systems did contain some of the other potential hosts of the Vestfold Hills *Chlorobium* viruses, like the members of *Gammaproteobacteria*, *Bacteroidetes*, *Actinobacteria*, *Betaproteobacteria*, *Firmicutes*, *Deltaproteobacteria,* and *Verrucomicrobia* (Bowman et al, 2000a; DeMaere et al, 2013; Yau et al, 2013; Tschitschko et al, 2018 unpublished data). This highlighted the broad host range of these viruses, which could predate on bacteria other than *Chlorobium*.

6.6 The importance of manual annotation in the era of high-throughput functional auto-annotation

The precise annotation of genes is important to predict and understand their biological functions in the systems in which they are identified. With the advent of HTS techniques that allow for parallel sequencing of large datasets like metagenomes, it has become important to have annotation pipelines that can automatically assign functions to the genes identified in these sizeable datasets. For metagenomes and genomes, the structural annotation pipeline used by JGI's IMG system includes the identification of protein coding genes, non-coding RNA genes, and CRISPR spacers and repeats, whereas their functional annotation pipeline includes assignment of COG numbers, KO terms, EC numbers, and Pfams to protein coding genes (Huntemann et al, 2015). The latest version (v5.0.0) of the functional annotation pipeline used by IMG system also includes assigning Cath-Funfam, SuperFamily, SMART, and TIGRFAMs to protein coding genes from metagenomes (https://img.jgi.doe.gov/docs/pipelineV5/).

In this thesis, the manual annotation of the genes under study was performed to verify their functional assignment, for precise prediction of their biological roles (Chapters 3, 4, and 5). For this purpose, the protein sequences of the genes were compared with the reference proteins in the UniProtKB/Swiss-Prot database, which is a manually annotated and curated database of protein sequences (Boutet et al, 2016). The protein matches were further explored to manually assess domain matches to ensure that the protein was capable of performing the predicted functions, and to check the matching reference protein for evidence at protein level (rather than inferred from sequence homology). For query proteins that had low identity or no matches in UniProtKB/Swiss-

Prot database were aligned against the UniProtKB or RefSeq databases, and the protein matches were manually analysed in search of domain matches.

While the auto-annotations of the manually parsed genes were found to be mostly correct, the analysis did reveal some inaccurate annotations. For example, an Ace Lake *Synechococcus* gene auto-annotated as bisphosphoglycerate-independent phosphoglycerate mutase (AlkP superfamily) was manually annotated as a BREX gene coding for PglZ domain-containing protein. This reannotation was also supported by the presence of a *brxL* gene adjacent to this *pglZ* gene, since *pglZ* and *brxL* are usually found together and are known to be co-transcribed (Goldfarb et al, 2015). Similarly, a *Synechococcus* gene auto-annotated as hypothetical was manually annotated as *brxC*, a BREX system gene. This *Synechococcus* gene was identified in a defence gene island, with *brxA* and *brxB* upstream of its location and some T-A system genes and a *pglX* gene downstream of it (Chapter 4 Figure 4.5). In Ace Lake *Chlorobium*, a gene auto-annotated as cobaltochelatase subunit N was manually reannotated as magnesium chelatase subunit H. This *Chlorobium* gene was located adjacent to two other magnesium chelatase genes coding for subunits D and I, which supported its reannotated function. A number of *Chlorobium* substrate transport genes were also manually reannotated based on their matches to reference proteins or specific domains on the reference proteins (Chapter 5 Table 5.10). Overall, the manual analysis and verification of gene functions were found to be essential for precise annotations of the genes and helped in confidently predicting their biological roles.

6.7 Concluding remarks

The use of metagenomes for the analyses described in this thesis have allowed for a thorough assessment of Ace Lake from various perspectives. The Cavlab pipeline used for the taxonomic, abundance, and functional analyses of Ace Lake, as well as Ellis Fjord and Taynaya Bay, can be used for similar evaluation of any metagenomes annotated by JGI's IMG system. The pipeline can also be easily upgraded to improve existing methods or to include better methods or analyses, which would allow for improved metagenomic analysis. The study in this thesis has expanded our knowledge of the Ace Lake microbial community — the repertoire of abundant microbes in the lake, their niche adaptation and functional potential, the effects of change in season on

their abundances, the viral population in the lake, the phylotypes and ecotypes of the two key bacteria (*Chlorobium* and *Synechococcus*) in the lake, and the endemicity of Ace Lake *Chlorobium* to the Vestfold Hills. With the availability of a time-series of metagenomes and MAGs from Ace Lake, good opportunities exist to further advance the understanding of microbes in this lake system. For example, the 'unassigned contigs' determined from Ace Lake metagenomes need to be re-evaluated to identify their contributions to the lake diversity and function. The complete genomes of the viruses identified in Ace Lake and other Antarctic metagenomes (complete phage catalogue; Chapter 3 section 3.2.6) would be very useful in assessing the overall viral population and function in the Antarctic systems. The data in the Antarctic virus catalogue and spacer database (Chapter 3 section 3.2.6) can be used to assess virus-host interactions in Antarctic systems. With the use of *C. phaeovibrioides* strain genomes (NCBI) and MAGs (IMG) isolated/generated from various stratified lakes from across the globe, the endemicity of Ace Lake *Chlorobium* to the Vestfold Hills can be comprehensively explored.

References

AASSP. Australian Commonwealth Government Australian Antarctic Science Strategic Plan 2011-12 to 2020-21. Australian Commonwealth Government, Barton. 2011.

Abedon ST. Bacterial 'immunity' against bacteriophages. Bacteriophage. 2012; 2:50–4.

Achberger AM, Christner BC, Michaud AB, Priscu JC, Skidmore ML, Vick-Majors TJ, the WISSARD Science Team. Microbial community structure of subglacial Lake Whillans, West Antarctica. Frontiers in Microbiology. 2016;7:1457.

Ahlgren NA, Rocap G. Culture isolation and culture-independent clone libraries reveal new marine *Synechococcus* ecotypes with distinctive light and N physiologies. Applied and Environmental Microbiology. 2006;72:7193–204.

Ahlgren NA, Rocap G. Diversity and distribution of marine *Synechococcus*: multiple gene phylogenies for consensus classification and development of qPCR assays for sensitive measurement of clades in the ocean. Frontiers in Microbiology. 2012;3:213.

Al-Shayeb B, Sachdeva R, Chen LX, Ward F, Munk P, Devoto A, Castelle CJ, Olm MR, Bouma-Gregson K, Amano Y, He C, Méheust R, Brooks B, Thomas A, Lavy A, Matheus-Carnevali P, Sun C, Goltsman DSA, Borton MA, Sharrar A, Jaffe AL, Nelson TC, Kantor R, Keren R, Lane KR, Farag IF, Lei S, Finstad K, Amundson R, Anantharaman K, Zhou J, Probst AJ, Power ME, Tringe SG, Li WJ, Wrighton K, Harrison S, Morowitz M, Relman DA, Doudna JA, Lehours AC, Warren L, Cate JHD, Santini JH, Banfield JF. Clades of huge phages from across Earth's ecosystems. Nature. 2020;578:425–31.

Alexander B, Andersen JH, Cox RP, Imhoff JF. Phylogeny of green sulfur bacteria on the basis of gene sequences of16S rRNA and of the Fenna–Matthews–Olson protein. Archives of Microbiology. 2002;178:131–40.

Alexander B, Imhoff JF. Communities of green sulfur bacteria in different marine and saline habitats analyzed by gene sequences of 16S rRNA and of the Fenna–Matthews–Olson protein. International Microbiology. 2006;9:259–66.

Alin SR, Johnson TC. Carbon cycling in large lakes of the world: A synthesis of production, burial, and lake-atmosphere exchange estimates. Global Biogeochemical Cycles. 2007;21:GB3002.

Allen MA, Lauro FM, Williams TJ, Burg D, Siddiqui KS, De Francisci D, Chong KWY, Pilak O, Chew HH, De Maere MA, Ting L, Katrib M, Ng C, Sowers KR, Galperin MY, Anderson IJ, Ivanova N, Dalin E, Martinez M, Lapidus A, Hauser L, Land M, Thomas T, Cavicchioli R. The genome sequence of the psychrophilic archaeon, *Methanococcoides burtonii*: the role of genome evolution in cold adaptation. The ISME Journal. 2009;3:1012–35.

Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ. Basic Local Alignment Search Tool. Journal of Molecular Biology. 1990;215:403–10.

Altschul SF, Madden TL, Schäffer AA, Zhang J, Zhang Z, Miller W, Lipman D. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Research. 1997;25:3389–402.

Anderson PJ, Lango J, Carkeet C, Britten A, Kräutler B, Hammock BD, Roth JR. One pathway can incorporate either adenine or dimethylbenzimidazole as an α-axial ligand of B12 cofactors in *Salmonella enterica*. Journal of Bacteriology. 2008;190:1160–71.

Anesio AM, Bellas CM. Are low temperature habitats hot spots of microbial evolution driven by viruses? Trends in Microbiology. 2011;19:52–7.

Archer SD, McDonald IR, Herbold CW, Cary SC. Characterisation of bacterioplankton communities in the meltwater ponds of Bratina Island, Victoria Land, Antarctica. FEMS Microbiology Ecology. 2014;89:451–64.

Arenas FA, Pugin B, Henríquez NA, Arenas-Salinas MA, Díaz-Vásquez WA, Pozo MF, Muñoz CM, Chasteen TG, Pérez-Donoso JM, Vásquez CC. Isolation, identification and characterization of highly tellurite-resistant, tellurite-reducing bacteria from Antarctica. Polar Science. 2014;8:40–52.

Avrani S, Wurtzel O, Sharon I, Sorek R, Lindell D. Genomic island variability facilitates *Prochlorococcus*-virus coexistence. Nature. 2011;474:604–8.

Bağci C, Beier S, Górska A, Huson DH. Introduction to the Analysis of Environmental Sequences: Metagenomics with MEGAN. In: Anisimova M (eds). Evolutionary Genomics. Methods in Molecular Biology. Humana, New York. 2019;1910.

Bak F, Cypionka H. A novel type of energy metabolism involving fermentation of inorganic sulphur compounds. Nature. 1987;326:891–2.

Baker AL, Kromer Baker K, Tyler PA. Fine-layer depth relationships of lake water chemistry, planktonic algae and photosynthetic bacteria in meromictic Lake Fidler, Tasmania. Freshwater Biology. 1985;15:735–47.

Barker R. Physical and chemical parameters of Deep Lake, Vestfold Hills, Antarctica. Australian National Antarctic Research Expeditions Series. 1981;B(V) Limnology Publication NO. 130.

Barrangou R, Fremaux C, Deveau H, Richards M, Boyaval P, Moineau S, Romero DA, Horvath P. CRISPR provides acquired resistance against viruses in prokaryotes. Science. 2007;315:1709–12.

Bastviken D, Cole JJ, Pace ML, Van de Bogert MC. Fates of methane from different lake habitats: Connecting whole-lake budgets and CH_4 emissions. Journal of Geophysical Research. 2008;113:G02024.

Beatty JT, Overmann J, Lince MT, Manske AK, Lang AS, Blankenship RE, Van Dover CL, Martinson TA, Plumley FG. An obligately photosynthetic bacterial anaerobe from a deep-sea hydrothermal vent. PNAS. 2005;102:9306–10.

Bell EM. Plankton dynamics in the saline lakes of the Vestfold Hills, Eastern Antarctica. PhD thesis, University of Nottingham; 1998.

Bell EM, Laybourn-Parry J. Annual plankton dynamics in an Antarctic saline lake. Freshwater Biology. 1999;41:507–19.

Bell EM, Laybourn-Parry J. Mixotrophy in the Antarctic phytoflagellate, *Pyramimonas gelidicola* (Chlorophyta: Prasinophyceae). Journal of Phycology. 2003;39:644–9.

Berg M, Goudeau D, Olmsted C, McMahon KD, Thweatt J, Bryant D, Eloe-Fadrosh EA, Malmstrom RR, Roux S. Host population diversity as a driver of viral infection cycle in wild populations of green sulfur bacteria with long standing virus-host interactions. BioRxiv. 2020.

Bernhard A. The nitrogen cycle: processes, players, and human impact. Nature Education Knowledge. 2010;3:25.

Bikard D, Marraffini LA. Innate and adaptive immunity in bacteria: mechanisms of programmed genetic variation to fight bacteriophages. Current Opinion in Immunology. 2012;24:15–20.

Blankenship RE, Matsuura K. Antenna complexes from green photosynthetic bacteria. In: Green BR, Parson WW (eds). Light-harvesting antennas in photosynthesis. Advances in Photosynthesis and Respiration. 2003;13:195–217.

Boldrin F, Ventura M, Degiacomi G, Ravishankar S, Sala C, Svetlikova Z, Ambady A, Dhar N, Kordulakova J, Zhang M, Serafini A, Vishwas VG, Kolly GS, Kumar N,Palù G, Guerin ME, Mikusova K, Cole ST, Manganelli R. The phosphatidyl-myo-inositol mannosyltransferase PimA is essential for *Mycobacterium tuberculosis* growth *in vitro* and *in vivo*. Journal of Bacteriology. 2014;196:3441–51.

Boldyreva D, Babenko VV, Kanygina AV, Lunina ON, Letarova MA, Kostryukova ES, Savvichev AS, Gorlenko VM, Letarov AV. Genome sequences of a green-colored *Chlorobium phaeovibrioides* strain containing two plasmids and a closely related plasmid-free brown-colored strain. Microbiology Resource Announcement. 2020;9:e01172–19.

Bolger AM, Lohse M, Usadel B. Trimmomatic: a flexible trimmer for Illumina sequence data. Bioinformatics. 2014;30:2114–20.

Boumann HA, Longo ML, Stroeve P, Poolman B, Hopmans EC, Stuart MCA, Damsté JSS, Schouten S. Biophysical properties of membrane lipids of anammox bacteria: I. Ladderane phospholipids form highly organized fluid membranes. Biochimica et Biophysica Acta. 2009;1788:1444–51.

Boutet E, Lieberherr D, Tognolli M, Schneider M, Bansal P, Bridge AJ, Poux S, Bougueleret L, Xenarios I. UniProtKB/Swiss-Prot, the manually annotated section of the UniProt KnowledgeBase: How to use the entry view. Methods in Molecular Biology. 2016;1374:23–54.

Bowman JP, McCammon SA, Rea SM, McMeekin TA. The microbial composition of three limnologically disparate hypersaline Antarctic lakes. FEMS Microbiology Letters 2000a;183:81-88.

Bowman JP, McCammon SA, Skerratt JH. *Methylosphaera hansonii* gen. nov., sp. nov., a psychrophilic, group I methanotroph from Antarctic marine-salinity, meromictic lakes. Microbiology. 1997;143:1451–9.

Bowman JP, Rea SM, McCammon SA, McMeekin TA. Diversity and community structure within anoxic sediment from marine salinity meromictic lakes and a coastal meromictic marine basin, Vestfold Hills, Eastern Antarctica. Environmental Microbiology. 2000b;2:227–37.

Brenner DJ. Deoxyribonucleic acid reassociation in the taxonomy of enteric bacteria. International Journal of Systematic Bacteriology. 1973;23:298–307.

Bridge PD, Spooner BM, Roberts PJ. Non-lichenized fungi from the Antarctic region. Mycotaxon. 2008;106:485–90.

Bryant DA, Frigaard N. Prokaryotic photosynthesis and phototrophy illuminated. Trends in Microbiology. 2006;14:488–96.

Buchanan BB, Arnon DI. A reverse Krebs cycle in photosynthesis: consensus at last. Photosynthesis Research. 1990;24:47–53.

Buchfink B, Xie C, Huson DH. Fast and sensitive protein alignment using DIAMOND. Nature Methods. 2014;12:59–60.

Burch MD. Annual cycle of phytoplankton in Ace Lake, an ice covered, saline meromictic lake. Hydrobiologia. 1988;165:59–75.

Burke CM, Burton HR. Photosynthetic bacteria in meromictic lakes and stratified fjords of the Vestfold Hills, Antarctica. Hydrobiologia. 1988a;165:13–23.

Burke CM, Burton HR. The ecology of photosynthetic bacteria in Burton Lake, Vestfold Hills, Antarctica. Hydrobiologia. 1988b;165:1–11.

Burton HR. Methane in a Saline Antarctic Lake. In: Trudinger PA, Walter MR, Ralph BJ (eds). Biogeochemistry of Ancient and Modern Environments. Springer, Berlin, Heidelberg. 1980;243–51.

Burton HR, Barker RJ. Sulfur chemistry and microbiological fractionation of sulfur isotopes in a saline Antarctic lake. Geomicrobiology Journal. 1979;1:329–40.

Cadieux N, Bradbeer C, Reeger-Schneider E, Köster W, Mohanty AK, Wiener MC, Kadner RJ. Identification of the periplasmic cobalamin-binding protein BtuF of *Escherichia coli*. Journal of Bacteriology. 2002;184:706–17.

Cai F, Axen SD, Kerfeld CA. Evidence for the widespread distribution of CRISPR-Cas system in the phylum Cyanobacteria. RNA Biology. 2013;10:687–93.

Callieri C, Slabakova V, Dzhembekova N, Slabakova N, Peneva E, Cabello-Yeves PJ, Cesare AD, Eckert EM, Bertoni RB, Corno G, Salcher MM, Kamburska L, Bertoni F, Moncheva S. The mesopelagic anoxic Black Sea as an unexpected habitat for *Synechococcus* challenges our understanding of global "deep red fluorescence". The ISME Journal. 2019;13:1676–87.

Camacho C, Coulouris G, Avagyan V, Ma N, Papadopoulos J, Bealer K, Madden TL. BLAST+: architecture and applications. BMC Bioinformatics. 2009;10:421.

Campbell PJ. Primary productivity of a hypersaline Antarctic lake. Australian Journal of Marine and Freshwater Research. 1978;29:717–24.

Canfield DE, Green WJ. The cycling of nutrients in a closed-basin Antarctic lake: Lake Vanda. Biogeochemistry. 1985;1:233–56.

Cary SC, McDonald IR, Barrett JE, Cowan DA. On the rocks: the microbiology of Antarctic Dry Valley soils. Nature Reviews Microbiology. 2010;8:129–38.

Castenholz RW, Bauld J, Jørgenson BB. Anoxygenic microbial mats of hot springs: thermophilic *Chlorobium* sp. FEMS Microbiology Ecology. 1990;74:325–36.

Caumette P. Distribution and characterization of phototrophic bacteria isolated from the water of Bietri Bay (Ebrie Lagoon, Ivory Coast). Canadian Journal of Microbiology. 1984;30:273–84.

Cavicchioli R. Microbial ecology of Antarctic aquatic systems. Nature Reviews Microbiology. 2015;13:691-706.

Cerdeño-Tárraga AM, Patrick S, Crossman LC, Blakely G, Abratt V, Lennard N, Poxton I, Duerden B, Harris B, Quail MA, Barron A, Clark L, Corton C, Doggett

J, Holden MT, Larke N, Line A, Lord A, Norbertczak H, Ormond D, Price C, Rabbinowitsch E, Woodward J, Barrell B, Parkhill J. Extensive DNA inversions in the B. fragilis genome control variable gene expression. Science. 2005;307:1463–5.

Cesare AD, Dzhembekova N, Cabello-Yeves PJ, Eckert EM, Slabakova V, Slabakova N, Peneva E, Bertoni R, Corno G, Salcher MM, Kamburska L, Bertoni F, Rodriguez-Valera F, Moncheva S, Callieri C. Genomic comparison and spatial distribution of different *Synechococcus* phylotypes in the Black Sea. Frontiers in Microbiology. 2020;11:1979.

Chamot D, Magee WC, Yu E, Owttrim GW. A cold-shock induced cyanobacterial RNA helicase. Journal of Bacteriology. 1999;181:1728–32.

Chan JZM, Halachev MR, Loman NJ, Constantinidou C, Pallen MJ. Defining bacterial species in the genomic era: insights from the genus *Acinetobacter*. BMC Microbiology. 2012;12:302.

Chapin B, Denoyelles F, Gaham DW, Smith VH. A deep maximum of green sulphur bacteria (*'Chlorochromatium aggregatum'*) in a strongly stratified reservoir. Freshwater Biology. 2004;49:1337–54.

Chasteen TG, Fuentes DE, Tantaleán JC, Vásquez CC. Tellurite: history, oxidative stress, and molecular mechanisms of resistance. FEMS Microbiology Reviews. 2009;33:820–32.

Cheng J, Poduska B, Morton RA, Finan TM. An ABC-type cobalt transport system is essential for growth of *Sinorhizobium meliloti* at trace metal concentrations. Journal of Bacteriology. 2011;193:4405–16.

Chown SL, Clarke A, Fraser CI, Cary SC, Moon KL, McGeoch MA. The changing form of Antarctic biodiversity. Nature. 2015;522:431–8.

Coleman ML, Sullivan MB, Martiny AC, Steglich C, Barry K, Delong EF, Chisholm SW. Genomic islands and the ecology and evolution of *Prochlorococcus*. Science. 2006;311:1768–70.

Coolen MJL, Hopmans EC, Rijpstra WIC, Muyzer G, Schouten S, Volkman JK, Damsté JSS. Evolution of the methane cycle in Ace Lake (Antarctica) during the

Holocene: response of methanogens and methanotrophs to environmental change. Organic Geochemistry. 2004a;35:1151–67.

Coolen MJL, Muyzer G, Rijpstra WIC, Schouten S, Volkman JK, Damsté JSS. Combined DNA and lipid analyses of sediments reveal changes in Holocene haptophyte and diatom populations in an Antarctic lake. Earth and Planetary Science Letters. 2004b;223:225–39.

Coolen MJL, Muyzer G, Schouten S, Volkman JK, Damsté JSS. Sulfur and methane cycling during the Holocene in Ace Lake (Antarctica) revealed by lipid and DNA stratigraphy. In: Neretin L (eds). Past and Present Water Column Anoxia. NATO Science Series: IV: Earth and Environmental Sciences. Springer, Dordrecht. 2006; 64:41–65.

Comeau AM, Harding T, Galand PE, Vincent WF, Lovejoy C. Vertical distribution of microbial communities in a perennially stratified Arctic lake with saline, anoxic bottom waters. Science Report. 2012;2:604.

Cromer L, Gibson JAE, Swadling KM, Ritz DA. Faunal microfossils: Indicators of Holocene ecological change in a saline Antarctic lake. Palaeogeography, Palaeoclimatology, Palaeoecology. 2005;221:83–97.

Crouzet J, Cameron B, Cauchois L, Rigault S, Blanche F, Guilhot C, Levy-schil S, Rouyez MC. Genetic and sequence analyses of a *Pseudomonas denitrificans* DNA fragment containing two cob genes. Journal of Bacteriology. 1991;173:6058–65.

Culver DA, Brunskill GJ. Fayetteville Green Lake, New York. V. Studies of primary production and zooplankton in a meromictic marl lake. Limnology and Oceanography 1969;14:862–73.

Cuvelier ML, Guo J, Ortiz AC, Van Baren MJ, Tariq MA, Partensky F, Worden AZ. Responses of the picoprasinophyte *Micromonas commoda* to light and ultraviolet stress. PLoS One. 2017;12:e0172135.

Darling AE, Jospin G, Lowe E, Matsen IV FA, Bik HM, Eisen JA. PhyloSift: phylogenetic analysis of genomes and metagenomes. PeerJ. 2014;2:e243.

Darling ACE, Mau B, Blattner FR, Perna NT. Mauve: multiple alignment of conserved genomic sequence with rearrangements. Genome Research. 2004;14:1394-403.

Darling AE, Mau B, Perna NT. progressiveMauve: multiple genome alignment with gene gain, loss and rearrangement. PLoS One. 2010;5:e11147.

Dartnall HJG. A Limnological Reconnaissance of the Vestfold Hills. ANARE reports. 2000;141.

Debussche L, Couder M, Thibaut D, Cameron B, Crouzet J, Blanche F. Assay, purification, and characterization of cobaltochelatase, a unique complex enzyme catalyzing cobalt insertion in hydrogenobyrinic acid a,c-diamide during coenzyme B12 biosynthesis in *Pseudomonas denitrificans*. Journal of Bacteriology. 1992;174:7445–51.

DeMaere MZ, Williams TJ, Allen MA, Brown MV, Gibson JAE, Rich J, Lauro FM, Dyall-Smith M, Davenport KW, Woyke T, Kyrpides NC, Tringe SG, Cavicchioli R. High level of intergenera gene exchange shapes the evolution of haloarchaea in an isolated Antarctic lake. PNAS. 2013;110:16939–44.

Devol AH. Nitrogen cycle: Solution to a marine mystery. Nature. 2003. 422:575–6.

Dibrova DV, Galperin MY, Mulkidjanian AY. Characterization of the N-ATPase, a distinct, laterally transferred Na^+-translocating form of the bacterial F-type membrane ATPase. Bioinformatics. 2010;26:1473–6.

Dirix G, Monsieurs P, Dombrecht B, Daniels R, Marchal K, Vanderleyden J, Michiels J. Peptide signal molecules and bacteriocins in Gram-negative bacteria: a genome-wide in silico screening for peptides containing a double-glycine leader sequence and their cognate transporters. Peptides. 2004;25:1425–40.

Drewry DJ. Antarctica: Glaciological and Geophysical Folio. Polar Research Institute, University of Cambridge. 1983.

Dy RL, Przybilski R, Semeijn K, Salmond GPC, Fineran PC. A widespread bacteriophage abortive infection system functions through a Type IV toxin–antitoxin mechanism. Nucleic Acids Research. 2014;42:4590–605.

Elena SF, Agudelo-Romero P, Lalić J. The evolution of viruses in multi-host fitness landscapes. Open Virology Journal. 2009;3:1–6.

Eisen JA, Nelson KE, Paulsen IT, Heidelberg JF, Wu M, Dodson RJ, Deboy R, Gwinn ML, Nelson WC, Haft DH, Hickey EK, Peterson JD, Durkin AS, Kolonay JL,

Yang F, Holt I, Umayam LA, Mason T, Brenner M, Shea TP, Parksey D, Nierman WC, Feldblyum TV, Hansen CL, Craven MB, Radune D, Vamathevan J, Khouri H, White O, Gruber TM, Ketchum KA, Venter JC, Tettelin H, Bryant DA, Fraser CM. The complete genome sequence of *Chlorobium tepidum* TLS, a photosynthetic, anaerobic, green-sulfur bacterium. Proc Natl Acad Sci U S A. 2002;99:9509–14.

Engelberg-Kulka H, Hazan R, Amitai S. mazEF: a chromosomal toxin-antitoxin module that triggers programmed cell death in bacteria. Journal of Cell Science. 2005;118:4327–32.

Ferris JM, Burton HR. The annual cycle of heat content and mechanical stability of hypersaline Deep Lake, Vestfold Hills, Antarctica. Hydrobiologia. 1988;165:115–28.

Fierer N, Leff JW, Adams BJ, Nielsen UN, Bates ST, Lauber CL, Owens S, Gilbert JA, Wall DH, Caporaso JG. Cross-biome metagenomic analyses of soil microbial communities and their functional attributes. PNAS. 2012;109:21390–5.

Finster KW, Kjeldsen KU, Kube M, Reinhardt R, Mussmann M, Amann R, Schreiber L. Complete genome sequence of *Desulfocapsa sulfexigens*, a marine deltaproteobacterium specialized in disproportionating inorganic sulfur compounds. Standards in Genomic Sciences. 2013;8:58–68.

Forchhammer K. Glutamine signalling in bacteria. Frontiers in Bioscience. 2007;12:358–70.

Francis CA, Beman JM, Kuypers MMM. New processes and players in the nitrogen cycle: the microbial ecology of anaerobic and archaeal ammonia oxidation. The ISME Journal. 2007;1:19–27.

Frank S, Brindley AA, Deery E, Heathcote P, Lawrence AD, Leech HK, Pickersgill RW, Warren MJ. Anaerobic synthesis of vitamin B12: Characterization of the early steps in the pathway. Biochemical Society Transactions. 2005;33:811–4.

Franzmann PD, Deprez PP, Burton HR, Van den Hoff J. Limnology of Organic Lake, Antarctica, a meromictic lake that contains high concentrations of dimethyl sulfide. Australian Journal of Freshwater Research. 1987;38:409–17.

Franrmann PD and Dobson SJ. Cell wall-less, free living Spirochctes in Antarctica. FEMS Microbiology Letters. 1992;97:289–92.

Franzmann PD, Hopfl P, Weiss N, Tindall BJ. Psychrotrophic, lactic acid-producing bacteria from anoxic waters in Ace Lake; *Carnobacterium funditum* sp. nov. and *Carnobacterium alterfutulitum* sp. nov. Archives of Microbiology. 1991a;156:255–62.

Franzmann PD, Liu Y, Balkwill D, Aldrich HC, De Macario EC, Boone DR. *Methanogenium frigidum* sp. nov., a psychrophilic, H^2-using methanogen from Ace Lake, Antarctica. International Journal of Systematic Bacteriology. 1997;47:1068–72.

Franrmann PD, Rohde M. An obligately anaerobic, coiled bacterium from Ace Lake, Antarctica. Journal of General Microbiology. 1991;137:2191–6.

Franzmann PD, Roberts NJ, Mancuso VA, Burton HR, McMeekin TA. Methane production in meromictic Ace Lake, Antarctica. Hydrobiologia. 1991b; 210:191–201.

Franzmann PD, Skyring GW, Burton HR, Deprez PP. Sulfate reduction rates and some aspects of the limnology of four lakes and a fjord in the Vestfold Hills, Antarctica. Hydrobiologia. 1988;165:25–33.

Franzmann PD, Springer N, Ludwig W, Conway de Macario E, Rohde M. A methanogenic archaeon from Ace Lake, Antarctica: *Methanococcoides burtonii* sp. nov. System Applied Microbiology. 1992;15:573–81.

Frias-Lopez J, Shi Y, Tyson GW, Coleman ML, Schuster SC, Chisholm SW, Delong EF. Microbial community gene expression in ocean surface waters. PNAS. 2008;105:3805–10.

Frigaard NU, Bryant DA. Genomic insights into the sulfur metabolism of phototrophic green sulfur bacteria. In: Hell R, Dahl C, Knaff D, Leustek T (eds). Sulfur metabolism in phototrophic organisms. Advances in Photosynthesis and Respiration. Springer, Dordrecht. 2008;27:337–55.

Fuhrmann S, Overmann J, Pfennig N, Fischer U. Influence of vitamin B12 and light on the formation of chlorosomes in green- and brown-colored *Chlorobium* species. Archives of Microbiology. 1993;160:193–8.

Gallagher JB, Burton HR. Seasonal mixing of Ellis Fjord, Vestfold Hills, East Antarctica. Estuarine, Coastal and Shelf Science. 1988;27:363–80.

Gallagher JB, Burton HR, Calf GE. Meromixis in an Antarctic fjord: a precursor to meromictic lakes on an isostatically rising coastline. Hydrobiologia. 1989;172:235–54.

Galperin MY, Makarova KS, Wolf YI, Koonin EV. Expanded microbial genome coverage and improved protein family annotation in the COG database. Nucleic Acids Res. 2015;43: D261–9.

Garcia-Dominguez M, Reyes JC, Florencio JF. Purification and characterization of a new type of glutamine synthetase from cyanobacteria. European Journal of Biochemistry. 1997;244:258–64.

Garcia-Gil LJ, Abellà CA. Population dynamics of phototrophic bacteria in three basins of Lake Banyoles (Spain). Hydrobiologia. 1992;243:87–94.

Gaufichon L, Reisdorf-Cren M, Rothstein SJ, Chardon F, Suzuki A. Biological functions of asparagine synthetase in plants. Plant Science. 2010;179:141–53.

Gerdes K, Christensen SK, Lobner-Olesen A. Prokaryotic toxin-antitoxin stress response loci. Nature Reviews Microbiology. 2005;3:371–82.

Gibson JAE. The meromictic lakes and stratified marine basins of the Vestfold Hills, East Antarctica. Antarctic Science. 1999;11:175–92.

Gibson JAE, Burton HR. Meromictic Antarctic lakes as recorders of climate change: the structures of Ace and Organic lakes, Vestfold hills, Antarctica. Papers and Proceedings of the Royal Society of Tasmania. 1996;130:73–8.

Gibson J, Pfennig N, Waterbury JB. *Chloroherpeton thalassium* gen. nov. et spec. nov., a non-filamentous, flexing and gliding green sulfur bacterium. Archives of Microbiology. 1984;138:96–101.

Gibson JAE, Swadling KM, Pitman TM, Burton HR. Over-wintering populations of *Mesodinium rubrum* (Ciliophora: Haptorida) in lakes of the Vestfold Hills, Antarctica. Polar Biology. 1997;17:175–9.

Goldfarb T, Sberro H, Weinstock E, Cohen O, Doron S, Charpak-Amikam Y, Afik S, Ofir G, Sorek R. BREX is a novel phage resistance system widespread in microbial genomes. EMBO J. 2015; 34:169–83.

Gomez P, Buckling A. Bacteria-phage antagonistic coevolution in soil. Science. 2011;332:106–9.

Gordon DA, Priscu J, Giovannoni S. Origin and phylogeny of microbes living in permanent Antarctic lake ice. Microbial Ecology. 2000;39:197–202.

Goris J, Konstantinidis KT, Klappenbach JA, Coenye T, Vandamme P, Tiedje JM. DNA-DNA hybridization values and their relationship to whole-genome sequence similarities. International Journal of Systematic and Evolutionary Microbiology. 2007;57:81–91.

Grada A, Weinbrecht K. Next-generation sequencing: methodology and application. The Journal of Investigative Dermatology. 2013;133:e11.

Gray MJ, Escalante-Semerena JC. The cobinamide amidohydrolase (cobyric acid-forming) CbiZ enzyme: A critical activity of the cobamide remodeling system of *Rhodobacter sphaeroides*. Molecular Microbiology. 2009;74:1198–210.

Gray MJ, Tavares NK, Escalante-Semerena JC. The genome of *Rhodobacter Sphaeroides* strain 2.4.1 encodes functional cobinamide salvaging systems of archaeal and bacterial origins. Molecular Microbiology. 2008;70:824–36.

Gregersen LH, Bryant DA, Frigaard NU. Mechanisms and evolution of oxidative sulfur metabolism in green sulfur bacteria. Frontiers in Microbiology. 2011;2:116.

Grouzdev DS, Lunina ON, Gaisin VA, Krutkina MS, Baslerov RV, Savvichev AS, Gorlenko VM. Genome sequences of green- and brown-colored strains of *Chlorobium phaeovibrioides* with gas vesicles. Microbiology Resource Announcements. 2019;8:e00711–19.

Grzymski JJ, Riesenfeld CS, Williams TJ, Dussaq AM, Ducklow H, Erickson M, Cavicchioli R, Murray AE. A metagenomic assessment of winter and summer

bacterioplankton from Antarctica Peninsula coastal surface waters. The ISME Journal. 2012;6:1901–15.

Hall AR, Scanlan PD, Morgan AD, Buckling A. Host-parasite coevolutionary arms races give way to fluctuating selection. Ecology Letters. 2011;14:635–42.

Hallet B. Playing Dr Jekyll and Mr Hyde: combined mechanisms of phase variation in bacteria. Current Opinion in Microbiology. 2001;4:570–81.

Hand RM. Bacterial populations of two saline Antarctic lakes. In: Trudinger PA, Walter MR, Ralph BJ (eds). Biogeochemistry of ancient and modern environments. Springer, Berlin, Heidelberg. 1980;123–9.

Hand RM and Burton HR. Microbial ecology of an Antarctic saline meromictic lake. Hydrobiologia. 1981;82:363–74.

Hazan R, Engelberg-Kulka H. *Escherichia coli* mazEF-mediated cell death as a defense mechanism that inhibits the spread of phage P1. Molecular Genetics and Genomics. 2004;272:227–34.

Hazra AB, Han AW, Mehta AP, Mok KC, Osadchiy V, Begley TP, Taga ME. Anaerobic biosynthesis of the lower ligand of vitamin B12. Proc Natl Acad Sci USA. 2015;112:10792–7.

Heal KR, Qin W, Ribalet F, Bertagnolli AD, Coyote-Maestas W, Hmelo LR, Moffett JW, Devol AH, Armbrust EV, Stahl DA, Ingalls AE. Two distinct pools of B12 analogs reveal community interdependencies in the ocean. PNAS. 2017;114:364–9.

Heck LW, Morihara K, Abrahamson DR. Degradation of soluble laminin and depletion of tissue-associated basement membrane laminin by *Pseudomonas aeruginosa* elastase and alkaline protease. Infection and Immunity. 1986;54:149–153.

Heising S, Richter L, Ludwig W, Schink B. *Chlorobium ferrooxidans* sp. nov., a phototrophic green sulfur bacterium that oxidizes ferrous iron in coculture with a "*Geospirillum*" sp. strain. Archives of Microbiology. 1999;172:116–24.

Heldt D, Lawrence AD, Lindenmeyer M, Deery E, Heathcote P, Rigby SE, Warren MJ. Aerobic synthesis of vitamin B12: Ring contraction and cobalt chelation. Biochemical Society Transactions. 2005;33:815–9.

Helliwell KA, Lawrence AD, Holzer A, Kudahl UJ, Sasso S, Kräutler B, Scanlan DJ, Warren MJ, Smith AG. Cyanobacteria and eukaryotic algae use different chemical variants of Vitamin B12. Current Biology. 2016;26:999–1008.

Herbert RA, Tanner AC. The isolation and some characteristics of photosynthetic bacteria (*Chromatiaceae* and *Chlorobiaceae*) from Antarctic marine sediments. Journal of Applied Microbiology. 1977; 43:437–45.

Hodgson DA, Vyverman IW, Sabbe K. Limnology and biology of saline lakes in the Rauer Islands, Eastern Antarctica. Antarctic Science. 2001;13:255–70.

Hofmann H, Federwisch L, Peeters F. Wave-induced release of methane: Littoral zones as a source of methane in lakes. Limnology and Oceanography. 2010;55:1990–2000.

Hogle SL, Thrash JC, Dupont CL, Barbeaua KA. Trace metal acquisition by marine heterotrophic bacterioplankton with contrasting trophic strategies. Applied and Environmental Microbiology. 2016;82:1613–1624.

Holmer M, Storkholm P. Sulphate reduction and sulphur cycling in lake sediments: a review. Freshwater Biology. 2001;46:431–51.

Holmkvist L, Ferdelman TG, Jørgensen BB. A cryptic sulfur cycle driven by iron in the methane zone of marine sediment (Aarhus Bay, Denmark). Geochimica et Cosmochimica Acta. 2011;75:3581–99.

Hordijk CA. Sulfur and carbon cycling in a stratifying freshwater lake. Landbouwuniversiteit te Wageningen, Netherlands. 1993

Huntemann M, Ivanova NN, Mavromatis K, Tripp HJ, Páez-Espino D, Tennessen K, Palaniappan K, Szeto E, Pillay M, Chen IMA, Pati A, Nielsen T, Markowitz VM, Kyrpides NC. The standard operating procedure of the DOE-JGI Metagenome Annotation Pipeline (MAP v.4). Standards in Genomic Science. 2015;11:17.

Huson DH, Albrecht B, Bagci1 C, Bessarab I, Gorska A, Jolic D, Williams RBH. MEGAN-LR: New algorithms allow accurate binning and easy interactive exploration of metagenomic long reads and contigs. Biology Direct. 2018;13:6.

Huson DH, Auch AF, Qi J, Schuster SC. MEGAN analysis of metagenomic data. Genome Research. 2007; 7:377–86.

Huson D, Beier S, Flade I, Gorska A, El-Hadidi M, Mitra S, Ruscheweyh H, Rewati Tappu D. MEGAN Community Edition - Interactive exploration and analysis of large-scale microbiome sequencing data. PLoS Computational Biology. 2016;12:e1004957.

Huson DH, Tappu R, Bazinet AL, Xie C, Cummings MP, Nieselt K, Williams R. Fast and simple protein-alignment-guided assembly of orthologous gene families from microbiome sequencing reads. Microbiome. 2017;5:11.

Hyman P, Abedon ST. Bacteriophage host range and bacterial resistance. In: Laskin AI, Sariaslani S, Gadd GM (eds). Advances in Applied Microbiology. Academic Press. 2010;217–48.

Imhoff JF. Phylogenetic taxonomy of the family *Chlorobiaceae* on the basis of 16S rRNA and fmo (Fenna–Matthews–Olson protein) gene sequences. International Journal of Systematic and Evolutionary Microbiology. 2003;53:941–51.

Imhoff JF. Biology of green sulfur bacteria. In: eLS. John Wiley & Sons, Ltd. Chichester. 2014.

Imhoff JF, Bias-Imhoff U. Lipids, quinones and fatty acids of anoxygenic phototropic bacteria. In: Blankenship RE, Madigan MT, Bauer CE (eds). Anoxygenic photosynthetic bacteria. Kluwer Academic Publishers, Dordrecht. 1995;179–205.

Jain C, Koren S, Dilthey A, Phillippy AM, Aluru S. A fast adaptive algorithm for computing whole-genome homology maps. Bioinformatics. 2018a;34:i748–56.

Jain C, Rodriguez-R LM, Phillippy AM, Konstantinidis KT, Aluru S. High throughput ANI analysis of 90K prokaryotic genomes reveals clear species boundaries. Nature Communications. 2018b;9:5114.

Jetten MSM, Van Niftrik L, Strous M, Kartal B, Keltjens JT, Op den Camp HJM. Biochemistry and molecular biology of anammox bacteria. Critical Reviews in Biochemistry and Molecular Biology. 2009;44:65–84.

Jing H, Liu H. Phylogenetic composition of *Prochlorococcus* and *Synechococcus* in cold eddies of the South China Sea. Aquatic Microbial Ecology. 2012;65:207–19.

Jones PG, Mitta M, Kim Y, Jiang W, Inouye M. Cold-shock induces a major ribosomal associated protein that unwinds double stranded RNA in *Escherichia coli*. PNAS. 1996;93,76–80.

Jørgensen BB, Findlay AJ, Pellerin A. The biogeochemical sulfur cycle of marine sediments. Frontiers in Microbiology. 2019;10:849.

Jørgensen BB, Kasten S. Sulfur cycling and methane oxidation. In: Schulz HD and Zabel M (eds). Marine Geochemistry. Springer, Berlin. 2006;271–309.

Kananavičiūtė R, Kvederavičiūtė K, Dabkevičienė D, Mackevičius G, Kuisienė N. Collagen-like sequences encoded by extremophilic and extremotolerant bacteria. Genomics. 2020;112:2271–81.

Kang DD, Froula J, Egan R, Wang Z. MetaBAT, an efficient tool for accurately reconstructing single genomes from complex microbial communities. PeerJ. 2015;3:e1165.

Karr EA, Sattley WM, Jung DO, Madigan MT, Achenbach LA. Remarkable diversity of phototrophic purple bacteria in a permanently frozen Antarctic lake. Applied Environmental Microbiology. 2003;69:4910–4.

Kepner RL, Wharton RA, Suttle CA. Viruses in Antarctic lakes. Limnology and Oceanography. 1998;43:1754–61.

Kielbasa SM, Wan R, Sato K, Horton P, Frith MC. Adaptive seeds tame genomic sequence comparison. Genome Research. 2011;21:487–93.

Kim M, Oh HS, Park SC, Chun J. Towards a taxonomic coherence between average nucleotide identity and 16S rRNA gene sequence similarity for species demarcation of prokaryotes. Int J Syst Evol Microbiol. 2014;64:346–351.

Kirsch F, Eitinger T. Transport of nickel and cobalt ions into bacterial cells by S components of ECF transporters. Biometals. 2014;27:653–60.

Kirk JTO. Light and photosynthesis in aquatic ecosystems. Cambridge University Press, Cambridge. 1994.

Kong W, Ream DC, Priscu JC, Morgan-Kiss RM. Diversity and expression of RubisCO genes in a perennially ice-covered Antarctic lake during the polar night transition. Applied Environmental Microbiology. 2012;78:4358–66.

Koonin EV, Makarova KS, Wolf YI. Evolutionary genomics of defense systems in archaea and bacteria. Annual Review of Microbiology. 2017;71:233–61.

Kraft B, Strous M, Tegetmeyer HE. Microbial nitrate respiration--genes, enzymes and environmental distribution. Journal of Biotechnology. 2011;155:104 – 17.

Kramm A, Kisiela M, Schulz R, Maser E. Short-chain dehydrogenases/reductases in cyanobacteria. FEBS Journal. 2012;279:1030–43.

Kristensen DM, Kannan L, Coleman MK, Wolf YI, Sorokin A, Koonin EV, Mushegian A. A low-polynomial algorithm for assembling clusters of orthologous groups from intergenomic symmetric best matches. Bioinformatics. 2010;26:1481-7.

Kumar S, Stecher G, Li M, Knyaz C, Tamura K. MEGA X: Molecular Evolutionary Genetics Analysis across computing platforms. Molecular Biology and Evolution. 2018;35:1547–9.

Kurtz S, Phillippy A, Delcher AL, Smoot M, Shumway M, Antonescu C, Salzberg SL. Versatile and open software for comparing large genomes. Genome Biology. 2004;5:R12.

Laarman AJ, Bardoel BW, Ruyken M, Fernie J, Milder FJ, Van Strijp JAG, Rooijakkers SHM. *Pseudomonas aeruginosa* alkaline protease blocks complement activation via the classical and lectin pathways. Journal of Immunology. 2012;188:386–93.

Laity C, Chater KF, Lewis CG, Buttner MJ. Genetic analysis of the phiC31-specific phage growth limitation (Pgl) system of *Streptomyces coelicolor* A3(2). Molecular Microbiology. 1993;7:329–36.

Lam P, Kuypers MMM. Microbial nitrogen cycling processes in oxygen minimum zones. Annual Review of Marine Science. 2011;3:317–45.

Langmead B, Salzberg S. Fast gapped-read alignment with Bowtie 2. Nature Methods. 2012;9:357–9.

Lauro FM, DeMaere MZ, Yau S, Brown MV, Ng C, Wilkins D, Raftery MJ, Gibson JAE, Andrews-Pfannkoch C, Lewis M, Hoffman JM, Thomas T, Cavicchioli R. An integrative study of a meromictic lake ecosystem in Antarctica. The ISME Journal. 2011;5:879–95.

Laybourn-Parry J, Bell EM. Ace Lake: three decades of research on a meromictic, Antarctic lake. Polar Biology. 2014;37:1685–99.

Laybourn-Parry J, Hofer JS, Sommaruga R. Viruses in the plankton of freshwater and saline Antarctic lakes. Freshwater Biology. 2001;46:1279–87.

Laybourn-Parry J, Marshall WA, Marchant HJ. Flagellate nutritional versatility as a key to survival in two contrasting Antarctic saline lakes. Freshwater Biology. 2005;50:830–8.

Laybourn-Parry J, Pearce DA. The biodiversity and ecology of Antarctic lakes: models for evolution. Philosophical transactions of the Royal Society of London. Series B, Biological sciences. 2007;362:2273–89.

Laybourn-Parry J, Pearce D. Heterotrophic bacteria in Antarctic lacustrine and glacial environments. Polar Biology. 2016;39:2207–25.

Laybourn-Parry J, Quayle W, Henshaw T. The biology and evolution of Antarctic saline lakes in relation to salinity and trophy. Polar Biology. 2002;25:542–52.

Li D, Liu CM, Luo R, Sadakane K, Lam TW. MEGAHIT: An ultra-fast single-node solution for large and complex metagenomics assembly via succinct de Bruijn graph. Bioinformatics. 2015;31:1674–6.

Li D, Luo R, Liu CM, Leung CM, Ting HF, Sadakane K, Yamashita H, Lam TW. MEGAHIT v1.0: A fast and scalable metagenome assembler driven by advanced methodologies and community practices. Methods. 2016;102:3–11.

Li H. BFC: correcting Illumina sequencing errors. Bioinformatics. 2015;31:2885–7.

Li H, Handsaker B, Wysoker A, Fennell T, Ruan J, Homer N, Marth G, Abecasis G, Durbin R, 1000 Genome Project Data Processing Subgroup. The Sequence alignment/map (SAM) format and SAMtools, Bioinformatics (2009) 25(16) 2078-9

Lim J, Thomas T, Cavicchioli R. Low Temperature Regulated DEAD-box RNA Helicase from the Antarctic Archaeon, *Methanococcoides burtonii*. Journal of Molecular Biology. 2000;297:553–67.

Llorens–Marès T, Liu Z, Allen LZ, Rusch DB, Craig MT, Dupont CL, Bryant DA, Casamayor EO. Speciation and ecological success in dimly lit waters: horizontal gene transfer in a green sulfur bacteria bloom unveiled by metagenomic assembly. The ISME Journal. 2017;11:201–11.

Loenen WAM, Raleigh EA. The other face of restriction: modification-dependent enzymes. Nucleic Acids Research. 2014;42:56–69.

Lopatina A, Tal N, Sorek R. Abortive Infection: Bacterial suicide as an antiviral immune strategy. Annual Review of Virology. 2020;7:371–84.

Lopes APY, Azevedo BOP, Emídio RC, Damiano DK, Nascimento ALTO, Barazzone GC. *In silico* analysis of genetic VapC profiles from the toxin-antitoxin type II VapBC modules among pathogenic, intermediate, and non-pathogenic *Leptospira*. Microorganisms. 2019;7:56.

López-Bueno A, Tamames J, Velázquez D, Moya A, Quesada A, Alcamí A. High diversity of the viral community from an Antarctic lake. Science. 2009;326:858–61.

Luo Y. Geochemical cycle and environmental effects of sulfur in lakes. IOP conference series: materials science and engineering. 2018;394:52039.

Madan NJ, Marshall WA, Laybourn-Parry J. Virus and microbial loop dynamics over an annual cycle in three contrasting Antarctic lakes. Freshwater Biology. 2005;50:1291–300.

Makarova KS, Anantharaman V, Aravind L, Koonin EV. Live virus-free or die: coupling of antivirus immunity and programmed suicide or dormancy in prokaryotes. Biology Direct. 2012;7:40.

Makarova KS, Wolf YI, Iranzo J, Shmakov JA, Alkhnbashi OS, Brouns SJJ, Charpentier E, Cheng D, Haft DH, Horvath P, Moineau S, Mojica FJM, Scott D, Shah SA, Siksnys V, Terns MP, Venclovas C, White MF, Yakunin AF, Yan W, Zhang F, Garrett RA, Backofen R, Van der Oost J, Barrangou R, Koonin EV.

Evolutionary classification of CRISPR–Cas systems: a burst of class 2 and derived variants. Nature Reviews Microbiology. 2020;18:67–83.

Makarova KS, Wolf YI, Snir S, Koonin EV. Defense islands in bacterial and archaeal genomes and prediction of novel defense systems. Journal of Bacteriology. 2011;193:6039–56.

Mancuso CA, Franzmann PD, Burton HR, Nichols PD. Microbial community structure and biomass estimates of a methanogenic Antarctic lake ecosystem as determined by phospholipid analysis. Microbial Ecology. 1990;19:73–95.

Mansor M and Macalady JL. Draft genome sequence of lampenflora *Chlorobium limicola* strain Frasassi in a sulfidic cave system. Genome Announcements. 2016; 4:e00357-16.

Masuda N, Nakaya S, Burton HR, Torii T. Trace element distribution in some saline lakes of the Vestfold Hills, Antarctica. Hydrobiologia.1988;165:103–14.

Matsen FA, Kodner RB, Armbrust EV. pplacer: linear time maximum-likelihood and Bayesian phylogenetic placement of sequences onto a fixed reference tree. BMC Bioinformatics. 2010;11:538.

Matsumoto GI. Biogeochemical study of organic-substances in Antarctic lakes. Hydrobiologia. 1989;172:265–89.

Mayr E. VIII-Nongeographic speciation. Systematics and the Origin of Species, from the Viewpoint of a Zoologist. Harvard University Press. 1999;194–5.

McKnight DM, Aiken GR, Smith RL. Aquatic fulvic-acids in microbially based ecosystems — results from two desert lakes in Antarctica. Limnology and Oceanography. 1991;36:998–1006.

McMinn A, Bleakley N, Steinburner K, Roberts D, Trenerry L. Effect of permanent sea ice cover and different nutrient regimes on the phytoplankton succession of fjords of the Vestfold Hills Oasis, Eastern Antarctica. Journal of Plankton Research. 2000;22:287–303.

Medlar AJ, Toronen P, Holm L. AAI-profiler: fast proteome-wide exploratory analysis reveals taxonomic identity, misclassification and contamination. Nucleic Acids Research. 2018;46:W479–85.

Menzel P, Ng KL, Krogh A. Fast and sensitive taxonomic classification for metagenomics with Kaiju. Nature Communications. 2016;7:11257.

Mesnage S, Dellarole M, Baxter NJ, Rouget JB, Dimitrov JD, Wang N, Fujimoto Y, Hounslow AM, Lacroix-Desmazes S, Fukase K, Foster SJ, Williamson MP. Molecular basis for bacterial peptidoglycan recognition by LysM domains. Nature Communications. 2014;5: 4269.

Meyer F, Overbeek R, Rodriguez A. FIGfams: yet another set of protein families. Nucleic Acids Research. 2009;37:6643–54.

Meyer F, Paarmann D, D'Souza M, Olson R, Glass EM, Kubal M, Paczian T, Rodriguez A, Stevens R, Wilke A, Wilkening J, Edwards RA. The metagenomics RAST server – a public resource for the automatic phylogenetic and functional analysis of metagenomes. BMC Bioinformatics. 2008;9:386.

Mikucki JA, Priscu JC. Bacterial Diversity Associated with Blood Falls, a Subglacial Outflow from the Taylor Glacier, Antarctica. Applied Environmental Microbiology. 2007;73:4029–39.

Miracle MR, Vicente E. Phytoplankton and photosynthetic sulphur bacteria production in the meromictic coastal lagoon of Cullera (Valencia, Spain). Verhandlungen des Internationalen Verein Limnologie. 1985;22:2214–20.

Miyamoto E, Tanioka Y, Nakao T, Barla F, Inui H, Fujita T, Watanabe F, Nakano Y. Purification and characterization of a corrinoid-compound in an edible cyanobacterium *Aphanizomenon flos-aquae* as a nutritional supplementary food. Journal of Agricultural and Food Chemistry. 2006;54:9604–7.

Montesinos E, Guerrero R, Abella C, Esteve I. Ecology and physiology of the competition for light between *Chlorobium limicola* and *Chlorobium phaeobacteroides* in natural habitats. Applied and Environmental Microbiology. 1983;46:1007–16.

Moreira D, López-García P. Phylotype. In: Gargaud M, Amils R, Quintanilla JC, CleavesII HJ, Irvine WM, Pinti DL, Viso M (eds). Encyclopedia of Astrobiology. Springer, Berlin, Heidelberg. 2011.

Morett E, Saab-Rincón G, Olvera L, Olvera M, Flores H, Grande R. Sensitive genome-wide screen for low secondary enzymatic activities: the YjbQ family shows thiamin phosphate synthase activity. Journal of Molecular Biology. 2008;376:839–53.

Mosier AC, Murray AE, Fritsen CH. Microbiota within the perennial ice cover of Lake Vida, Antarctica. FEMS Microbiology Ecology. 2007;59:274–88.

Nadeau TL, Castenholz RW. Characterization of psychrophilic oscillatorians (cyanobacteria) from Antarctic meltwater ponds. Journal of Phycology. 2000;36:914–23.

Nawrocki EP, Eddy SR. Infernal 1.1: 100-fold faster RNA homology searches. Bioinformatics. 2013;29:2933–5.

Ng C, DeMaere MZ, Williams TJ, Lauro FM, Raftery M, Gibson JAE, Andrews-Pfannkoch C, Lewis M, Hoffman JM, Thomas T, Cavicchioli R. Metaproteogenomic analysis of a dominant green sulfur bacterium from Ace Lake, Antarctica. The ISME Journal. 2010;4:1002–19.

Nicol JW, Helt GA, Blanchard SG, Raja JA, Loraine AE. The Integrated Genome Browser: free software for distribution and exploration of genome-scale datasets. Bioinformatics. 2009;25:2730–31.

Nishitani G, Yamaguchi M. Seasonal succession of ciliate *Mesodinium* spp. with red, green, or mixed plastids and their association with cryptophyte prey. Scientific Reports. 2018;8:17189.

Nurk S, Bankevich A, Antipov D, Gurevich AA, Korobeynikov A, Lapidus A, Prjibelski AD, Pyshkin A, Sirotkin A, Sirotkin Y, Stepanauskas R, Clingenpeel SR, Woyke T, McLean JS, Lasken R, Tesler G, Alekseyev MA, Pevzner PA. Assembling single-cell genomes and mini-metagenomes from chimeric MDA products. Journal of Computational Biology. 2013;20:714–37.

Nurk S, Meleshko D, Korobeynikov A, Pevzner PA. metaSPAdes: a new versatile metagenomic assembler. Genome Research. 2017;27: 824–34.

Ofir G, Melamed S, Sberro H, Mukamel Z, Silverman S, Yaakov G, Doron S, Sorek R. DISARM is a widespread bacterial defence system with broad anti-phage activities. Nat Microbiol. 2018;3:90–8.

Ondov BD, Bergman NH, Phillippy AM. Interactive metagenomic visualization in a Web browser. BMC Bioinformatics. 2011;12:385.

Overbeek R, Begley T, Butler RM, Choudhuri JV, Chuang HY, Cohoon M, De Crécy-Lagard V, Diaz N, Disz T, Edwards R, Fonstein M, Frank ED, Gerdes S, Glass EM, Goesmann A, Hanson A, Iwata-Reuyl D, Jensen R, Jamshidi N, Krause L, Kubal M, Larsen N, Linke B, McHardy AC, Meyer F, Neuweger H, Olsen G, Olson R, Osterman A, Portnoy V, Pusch GD, Rodionov DA, Rückert C, Steiner J, Stevens R, Thiele I, Vassieva O, Ye Y, Zagnitko O, Vonstein V. The subsystems approach to genome annotation and its use in the project to annotate 1000 genomes. Nucleic Acids Research. 2005;33:5691–702.

Overmann J, Cypionka H, Pfennig N. An extremely low-light-adapted green sulfur bacterium from the Black Sea. Limnology and Oceanography. 1992;37:150–5.

Páez-Espino D, Chen IMA, Palaniappan K, Ratner A, Chu K, Szeto E, Pillay M, Huang J, Markowitz VM, Nielsen T, Huntemann M, Reddy TBK, Pavlopoulos GA, Sullivan MB, Campbell BJ, Chen F, McMahon K, Hallam SJ, Denef V, Cavicchioli R, Caffrey SM, Streit WR, Webster J, Handley KM, Salekdeh GH, Tsesmetzis N, Setubal JC, Pope PB, Liu WT, Rivers AR, Ivanova NN, Kyrpides NC. IMG/VR: a database of cultured and uncultured DNA Viruses and retroviruses. Nucleic Acids Research. 2017;45:D457–65.

Páez-Espino D, Eloe-Fadrosh EA, Pavlopoulos GA, Thomas AD, Huntemann M, Mikhailova N, Rubin E, Ivanova NN, Kyrpides NC. Uncovering Earth's virome. Nature. 2016;536:425–30.

Páez-Espino D, Roux S, Chen IA, Palaniappan K, Ratner A, Chu K, Huntemann M, Reddy TBK, Pons JC, Llabrés M, Eloe-Fadrosh EA, Ivanova NN, Kyrpides NC. IMG/VR v.2.0: an integrated data management and analysis system for cultivated and environmental viral genomes. Nucleic Acids Research. 2019a;47:D678–86.

Páez-Espino D, Zhou J, Roux S, Nayfach S, Pavlopoulos GA, Schulz F, McMahon KD, Walsh D, Woyke T, Ivanova NN, Eloe-Fadrosh EA, Tringe SG, Kyrpides NC.

Diversity, evolution, and classification of virophages uncovered through global metagenomics. Microbiome. 2019b;7:157.

Palenik B, Brahamsha B, Larimer FW, Land M, Hauser L, Chain P, Lamerdin J, Regala W, Allen EE, McCarren J, Paulsen I, Dufresne A, Partensky F, Webb EA, Waterbury J. The genome of a motile marine *Synechococcus*. Nature. 2003;424:1037–42.

Panwar P, Allen MA, Williams TJ, Hancock AM, Brazendale S, Bevington J, Roux S, Páez-Espino D, Nayfach S, Berg M, Schulz F, Chen IMA, Huntemann M, Shapiro N, Kyrpides NC, Woyke T, Eloe-Fadrosh EA, Cavicchioli R. Influence of the polar light cycle on seasonal dynamics of an Antarctic lake microbial community. Microbiome. 2020;8:116.

Parks DH, Imelfort M, Skennerton CT, Hugenholtz P, Tyson GW. CheckM: assessing the quality of microbial genomes recovered from isolates, single cells, and metagenomes. Genome Research. 2015;25:1043–55.

Parks DH, Rinke C, Chuvochina M, Chaumeil PA, Woodcroft BJ, Evans PA, Hugenholtz P, Tyson GW. Recovery of nearly 8,000 metagenome-assembled genomes substantially expands the tree of life. Nature Microbiology. 2017;2:1533–42.

Pearce DA, Wilson WH. Viruses in Antarctic ecosystems. Antarctic Science. 2003;15:319–31.

Peat HJ, Clarke A, Convey P. Diversity and biogeography of the Antarctic flora. Journal of Biogeography. 2007;34:132–46.

Perčulija V, Ouyang S. Diverse Roles of DEAD/DEAH-Box Helicases in Innate Immunity and Diseases. In: Tuteja R (eds). Helicases from All Domains of Life. Academic Press. 2019;141–71.

Perriss SJ, Laybourn-Parry J, Marehont HJ. Widespread occurrence of populations of the unique autotrophic ciliate *Mesodinium rubrum* (Ciliophora: Haptorida) in brackish and saline lakes of the Vestfold Hills (eastern Antarctica). Polar Biology. 1995;15:423–8.

Petersen BL, Jensen PE, Gibson LC, Stummann BM, Hunter CN, Henningsen KW. Reconstitution of an active magnesium chelatase enzyme complex from the bchI, -D, and -H gene products of the green sulfur bacterium *Chlorobium vibrioforme* in *Escherichia coli*. Journal of Bacteriology. 1998;180:699–704.

Pfennig N, Trüper HG. Higher taxa of the phototrophic bacteria. International Journal of Systematic Bacteriology. 1971;21:17–18.

Pieńko T, Trylska J. Extracellular loops of BtuB facilitate transport of vitamin B12 through the outer membrane of *E. coli*. PLoS Computational Biology. 2020;16:e1008024.

Powell LM, Bowman JP, Skerratt JH, Franzamnn PD, Burton HR. Ecology of a novel *Synechococcus* clade occurring in dense populations in saline Antarctic lakes. Mar Ecol Prog Ser. 2005;291:65–80.

Prakash T, Taylor TD. Functional assignment of metagenomic data: challenges and applications. Briefings in Bioinformatics. 2012;13:711–27.

Price MN, Dehal PS, Arkin AP. FastTree 2 – Approximately maximum-likelihood trees for large alignments. PLOS ONE. 2010;5: e9490.

Pritchard L, Glover RH, Humphris S, Elphinstone JG, Toth IK. Genomics and taxonomy in diagnostics for food security: soft-rotting enterobacterial plant pathogens. Analytical Methods. 2016;8:12–24.

Rankin LM. The ecology and taxonomy of *Synechocccus* from saltwater lakes in the Vestfold Hills, Antarctica. PhD thesis, University of Tasmania; 1998.

Rankin LM, Franzmann PD, McMeekin TA, Burton HR. Seasonal distribution of picocyanobacteria in Ace Lake, a marine derived Antarctic lake. In: Battaglia B, Valencia J, Walton DWH (eds). Antarctic communities, species, structure and survival. Cambridge: Cambridge University Press. 1997;178–84.

Rankin LM, Gibson JAE, Franzmann PD, Burton HR. The chemical stratification and microbial communities of Ace Lake: a review of the characteristics of a marine derived meromictic lake. Polarforschung. 1999;66:33–52.

Redder P, Hausmann S, Khemici V, Yasrebi H, Linder P. Bacterial versatility requires DEAD-box RNA helicases. FEMS Microbiology Reviews. 2015;39:392–412.

Reid IN, Sparks WB, Lubnow S, McGrath M, Livio M, Valenti J, Sowers KR, Shukla HD, MacAuley S, Miller T, Suvanasuthi R, Belas R, Colman A, Robb FT, DasSarma P, Müller JA, Coker JA, Cavicchioli1 R, Chen F, DasSarma S. Terrestrial models for extraterrestrial life: methanogens and halophiles at Martian temperatures. International Journal of Astrobiology. 2006;5:89–97.

Reitzer LJ, Magasanik B. Asparagine synthetases of *Klebsiella aerogenes*: properties and regulation of synthesis. Journal of Bacteriology. 1982;151:1299–313.

Repeta DJ, Simpson DJ, Jørgensen BB, Jannasch HW. Evidence for the existence of anoxygenic photosynthesis from the distribution of bacteriochlorophylls in the Black Sea. Nature. 1989;342:69–72.

Rickard D, Luther GW 3rd. Chemistry of iron sulfides. Chemical Reviews. 2007;107:514–62.

Richter M, Rosselló-Móra R, Glöckner FO, Peplies J. JSpeciesWS: a web server for prokaryotic species circumscription based on pairwise genome comparison. Bioinformatics. 2016;32:929–31.

Robinson JT, Thorvaldsdóttir H, Winckler W, Guttman M, Lander ES, Getz G, Mesirov JP. Integrative Genomics Viewer. Nature Biotechnology. 2011;29:24–6.

Rodionov DA, Vitreschak AG, Mironov AA, Gelfand MS. Comparative genomics of the vitamin B12 metabolism and regulation in prokaryotes. Journal of Biological Chemistry. 2003;278:41148–59.

Rodionov DA, Hebbeln P, Gelfand MS, Eitinger T. Comparative and functional genomic analysis of prokaryotic nickel and cobalt uptake transporters: evidence for a novel group of ATP-binding cassette transporters. Journal of Bacteriology. 2006;188:317–27.

Roeselers G, Norris TB, Castenholz RW, Rysgaard S, Glud RN, Kühl M, Muyzer G. Diversity of phototrophic bacteria in microbial mats from Arctic hot springs (Greenland). Environmental Microbiology. 2007;9:26–38.

Roessner CA, Huang KX, Warren MJ, Raux E, Scott AI. Isolation and characterization of 14 additional genes specifying the anaerobic biosynthesis of cobalamin

(vitamin B12) in *Propionibacterium freudenreichii* (*P. shermanii*). Microbiology. 2002;148:1845–53.

Rosendahl S, Tamman H, Brauer A, Remm M, Hõrak R. Chromosomal toxin-antitoxin systems in Pseudomonas putida are rather selfish than beneficial. Scientific Reports. 2020;10:9230.

Richter M, Rosselló-Móra R. Shifting the genomic gold standard for the prokaryotic species definition. PNAS. 2009;106:19126–31.

Roth JR, Lawrence JG, Rubenfield M, Kieffer-Higgins S, Church GM. Characterization of the cobalamin (vitamin B12) biosynthetic genes of *Salmonella typhimurium*. Journal of Bacteriology. 1993;175:3303–16.

Roux S, Enault F, Hurwitz BL, Sullivan MB. VirSorter: mining viral signal from microbial genomic data. PeerJ. 2015;3:e985.

Rutherford K, Parkhill J, Crook J, Horsnell T, Rice P, Rajandream MA, Barrell B. Artemis: sequence visualization and annotation. Bioinformatics. 2000;16:944–5.

Ryu JY, Kim HU, Lee SY. Deep learning enables high-quality and high-throughput prediction of enzyme commission numbers. PNAS. 2019;116:13996–14001.

Sakurai H, Ogawa T, Shiga M, Inoue K. Inorganic sulfur oxidizing system in green sulfur bacteria. Photosynthesis Research. 2010;104:163–76.

Santos JA, Rempel S, Mous STM, Pereira CT, Ter Beek J,De Gier JW, Guskov A, Slotboom DJ. Functional and structural characterization of an ECF-type ABC transporter for vitamin B12. eLife. 2018;7:e35828.

Saravanan M, Vasu K, Nagaraja V. Evolution of sequence specificity in a restriction endonuclease by a point mutation. 2008;105:10344–7.

Sato K, Ishida K, Kuno T, Mizuno A, Shimizu S. Regulation of vitamin B12 and bacteriochlorophyll biosynthesis in a facultative methylotroph, *Protaminobacter ruber*. Journal of Nutritional Science and Vitaminology (Tokyo). 1981;27:439-47.

Sattley WM, Madigan MT. Isolation, characterization, and ecology of cold-active, chemolithotrophic, sulfur-oxidizing bacteria from perennially ice-covered Lake Fryxell, Antarctica. Applied Environmental Microbiology. 2006;72:5562–8.

Sauer J, Dirmeier U, Forchhammer. The *Synechococcus* Strain PCC 7942 glnN Product (Glutamine Synthetase III) helps recovery from prolonged nitrogen chlorosis. Journal of Bacteriology. 2000;182:5615–9.

Säwström C, Anesio MA, Granéli W, Laybourn-Parry J. Seasonal viral loop dynamics in two large ultraoligotrophic Antarctic freshwater lakes. Microbial Ecology. 2007;53:1–11.

Schmid J, Heider D, Wendel NJ, Sperl N, Sieber V. Bacterial glycosyltransferases: challenges and opportunities of a highly diverse enzyme class toward tailoring natural products. Frontiers in Microbiology. 2016;7:182.

Schmidt K. Biosynthesis of carotenoids. In: Clayton RK and Sistrom WR (eds). The photosynthetic bacteria. Plenum Press, New York. 1978;729–50.

Schouten S, Rijpstra WIC, Kok M, Hopmans EC, Summons RE, Volkman JK, Damsté JSS. Molecular organic tracers of biogeochemical processes in a saline meromictic lake (Ace Lake). Geochimica et Cosmochimica Acta. 2001;65:1629–40.

Schulz F, Roux S, Páez-Espino D, Jungbluth S, Walsh DA, Denef VJ, McMahon KD, Konstantinidis KT, Eloe-Fadrosh EA, Kyrpides NC, Woyke T. Giant virus diversity and host interactions through global metagenomics. Nature. 2020;578:432–6.

Schulz S, Wilkes M, Mills DJ, Kühlbrandt W, Meier T. Molecular architecture of the N-type ATPase rotor ring from *Burkholderia pseudomallei*. EMBO Reports. 2017;18:526–35.

Segata N, Waldron L, Ballarini A, Narasimhan V, Jousson O, Huttenhower C. Metagenomic microbial community profiling using unique clade-specific marker genes. Nature Methods. 2012;9:811–4.

Shelton AN, Seth EC, Mok KC, Han AW, Jackson SN, Haft DR, Taga ME. Uneven distribution of cobamide biosynthesis and dependence in bacteria predicted by comparative genomics. The ISME Journal. 2019;13:789–804.

Singh SM, Elster J. Cyanobacteria in Antarctic Lake Environments. In: Seckbach J (eds). Algae and Cyanobacteria in Extreme Environments. Cellular Origin, Life in Extreme Habitats and Astrobiology. Springer, Dordrecht. 2007;11:303–320.

Sohm JA, Ahlgren NA, Thomson ZJ, Williams C, Moffett JW, Saito MA, Webb EA, Rocap G. Co-occurring *Synechococcus* ecotypes occupy four major oceanic regimes defined by temperature, macronutrients and iron. The ISME Journal. 2016;10:333–45.

Sorokin YI, Donato N. On the carbon and sulfur metabolism in the meromictic Lake Faro (Sicily) Italy. Hydrobiologia. 1975;47:241–52.

Stamatakis A. RAxML version 8: a tool for phylogenetic analysis and post-analysis of large phylogenies. Bioinformatics. 2014;30:1312–3.

Stackebrandt E, Goebel BM. Taxonomic note: a place for DNA-DNA reassociation and 16S rRNA sequence analysis in the present species definition in bacteriology. Int J Syst Bact. 1994;44:846–849.

Stern A, Sorek R. The phage-host arms race: shaping the evolution of microbes. BioEssays. 2011;33:43–51.

Stevens MI, Hogg ID. In: Bergstrom DM, Convey P, Huiskes AHL (eds). Trends in Antarctic Terrestrial and Limnetic Ecosystems. Springer. 2006;177–92.

Strock JS. Ammonification. In: Jørgensen SE, Fath BD (eds). Encyclopedia of Ecology. Elsevier Science. 2008;162–5.

Strous M, Fuerst JA, Kramer EHM, Logemann S, Muyzer G, Van de Pas-Schoonen KT, Webb R, Kuenen JG, Jetten MSM. Missing lithotroph identified as new planctomycete. Nature. 1999;400:446–9.

Sumby P, Smith MC. Phase variation in the phage growth limitation system of *Streptomyces coelicolor* A3(2). Journal of Bacteriology. 2003;185:4558–63.

Swadling KM. Influenee of seasonal iee formation on life eycle strategies of Antarctie copepods. PhD thesis, University of Tasmania; 1998.

Takacs CD, Priscu J, McKnight D. Bacterial dissolved organic carbon demand in McMurdo Dry Valley lakes, Antarctica. Limnology and Oceanography. 2001;46:1189–94.

Taga ME, Larsen NA, Howard-Jones AR, Walsh CT, Walker GC. BluB cannibalizes flavin to form the lower ligand of vitamin B12. Nature. 2007;446:449–453.

Tang KH, Blankenship RE. Both forward and reverse TCA cycles operate in green sulfur bacteria. The Journal of Biological Chemistry. 2010;285:35848–54.

Tang J, Du LM, Liang YM, Daroch M. Complete genome sequence and comparative analysis of *Synechococcus* sp. CS-601 (SynAce01), a cold-adapted cyanobacterium from an oligotrophic Antarctic habitat. International Journal of Molecular Sciences. 2019; 20:152.

Taylor DE. Bacterial tellurite resistance. Trends in Microbiology. 1999;7:111–5.

Teeling H, Glöckner FO. Current opportunities and challenges in microbial metagenome analysis—a bioinformatic perspective. Briefings in Bioinformatics. 2012;13:728–42.

Thiel J, Byrne JM, Kappler A, Schink B, Pester M. Pyrite formation from FeS and H_2S is mediated through microbial redox activity. PNAS. 2019;116:6897–902.

Tillett D, Neilan BA. Xanthogenate nucleic acid isolation from cultured and environmental cyanobacteria. Journal of Phycology. 2000;36:251–8.

Tschitschko B, Erdmann S, DeMaere MZ, Roux S, Panwar P, Allen MA, Williams TJ, Brazendale S, Hancock AM, Eloe-Fadrosh EA, Cavicchioli R. Genomic variation and biogeography of Antarctic haloarchaea. Microbiome. 2018;6:113.

Tschitschko B, Williams TJ, Allen MA, Páez-Espino D, Kyrpides N, Zhong L, Raftery MJ, Cavicchioli R. Antarctic archaea–virus interactions: metaproteome-led analysis of invasion, evasion and adaptation. The ISME Journal. 2015;9:2094–107.

Tschitschko B, Williams TJ, Allen MA, Zhong L, Raftery MJ, Cavicchioli R. Ecophysiological distinctions of Haloarchaea from a hypersaline Antarctic lake as determined by metaproteomics. Applied and Environmental Microbiology. 2016;82:3165–73.

Ueno Y, Arita M, Kumagai T, Asai K. Processing sequence annotation data using the Lua programming language. Genome Informatics. 2003;14:154–63.

Van Gemerden H, Mas J. Ecology of phototrophic sulfur bacteria. In: Blankenship RE, Madigan MT, Bauer CE (eds). Anoxygenic Photosynthetic Bacteria. Kluwer Academic Publishers, The Netherlands. 1995;49–85.

Van Trappen S, Mergaert J, Swings J. *Loktanella salsilacus* gen. nov., sp. nov., *Loktanella fryxellensis* sp. nov. and *Loktanella vestfoldensis* sp. nov., new members of the Rhodobacter group, isolated from microbial mats in Antarctic lakes. International Journal of Systematic and Evolutionary Microbiology. 2004;54:1263–9.

Velasco-Castrillón A, Gibson JAE, Stevens MI. A review of current Antarctic limno-terrestrial microfauna. Polar Biology. 2014;37:1517–31.

Vick-Majors TJ, Priscu JC, Amaral-Zettler LA. Modular community structure suggests metabolic plasticity during the transition to polar night in ice-covered Antarctic lakes. The ISME Journal. 2014;8:778–89.

Von Ballmoos C, Cook GM, Dimroth P. Unique rotary ATP synthase and its biological diversity. Annual Review of Biophysics. 2008;37:43–64.

Wahlund TM, Woese CR, Castenholz RW, Madigan MT. A thermophilic green sulfur bacterium from New Zealand hot springs, *Chlorobium tepidum* sp. nov. Archives of Microbiology. 1991;156:81–90.

Walker KF. The stability of meromictic lakes in central Washington. Limnology and Oceanography. 1974;19:209–22.

Wang H, Fewer DP, Sivonen K. Genome mining demonstrates the widespread occurrence of gene clusters encoding bacteriocins in cyanobacteria. PLoS ONE. 2011;6:e22384.

Watanabe F, Katsura H, Takenaka S, Fujita T, Abe K, Tamura Y, Nakatsuka T, Nakan Y. Pseudovitamin b12 is the predominant cobamide of an algal health food, spirulina tablets. Journal of Agricultural and Food Chemistry. 1999;47:4736−41.

Watanabe F, Miyamoto E, Fujita T, Tanioka Y, Nakano Y. Characterization of a corrinoid compound in the edible (blue-green) alga, Suizenji-nori. Bioscience, Biotechnology, and Biochemistry. 2006;70:3066–8.

Watanabe F, Tanioka Y, Miyamoto E, Fujita T, Takenaka H, Nakano Y. Purification and characterization of corrinoid-compounds from the dried powder of an edible cyanobacterium, *Nostoc commune* (Ishikurage). Journal of Nutritional Science and Vitaminology. 2007;53:183–6.

Wetzel RG. Limnology: Lake and river ecosystems. Elsevier Academic Press, London. 2001.

Wilkins D, Yau S, Williams TJ, Allen MA, Brown MV, DeMaere MZ, Lauro FM, Cavicchioli R. Key microbial drivers in Antarctic aquatic environments. FEMS Microbiology Reviews. 2013;37:303–35.

Williams R. Phytoplankton populations in an Antarctic saline lake. MSc thesis, University of Melbourne; 1979.

Williams TJ, Lauro FM, Ertan H, Burg DW, Poljak A, Raftery MJ, Cavicchioli R. Defining the response of a microorganism to temperatures that span its complete growth temperature range (-2°C to 28°C) using multiplex quantitative proteomics. Environmental Microbiology. 2011;13:2186–203.

Williams TJ, Liao Y, Ye Jun, Kuchel RP, Poljak A, Raftery MJ, Cavicchioli R. Cold adaptation of the Antarctic haloarchaea *Halohasta litchfieldiae* and *Halorubrum lacusprofundi*. Environmental Microbiology. 2017;19:2210–27.

Williams TJ, Long E, Evans F, DeMaere MZ, Lauro FM, Raftery MJ, Ducklow H, Grzymski JJ, Murray AE, Cavicchioli R. A metaproteomic assessment of winter and summer bacterioplankton from Antarctic Peninsula coastal surface waters. The ISME Journal. 2012;6:1883–1900.

Willows RD, Al-Karadaghi S, Hansson M, Fodje MN, Hansson A, Olsen JG, Gough S. Interplay between an AAA module and an integrin I domain may regulate the function of magnesium chelatase. Journal of Molecular Biology. 2001;311:111–22.

Woodson JD, Escalante-Semerena JC. CbiZ, an amidohydrolase enzyme required for salvaging the coenzyme B12 precursor cobinamide in archaea. PNAS. 2004;101:3591–6.

Woodson JD, Zayas CL, Escalante-Semerena JC. A new pathway for salvaging the coenzyme B12 precursor cobinamide in archaea requires cobinamide-phosphate synthase (CbiB) enzyme activity. Journal of Bacteriology. 2003;185:7193–201.

Wooley JC, Ye Y. Metagenomics: Facts and artifacts, and computational challenges. Journal of Computer Science and Technology. 2009;25:71–81.

Wright SW, Burton HR. The biology of Antarctic saline lakes. In: Williams WD (eds). Salt Lakes. Developments in Hydrobiology. Springer, Dordrecht. 1981;5:319–38.

Yamaguchi Y, Park JH, Inouye M. Toxin-antitoxin systems in bacteria and archaea. Annual Review of Genetics. 2011;45:61–79.

Yau S, Lauro FM, DeMaere MZ, Brown MV, Thomas T, Raftery MJ, Andrews-Pfannkoch C, Lewis M, Hoffman JM, Gibson JA, Cavicchioli R. Virophage control of Antarctic algal host–virus dynamics. PNAS. 2011;108:6163–8.

Yau S, Lauro FM, Williams TJ, DeMaere MZ, Brown MV, Rich J, Gibson JAE, Cavicchioli R. Metagenomic insights into strategies of carbon conservation and unusual sulfur biogeochemistry in a hypersaline Antarctic lake. The ISME Journal. 2013;7:1944-61.

Yu Z, An B, Ramshaw JAM, Brodsky B. Bacterial collagen-like proteins that form triple-helical structures. Journal of Structural Biology. 2014;186:451–61.

Zablocki O, Van Zyl L, Adriaenssens EM, Rubagotti E, Tuffin M, Cary SC, Cowan D. High-level diversity of tailed phages, eukaryote-associated viruses, and virophage-like elements in the metaviromes of Antarctic soils. Applied Environmental Microbiology. 2014;80:6888–97.

Zborowskya S and Lindell D. Resistance in marine cyanobacteria differs against specialist and generalist cyanophages. PNAS. 2019;116:16899–908.

Zopfi J, Böttcher ME, Jørgensen BB. Biogeochemistry of sulfur and iron in Thioploca-colonized surface sediments in the upwelling area off central chile. Geochimica et Cosmochimica Acta. 2008;72:827–43.

Appendix A

Antarctic metagenomes and MAGs

Table A1. List of Antarctic metagenomes. The Megahit-assembled metagenomes were used for testing various Cavlab pipeline and arCOG pipeline methods (Chapter 2). The Spades-assembled metagenomes were used for in-depth analysis of Ace Lake (Chapter 3), analysis of genomic variation in Ace Lake *Synechococcus* (Chapter 4), and analysis of genomic variation in *Chlorobium* from Ace Lake, Ellis Fjord, and Taynaya Bay (Chapter 5). The blue-highlighted metagenomes from Ace Lake and Deep Lake are Spades-assembled metagenomes that were used for comparison between software/methods for taxonomic classification and abundance estimation (Chapter 2 Figure 2.7; Table 2.5; section 2.3.1.4). The Ace Lake depths were named in Chapter 3 — Upper 1: 0 m; Upper 2: 5 m; Upper 3: 11.5–13 m; Interface: 12.7–14.5 m; Lower 1: 14–16 m; Lower 2: 18–19 m; Lower 3: 23–24 m, and this nomenclature was used in Chapters 4 and 5 as well. *Assembled metagenome size in the table refers to the total length of all contigs assembled from a metagenome. † The Taynaya Bay Spades-assembled metagenomes were assembled from QC filtered reads, with read corrections using BFC software as well as without read corrections (indicated by nbfc) (Chapter 5 section 5.2.1). Filter fractions: 3, 3–20 μm; 0.8, 0.8–3 μm; 0.1, 0.1–0.8 μm; 0.22, 0.22–20 μm.

Sample collection date (DDMMYYYY); Depth; Filter fraction	IMG Genome IDs	Metagenome filtered reads (bp)	Assembled metagenome size (bp)*	Total protein-coding genes
Megahit-assembled metagenomes				
Ace Lake				
19/11/2008; 5 m; 3 μm	3300005913	10,168,447,444	609,947,171	1,400,648
19/11/2008; 5 m; 0.8 μm	3300005912	8,608,322,293	548,632,179	1,169,129
19/11/2008; 5 m; 0.1 μm	3300007074	9,326,252,194	300,659,399	752,354
21/11/2008; 11.8 m; 3 μm	3300005914	9,958,328,840	523,650,583	1,207,764
21/11/2008; 11.8 m; 0.8 μm	3300005933	10,372,524,015	613,006,575	1,355,170
21/11/2008; 11.8 m; 0.1 μm	3300005931	8,652,779,583	332,185,299	818,007
21/11/2008; 12.8 m; 3 μm	3300005911	7,377,945,147	290,605,516	645,253
21/11/2008; 12.8 m; 0.8 μm	3300005909	7,969,400,898	186,558,379	427,573
21/11/2008; 12.8 m; 0.1 μm	3300005910	15,030,492,867	167,053,133	453,549
21/11/2008; 14.1 m; 3 μm	3300005919	8,878,877,148	779,524,746	1,963,156
21/11/2008; 14.1 m; 0.8 μm	3300005917	9,024,438,900	713,249,915	1,804,870
21/11/2008; 14.1 m; 0.1 μm	3300005918	7,433,358,222	769,682,061	2,122,266
21/11/2008; 18 m; 3 μm	3300005916	9,701,518,914	761,641,541	1,746,645
21/11/2008; 18 m; 0.8 μm	3300005932	10,550,636,481	565,206,648	1,301,029
21/11/2008; 18 m; 0.1 μm	3300005915	8,489,799,212	702,459,535	1,734,687
23/11/2008; 23 m; 3 μm	3300005939	8,926,498,848	835,219,590	2,212,602

23/11/2008; 23 m; 0.8 µm	3300005936	8,835,913,368	777,839,517	2,005,345
23/11/2008; 23 m; 0.1 µm	3300005935	8,391,237,271	855,455,100	2,375,770
Deep Lake				
1/12/2006; 0 m; 3 µm	3300005928	10,602,162,939	127,444,550	303,787
1/12/2006; 0 m; 0.8 µm	3300005930	9,873,818,475	188,685,715	442,756
1/12/2006; 0 m; 0.1 µm	3300005929	4,982,821,567	119,306,653	312,415
1/12/2006; 0 m; <0.1 µm (1)	3300012127	4,699,521,915	110,118,418	217,941
1/12/2006; 0 m; <0.1 µm (2)	3300012027	3,227,977,575	79,462,683	155,026
30/11/2008; 0 m; 3 µm	3300012262	3,687,498,539	86,277,388	165,177
30/11/2008; 0 m; 0.8 µm	3300011181	4,100,749,829	101,045,660	194,396
30/11/2008; 5 m; 0.1 µm	2084038019	354,069,196	52,160,100	124,915
30/11/2008; 13 m; 0.1 µm	2100351014	432,228,299	80,245,247	199,955
30/11/2008; 24 m; 3 µm	3300005268	347,753,163	46,756,487	88,928
30/11/2008; 24 m; 0.8 µm	3300005273	250,425,457	60,099,211	129,645
30/11/2008; 24 m; 0.1 µm	2084038011	400,299,239	59,385,153	140,400
30/11/2008; 36 m; 3 + 0.8 + 0.1 µm pooled	2140918027	1,070,926,568	190,563,105	472,623
30/11/2008; 5 + 13 + 24 m pooled; <0.1 µm	3300012265	3,882,704,783	102,592,878	199,607
13/12/2013; 0 m; 3 µm	3300012121	4,982,361,265	100,145,597	198,025
13/12/2013; 0 m; 0.8 µm	3300012145	5,072,921,653	161,744,775	322,931
13/12/2013; 0 m; 0.1 µm	3300012104	3,566,648,595	67,783,456	142,474
11/2/2014; 0 m; 3 µm	3300012107	4,555,857,832	76,963,716	150,151
11/2/2014; 0 m; 0.8 µm	3300011169	3,623,989,873	56,174,165	109,777
11/2/2014; 0 m; 0.1 µm	3300011170	2,958,540,831	53,500,386	121,610
12/6/2014; 0 m; 3 µm	3300012025	3,855,296,102	73,641,348	132,179
12/6/2014; 0 m; 0.8 µm	3300012029	3,868,167,607	100,880,304	193,226
12/6/2014; 0 m; 0.1 µm	3300012250	2,754,110,568	60,323,580	132,580
25/8/2014; 0 m; 3 µm	3300011177	4,243,734,110	80,910,744	148,663
25/8/2014; 0 m; 0.8 µm	3300012106	4,342,913,643	83,319,071	146,854
25/8/2014; 0 m; 0.1 µm	3300012026	3,342,585,644	65,743,123	139,317
24/11/2014; 0 m; 3 µm	3300012111	4,432,369,610	82,492,691	156,358
24/11/2014; 0 m; 0.8 µm	3300012103	4,008,007,557	74,612,275	138,596
24/11/2014; 0 m; 0.1 µm	3300011179	3,696,124,203	64,342,469	121,108
24/11/2014; 0 m; <0.1 µm	3300012115	3,544,797,704	84,892,439	169,049

18/12/2014; 0 m; 3 µm	3300011171	4,042,438,499	70,990,421	131,521
18/12/2014; 0 m; 0.8 µm	3300012116	4,913,028,941	88,575,811	175,581
18/12/2014; 0 m; 0.1 µm	3300012128	4,381,365,470	107,097,935	226,093
18/12/2014; 0 m; <0.1 µm	3300012110	3,925,090,941	78,340,986	155,340
19/1/2015; 0 m; 3 µm	3300012109	4,088,535,832	76,942,819	150,691
19/1/2015; 0 m; 0.8 µm	3300012028	4,177,959,071	84,152,334	167,030
19/1/2015; 0 m; <0.1 µm	3300012118	4,427,446,802	95,235,823	191,329
Club Lake				
26/11/2014; 0 m; 3 µm	3300012108	4,870,608,443	83,416,154	150,162
26/11/2014; 0 m; 0.8 µm	3300012261	3,341,521,330	71,167,471	128,680
26/11/2014; 0 m; 0.1 µm	3300012263	3,759,312,452	86,361,097	176,426
26/11/2014; 0 m; <0.1 µm	3300012114	4,118,186,192	84,310,686	168,029
Organic Lake				
24/12/2006; 0 m; 3 µm	3300017901	191,828,606	43,859,466	87,032
24/12/2006; 0 m; 0.8 µm	3300017457	172,712,662	11,287,428	22,692
24/12/2006; 0 m; 0.1 µm	3300017534	169,560,274	8,636,023	19,826
Rauer Lake 1				
11/1/2015; 0 m; 3 µm	3300012272	3,744,837,536	317,629,528	602,146
11/1/2015; 0 m; 0.8 µm	3300011188	4,121,482,920	339,573,373	631,444
11/1/2015; 0 m; 0.1 µm	3300012147	5,060,314,613	153,987,876	335,745
Rauer Lake 3				
11/1/2015; 0 m; 3 µm	3300012033	4,113,662,935	282,086,351	502,534
11/1/2015; 0 m; 0.8 µm	3300012268	2,989,166,916	171,063,290	315,976
11/1/2015; 0 m; 0.1 µm	3300012267	3,678,764,556	157,623,777	314,905
Rauer Lake 6				
11/1/2015; 0 m; 3 µm	3300012182	4,850,916,256	338,478,872	630,884
11/1/2015; 0 m; 0.8 µm	3300012178	4,380,578,636	319,376,726	576,497
11/1/2015; 0 m; 0.1 µm	3300011189	4,273,562,577	329,351,461	632,272
Rauer Lake 11				
11/1/2015; 0 m; 3 µm	3300012270	4,491,510,058	269,022,148	521,320
Rauer Lake 13				
11/1/2015; 0 m; 3 µm	3300011187	3,876,060,564	244,559,929	494,693
11/1/2015; 0 m; 0.8 µm	3300011185	3,632,173,890	201,429,522	385,466
11/1/2015; 0 m; 0.1 µm	3300012170	4,252,101,604	251,362,283	492,062
Spades-assembled metagenomes				

Ace Lake				
20/12/2006; 5 m; 3 µm	3300028202	65,944,407	9,717,163	18,015
20/12/2006; 5 m; 0.8 µm	3300028221	188,760,566	27,952,213	53,952
20/12/2006; 5 m; 0.1 µm	3300028228	514,425,517	33,518,956	64,687
20/12/2006; 11.5 m; 3 µm	3300028205	152,109,562	22,138,314	39,285
20/12/2006; 11.5 m; 0.8 µm	3300028289	194,556,802	16,906,227	32,171
20/12/2006; 11.5 m; 0.1 µm	3300028222	501,692,433	29,126,306	60,086
20/12/2006; 12.7 m; 3 µm	3300028203	83,214,739	10,703,483	20,757
20/12/2006; 12.7 m; 0.8 µm	3300028201	208,538,507	11,925,309	23,740
20/12/2006; 12.7 m; 0.1 µm	3300028204	240,290,391	6,971,450	13,087
20/12/2006; 14 m; 3 µm	3300028200	118,655,678	15,468,656	31,907
20/12/2006; 14 m; 0.8 µm	3300028302	165,208,287	27,504,336	56,468
20/12/2006; 14 m; 0.1 µm	3300028219	169,703,894	23,317,396	54,216
20/12/2006; 18 m; 3 µm	3300028199	114,460,928	12,486,049	26,210
20/12/2006; 18 m; 0.8 µm	3300028227	214,177,665	34,270,862	71,009
20/12/2006; 18 m; 0.1 µm	3300028216	145,906,502	15,860,072	40,100
20/12/2006; 23 m; 3 µm	3300028292	105,388,116	11,819,279	24,794
20/12/2006; 23 m; 0.8 µm	3300028226	231,162,768	33,899,871	71,413
20/12/2006; 23 m; 0.1 µm	3300028296	292,220,289	26,024,886	62,208
19/11/2008; 5 m; 3 µm	3300025601	10,168,447,444	374,845,559	637,417
19/11/2008; 5 m; 0.8 µm	3300025513	8,608,322,293	358,461,005	555,436
19/11/2008; 5 m; 0.1 µm	3300025425	9,326,252,194	190,824,688	354,920
21/11/2008; 11.8 m; 3 µm	3300025502	9,958,328,840	309,922,874	529,432
21/11/2008; 11.8 m; 0.8 µm	3300025603	10,372,524,015	387,727,814	649,215
21/11/2008; 11.8 m; 0.1 µm	3300025438	8,652,779,583	208,281,887	381,283
21/11/2008; 12.8 m; 3 µm	3300025433	7,377,945,147	191,332,554	330,516
21/11/2008; 12.8 m; 0.8 µm	3300025380	7,969,400,898	118,925,863	224,047
21/11/2008; 12.8 m; 0.1 µm	3300025362	15,030,492,867	90,472,821	190,960
21/11/2008; 14.1 m; 3 µm	3300025649	8,878,877,148	403,510,882	775,430
21/11/2008; 14.1 m; 0.8 µm	3300025628	9,024,438,900	379,168,081	728,210
21/11/2008; 14.1 m; 0.1 µm	3300025697	7,433,358,222	401,517,242	923,143
21/11/2008; 18 m; 3 µm	3300025642	9,701,518,914	444,311,389	775,322
21/11/2008; 18 m; 0.8 µm	3300025586	10,550,636,481	338,938,472	589,716
21/11/2008; 18 m; 0.1 µm	3300025669	8,489,799,212	415,535,816	832,930
23/11/2008; 23 m; 3 µm	3300025698	8,926,498,848	428,043,704	894,948

23/11/2008; 23 m; 0.8 µm	3300025661	8,835,913,368	414,688,901	822,281
23/11/2008; 23 m; 0.1 µm	3300025736	8,391,237,271	477,169,979	1,113,701
24/11/2013; 5 m; 3 µm	3300022867	4,225,013,370	144,719,058	289,211
24/11/2013; 5 m; 0.8 µm	3300023243	4,462,325,958	205,826,389	369,592
24/11/2013; 5 m; 0.1 µm	3300022843	3,805,948,564	100,883,143	212,850
25/11/2013; 12.5 m; 3 µm	3300022842	4,534,814,707	163,226,887	302,245
25/11/2013; 12.5 m; 0.8 µm	3300022847	4,208,778,962	155,718,155	244,054
25/11/2013; 12.5 m; 0.1 µm	3300023235	4,703,733,094	143,622,133	282,929
26/11/2013; 13.5 m; 3 µm	3300022882	4,632,992,773	197,528,912	370,963
26/11/2013; 13.5 m; 0.8 µm	3300023244	4,017,414,066	152,968,368	281,280
26/11/2013; 13.5 m; 0.1 µm	3300022871	4,289,343,500	153,918,125	304,781
26/11/2013; 15 m; 3 µm	3300023234	2,830,397,582	132,062,988	251,704
26/11/2013; 15 m; 0.8 µm	3300022854	4,179,971,653	189,382,169	349,194
26/11/2013; 15 m; 0.1 µm	3300023435	3,982,384,098	204,889,614	458,784
26/11/2013; 19 m; 3 µm	3300023298	3,861,886,442	173,692,067	351,338
26/11/2013; 19 m; 0.8 µm	3300023262	5,356,530,473	256,708,329	493,455
26/11/2013; 19 m; 0.1 µm	3300023297	4,526,133,618	236,042,504	568,485
27/11/2013; 24 m; 3 µm	3300022828	2,032,322,733	65,695,823	149,469
27/11/2013; 24 m; 0.8 µm	3300022887	4,489,480,975	197,136,157	423,504
27/11/2013; 24 m; 0.1 µm	3300031227	21,163,513,792	1,050,144,399	2,413,590
17/12/2014; 0 m; 3 µm	3300022841	3,505,709,238	109,878,484	205,134
17/12/2014; 0 m; 0.8 µm	3300022833	3,007,301,388	112,095,376	172,874
17/12/2014; 0 m; 0.1 µm	3300022822	3,926,440,146	72,848,168	141,301
15/02/2014; 0 m; 3 µm	3300022827	4,445,471,441	150,261,289	262,344
15/02/2014; 0 m; 0.8 µm	3300023054	4,101,153,533	186,668,359	262,345
15/02/2014; 0 m; 0.1 µm	3300022839	4,105,154,760	94,401,441	195,630
2/07/2014; 5 m; 3 µm	3300023237	4,712,346,032	179,194,270	291,313
2/07/2014; 5 m; 0.8 µm	3300022866	4,450,973,256	227,490,836	403,969
2/07/2014; 5 m; 0.1 µm	3300022853	4,388,723,345	128,153,264	250,568
3/07/2014; 12.5 m; 3 µm	3300022857	3,349,508,936	162,523,775	274,815
3/07/2014; 12.5 m; 0.8 µm	3300022836	3,812,123,689	173,061,625	297,508
3/07/2014; 12.5 m; 0.1 µm	3300023245	4,389,831,560	141,659,134	285,227
3/07/2014; 13.5 m; 3 µm	3300022834	3,025,335,676	150,334,734	279,053
3/07/2014; 13.5 m; 0.8 µm	3300023241	3,917,460,255	176,108,874	316,827
3/07/2014; 13.5 m; 0.1 µm	3300023257	4,754,144,028	246,566,898	516,984

20/08/2014; 5 m; 3 µm	3300023236	3,535,315,349	145,971,573	260,745
20/08/2014; 5 m; 0.8 µm	3300023239	3,675,443,392	161,858,999	300,236
20/08/2014; 5 m; 0.1 µm	3300023229	3,581,244,138	112,903,843	219,283
21/08/2014; 13 m; 3 µm	3300022885	4,805,699,185	232,800,896	422,661
21/08/2014; 13 m; 0.8 µm	3300022845	3,046,800,658	127,278,965	240,608
21/08/2014; 13 m; 0.1 µm	3300023296	4,126,784,684	163,100,043	305,555
21/08/2014; 14.5 m; 3 µm	3300022864	4,208,293,249	203,541,480	379,585
21/08/2014; 14.5 m; 0.8 µm	3300024048	4,438,778,032	185,710,747	327,952
21/08/2014; 14.5 m; 0.1 µm	3300022890	3,761,803,592	196,439,047	427,804
20/10/2014; 5 m; 3 µm	3300022865	3,718,130,970	159,691,784	283,171
20/10/2014; 5 m; 0.8 µm	3300022825	3,500,964,757	137,992,510	261,144
20/10/2014; 5 m; 0.1 µm	3300023294	4,051,255,334	135,330,843	259,473
20/10/2014; 12 m; 3 µm	3300022848	3,461,486,260	157,234,838	316,382
20/10/2014; 12 m; 0.8 µm	3300023238	3,185,298,810	140,908,866	262,229
20/10/2014; 12 m; 0.1 µm	3300023240	3,685,976,302	125,847,023	262,910
21/10/2014; 13 m; 3 µm	3300022856	3,793,702,914	185,885,369	366,842
21/10/2014; 13 m; 0.8 µm	3300022859	3,615,901,126	148,572,713	281,988
21/10/2014; 13 m; 0.1 µm	3300022821	3,169,765,298	119,795,036	247,086
21/10/2014; 16 m; 3 µm	3300022855	2,823,639,110	137,224,766	262,841
21/10/2014; 16 m; 0.8 µm	3300023249	3,472,734,434	161,447,324	294,441
21/10/2014; 16 m; 0.1 µm	3300022858	3,214,387,734	162,887,351	368,840
21/10/2014; 19 m; 3 µm	3300023434	3,699,374,508	165,008,949	330,503
21/10/2014; 19 m; 0.8 µm	3300022838	3,195,707,102	158,062,637	299,108
21/10/2014; 19 m; 0.1 µm	3300023246	3,202,188,919	153,939,570	372,354
21/10/2014; 24 m; 3 µm	3300023251	3,707,575,608	149,036,067	306,831
21/10/2014; 24 m; 0.8 µm	3300023295	4,015,996,994	166,137,713	367,296
21/10/2014; 24 m; 0.1 µm	3300022874	3,523,521,042	181,923,112	450,383
4/12/2014; 5 m; 3 µm	3300023501	3,558,906,481	126,636,802	250,738
4/12/2014; 5 m; 0.8 µm	3300022844	3,528,199,602	163,618,968	306,086
4/12/2014; 5 m; 0.1 µm	3300023293	3,287,944,538	81,894,154	178,097
4/12/2014; 12 m; 3 µm	3300023231	3,372,774,996	116,441,688	240,321
4/12/2014; 12 m; 0.8 µm	3300023227	3,766,666,990	103,396,553	207,492
4/12/2014; 12 m; 0.1 µm	3300022851	3,582,064,538	119,299,278	248,470
4/12/2014; 13.4 m; 3 µm	3300031697	14,149,086,706	400,324,806	718,959
4/12/2014; 13.4 m; 0.8 µm	3300022826	2,989,229,242	78,299,135	145,800

4/12/2014; 13.4 m; 0.1 µm	3300023292	3,878,932,484	85,111,111	181,733
4/12/2014; 14 m; 3 µm	3300023253	3,420,681,173	167,955,693	307,470
4/12/2014; 14 m; 0.8 µm	3300023233	3,250,064,514	144,877,168	252,928
4/12/2014; 14 m; 0.1 µm	3300022868	3,895,509,417	195,190,896	414,173
3/12/2014; 19 m; 3 µm	3300022860	4,079,964,767	181,977,179	369,802
3/12/2014; 19 m; 0.8 µm	3300022846	3,983,828,178	165,102,958	309,999
3/12/2014; 19 m; 0.1 µm	3300023061	3,209,269,596	152,256,002	384,107
3/12/2014; 24 m; 3 µm	3300022884	4,021,442,672	179,261,304	381,611
3/12/2014; 24 m; 0.8 µm	3300023299	5,006,350,890	217,304,898	440,798
3/12/2014; 24 m; 0.1 µm	3300023256	3,621,396,862	179,844,837	445,634
8/01/2015; 0 m; 3 µm	3300022829	3,645,848,765	78,301,103	152,629
8/01/2015; 0 m; 0.8 µm	3300022832	3,757,499,746	136,667,441	270,106
8/01/2015; 0 m; 0.1 µm	3300023242	3,407,544,904	121,628,756	269,881
27/01/2015; 0 m; 3 µm	3300023230	3,829,689,694	116,684,467	219,301
27/01/2015; 0 m; 0.8 µm	3300023429	3,298,326,784	165,138,532	262,012
27/01/2015; 0 m; 0.1 µm	3300022837	3,616,258,196	93,765,159	194,928
Deep Lake				
1/12/2006; 0 m; 0.1 µm	3300025352	4,982,821,567	68,214,218	125,418
24/11/2014; 0 m; <0.1 µm	3300028353	3,544,797,704	50,177,156	82,167
Ellis Fjord				
9/10/2014; 5 m; 3 µm	3300031658	22,583,676,006	512,812,684	1,103,847
9/10/2014; 5 m; 0.8 µm	3300031629	17,208,422,590	806,100,132	1,620,667
9/10/2014; 5 m; 0.1 µm	3300031659	22,404,979,271	880,035,495	1,841,661
9/10/2014; 45 m; 3 µm	3300031631	16,642,938,015	430,745,788	939,013
9/10/2014; 45 m; 0.8 µm	3300031741	18,009,360,230	575,028,230	1,095,421
9/10/2014; 45 m; 0.1 µm	3300031603	13,346,656,379	637,704,446	1,320,380
8/10/2014; 60 m; 3 µm	3300031645	19,804,223,504	908,736,180	1,987,014
8/10/2014; 60 m; 0.8 µm	3300031657	15,894,964,153	433,198,916	809,885
8/10/2014; 60 m; 0.1 µm	3300031601	17,233,115,763	295,045,072	509,553
2/11/2014; 18 m; 3 µm	3300031602	18,753,292,708	211,651,588	361,601
2/11/2014; 18 m; 0.8 µm	3300031660	15,932,943,833	734,660,849	1,350,693
2/11/2014; 18 m; 0.1 µm	3300031696	16,951,600,293	508,273,244	909,436
Taynaya Bay†				
28/11/2014; 5 m; 0.22 µm	3300038912	11,415,818,068	256,201,821	474,896

28/11/2014; 5 m; 0.22 µm-nbfc	3300038786		675,093,048	1,399,527
28/11/2014; 11 m; 0.22 µm	3300039187		439,736,431	928,219
28/11/2014; 11 m; 0.22 µm-nbfc	3300039186	12,286,944,772	1,039,896,900	2,382,412

Table A2. List of *Synechococcus* and *Chlorobium* MAGs from stratified systems in the Vestfold Hills. The table includes description of *Chlorobium* MAGs from Ace Lake, Ellis Fjord, and Taynaya Bay as well *Synechococcus* MAGs from Ace Lake. [A] The high and medium quality MAGs were generated by JGI's IMG system from the Antarctic metagenomes (Table A1) mentioned in the first column. [B] The high-quality bins are highlighted with a green background in the column. [C] The values highlighted in red indicate that the bin contamination was >1%. The bin contamination of all *Chlorobium* MAGs was <3% and that of *Synechococcus* MAGs was ≤4%. The IMG MAGs were used for analysis of *Synechococcus* in Chapter 4 and *Chlorobium* in Chapter 5. Filter fractions: 3, 3–20 µm; 0.8, 0.8–3 µm; 0.1, 0.1–0.8 µm; 0.22, 0.22–20 µm.

Metagenome[A]	IMG Bin IDs[B]	Bin completeness (%)[C]	Total base pair count (bp)	Gene count	Scaffold count
Ace Lake *Chlorobium* MAGs					
20/12/2006; 12.7 m; 3 µm	3300028203_1	98	1,799,622	1968	32
20/12/2006; 12.7 m; 0.8 µm	3300028201_1	98	1,846,253	1956	17
20/12/2006; 14 m; 0.8 µm	3300028302_2	95	1,719,822	2066	58
20/12/2006; 18 m; 0.8 µm	3300028227_2	68	1,219,845	1799	195
21/11/2008; 12.8 m; 3 µm	3300025433_15	87	1,561,142	1554	27
21/11/2008; 12.8 m; 0.8 µm	3300025380_8	60	915,115	905	13
21/11/2008; 12.8 m; 0.1 µm	3300025362_8	66	1,027,280	1021	10
21/11/2008; 14.1 m; 3 µm	3300025649_20	99	1,717,607	1718	37
21/11/2008; 14.1 m; 0.8 µm	3300025628_24	72	1,339,744	1315	21
21/11/2008; 14.1 m; 0.1 µm	3300025697_16	54	946,925	1090	184
21/11/2008; 18 m; 3 µm	3300025642_35	64	1,147,570	1132	20
21/11/2008; 18 m; 0.8 µm	3300025586_24	99	1,760,585	1740	23
21/11/2008; 18 m; 0.1 µm	3300025669_14	99	1,772,585	1750	20
23/11/2008; 23 m; 3 µm	3300025698_17	99	1,750,727	1739	22
23/11/2008; 23 m; 0.8 µm	3300025661_20	72	1,347,288	1345	27
26/11/2013; 13.5 m; 3 µm	3300022882_7	99	1,780,829	1765	28
26/11/2013; 13.5 m; 0.8 µm	3300023244_8	99	1,784,037	1767	28
26/11/2013; 13.5 m; 0.1 µm	3300022871_5	99	1,792,085	1781	27
26/11/2013; 15 m; 3 µm	3300023234_7	97	1,681,376	1733	109
26/11/2013; 15 m; 0.8 µm	3300022854_6	99	1,793,372	1777	23
26/11/2013; 15 m; 0.1 µm	3300023435_5	99	1,746,873	1742	27
26/11/2013; 19 m; 0.8 µm	3300023262_7	99	1,741,261	1742	34

27/11/2013; 24 m; 0.8 µm	3300022887_7	97	1,660,438	1732	141
27/11/2013; 24 m; 0.1 µm	3300031227_17	95	1,650,130	1765	134
3/07/2014; 13.5 m; 3 µm	3300022834_6	99	1,745,898	1735	26
3/07/2014; 13.5 m; 0.8 µm	3300023241_6	99	1,784,741	1776	30
3/07/2014; 13.5 m; 0.1 µm	3300023257_7	99	1,789,942	1779	23
21/08/2014; 14.5 m; 3 µm	3300022864_8	99	1,748,321	1738	23
21/08/2014; 14.5 m; 0.8 µm	3300024048_8	99	1,784,669	1772	24
21/08/2014; 14.5 m; 0.1 µm	3300022890_5	99	1,753,999	1747	19
21/10/2014; 13 m; 0.8 µm	3300022859_8	92	1,620,807	1696	120
21/10/2014; 13 m; 0.1 µm	3300022821_10	67	1,219,235	1436	207
21/10/2014; 16 m; 0.8 µm	3300023249_8	99	1,737,575	1754	52
21/10/2014; 19 m; 0.8 µm	3300022838_8	89	1,431,222	1555	180
21/10/2014; 24 m; 0.8 µm	3300023295_7	61	1,100,512	1254	200
4/12/2014; 12 m; 3 µm	3300023231_5	99	1,783,647	1765	23
4/12/2014; 12 m; 0.8 µm	3300023227_6	99	1,783,085	1763	26
4/12/2014; 12 m; 0.1 µm	3300022851_4	99	1,796,868	1770	19
4/12/2014; 13.4 m; 3 µm	3300031697_14	99	1,807,042	1791	33
4/12/2014; 13.4 m; 0.8 µm	3300022826_4	99	1,801,610	1778	31
4/12/2014; 13.4 m; 0.1 µm	3300023292_2	99	1,785,555	1760	28
4/12/2014; 14 m; 3 µm	3300023253_8	99	1,797,888	1783	22
4/12/2014; 14 m; 0.8 µm	3300023233_7	99	1,811,803	1791	24
4/12/2014; 14 m; 0.1 µm	3300022868_7	99	1,777,496	1761	26
3/12/2014; 19 m; 3 µm	3300022860_8	98	1,773,797	1750	22
3/12/2014; 19 m; 0.8 µm	3300022846_6	99	1,797,570	1772	29
3/12/2014; 19 m; 0.1 µm	3300023061_2	99	1,812,610	1797	27
3/12/2014; 24 m; 3 µm	3300022884_9	99	1,735,816	1755	69
3/12/2014; 24 m; 0.8 µm	3300023299_6	99	1,795,237	1771	22
3/12/2014; 24 m; 0.1 µm	3300023256_3	99	1,797,328	1785	22
Ellis Fjord *Chlorobium* MAGs					
9/10/2014; 5 m; 0.1 µm	3300031659_20	63	890,084	1006	170
9/10/2014; 45 m; 3 µm	3300031631_9	99	1,836,564	1807	32
9/10/2014; 45 m; 0.8 µm	3300031741_10	99	1,820,609	1801	31
9/10/2014; 45 m; 0.1 µm	3300031603_6	99	1,820,941	1799	33
8/10/2014; 60 m; 3 µm	3300031645_24	89	1,450,081	1532	187
8/10/2014; 60 m; 0.8 µm	3300031657_13	99	1,753,701	1756	34

8/10/2014; 60 m; 0.1 µm	3300031601_7	99	1,770,724	1775	46
Taynaya Bay *Chlorobium* MAGs					
28/11/2014; 5 m; 0.22 µm	3300038912_10	99	1,808,383	1834	57
28/11/2014; 5 m; 0.22 µm-nbfc	3300038786_10	99	1,805,285	1822	53
28/11/2014; 11 m; 0.22 µm	3300039187_7	99	1,822,415	1829	24
28/11/2014; 11 m; 0.22 µm-nbfc	3300039186_9	99	1,823,916	1829	22
Ace Lake *Synechococcus* MAGs					
20/12/2006; 5 m; 0.8 µm	3300028221_1	61	1,805,389	2389	223
20/12/2006; 11.5 m; 0.8 µm	3300028289_1	83	2,293,100	2808	188
19/11/2008; 5 m; 3 µm	3300025601_8	96	2,478,229	2673	42
19/11/2008; 5 m; 0.8 µm	3300025513_11	98	2,766,682	3058	80
21/11/2008; 11.8 m; 3 µm	3300025502_14	73	1,976,130	2157	52
21/11/2008; 11.8 m; 0.8 µm	3300025603_17	98	2,596,736	2878	76
21/11/2008; 12.8 m; 3 µm	3300025433_11	97	2,429,787	2704	110
21/11/2008; 12.8 m; 0.8 µm	3300025380_5	84	2,075,048	2435	270
21/11/2008; 14.1 m; 3 µm	3300025649_16	75	2,079,529	2260	72
21/11/2008; 14.1 m; 0.8 µm	3300025628_11	99	2,754,716	3055	93
21/11/2008; 18 m; 3 µm	3300025642_19	93	2,483,571	2756	105
21/11/2008; 18 m; 0.8 µm	3300025586_14	98	2,876,270	3212	142
23/11/2008; 23 m; 3 µm	3300025698_9	99	2,619,579	2928	81
23/11/2008; 23 m; 0.8 µm	3300025661_8	99	2,786,666	3096	98
25/11/2013; 12.5 m; 3 µm	3300022842_9	99	2,748,300	3057	95
25/11/2013; 12.5 m; 0.8 µm	3300022847_9	96	2,718,101	3040	122
26/11/2013; 13.5 m; 3 µm	3300022882_6	92	2,440,655	2779	190
26/11/2013; 13.5 m; 0.8 µm	3300023244_5	95	2,479,789	2794	142
26/11/2013; 15 m; 3 µm	3300023234_6	98	2,654,330	2998	123
26/11/2013; 15 m; 0.8 µm	3300022854_5	97	2,713,278	3009	106
26/11/2013; 19 m; 3 µm	3300023298_7	98	2,639,373	2963	121
26/11/2013; 19 m; 0.8 µm	3300023262_5	99	2,777,510	3106	103
27/11/2013; 24 m; 3 µm	3300022828_1	96	2,496,541	2815	149
27/11/2013; 24 m; 0.8 µm	3300022887_4	99	2,799,985	3135	91
2/07/2014; 5 m; 3 µm	3300023237_10	99.7	2,644,322	2929	64
2/07/2014; 5 m; 0.8 µm	3300022866_9	99.6	2,691,375	2964	59

3/07/2014; 12.5 m; 3 µm	3300022857_10	99.7	2,711,173	3001	60
3/07/2014; 12.5 m; 0.8 µm	3300022836_9	99.7	2,843,718	3156	80
3/07/2014; 13.5 m; 3 µm	3300022834_4	97	2,584,503	2887	129
3/07/2014; 13.5 m; 0.8 µm	3300023241_5	98	2,875,681	3230	145
20/08/2014; 5 m; 3 µm	3300023236_6	98	2,574,224	2895	136
20/08/2014; 5 m; 0.8 µm	3300023239_8	99.7	2,768,073	3045	72
21/08/2014; 13 m; 3 µm	3300022885_12	99	2,908,751	3230	75
21/08/2014; 13 m; 0.8 µm	3300022845_7	97	3,008,323	3375	144
21/08/2014; 14.5 m; 3 µm	3300022864_6	95	2,565,433	2897	144
21/08/2014; 14.5 m; 0.8 µm	3300024048_5	97	2,944,912	3295	160
20/10/2014; 5 m; 3 µm	3300022865_7	99.7	2,862,630	3153	74
20/10/2014; 5 m; 0.8 µm	3300022825_3	99.7	3,001,000	3323	93
20/10/2014; 12 m; 3 µm	3300022848_5	99.7	2,874,195	3194	79
20/10/2014; 12 m; 0.8 µm	3300023238_7	99	3,035,627	3363	104
21/10/2014; 13 m; 3 µm	3300022856_3	99	2,900,943	3236	98
21/10/2014; 13 m; 0.8 µm	3300022859_7	98	2,853,405	3167	133
21/10/2014; 16 m; 3 µm	3300022855_5	96	2,559,259	2865	132
21/10/2014; 16 m; 0.8 µm	3300023249_7	99	2,784,281	3124	118
21/10/2014; 19 m; 3 µm	3300023434_6	98	2,654,154	2971	112
21/10/2014; 19 m; 0.8 µm	3300022838_5	95	2,547,267	2867	150
21/10/2014; 24 m; 0.8 µm	3300023295_4	98	2,663,178	3017	138
4/12/2014; 5 m; 3 µm	3300023501_3	99	2,823,490	3136	87
4/12/2014; 5 m; 0.8 µm	3300022844_3	99.7	2,856,174	3163	68
4/12/2014; 12 m; 3 µm	3300023231_3	99	2,916,660	3228	82
4/12/2014; 12 m; 0.8 µm	3300023227_3	99.7	3,107,104	3456	104
4/12/2014; 13.4 m; 3 µm	3300031697_11	94	2,571,291	2882	135
4/12/2014; 14 m; 3 µm	3300023253_6	97	2,654,228	2956	120
4/12/2014; 14 m; 0.8 µm	3300023233_5	98	2,773,735	3110	133
3/12/2014; 19 m; 3 µm	3300022860_5	98	2,774,073	3102	111
3/12/2014; 19 m; 0.8 µm	3300022846_5	99	2,848,909	3184	106
3/12/2014; 24 m; 3 µm	3300022884_6	88	2,175,309	2549	207
3/12/2014; 24 m; 0.8 µm	3300023299_4	99	2,849,381	3166	95
27/01/2015; 0 m; 0.8 µm	3300023429_4	99	2,694,939	2977	60

Appendix B

Cavlab pipeline v1.2 — the preliminary metagenome analysis pipeline

Code B1. Python code for Cavlab pipeline v1.2. This code represents the preliminary Cavlab pipeline developed for the analysis of Antarctic metagenomes.

```
""" Original version of Cavlab pipeline created on Sun Sep 25 13:35:23 2016
@author: jay3
This is the head script for the metagenomics pipeline. It should be run from the JGI sample
folder. It depends on consistent folder structure (IMG_Data and QC_and_Genome_Assembly).

v1.2:
Decided to keep databases in a single folder rather than rewrite every run. Specified # of threads
in DIAMOND and BBMap lines. Minimum request job time is set to 12 hrs."""

from datetime import date
import os
import subprocess
import sys
import csv
current_dir = subprocess.check_output('pwd', shell = True).decode().strip() + '/' # get current dir

#### find reads file and save path
go = []
file_found = 0
if os.path.isdir('./QC_and_Genome_Assembly') == True:
    QC_folder = os.listdir('./QC_and_Genome_Assembly')
    if len(QC_folder) == 1:
        for file in os.listdir('./QC_and_Genome_Assembly/' + QC_folder[0]):
            if file[-5:] == 'fastq' and file.split('.')[1] == 'filtered':
                raw_num = file.split('.')[0]
                file_found = 1
                read_file = current_dir + 'QC_and_Genome_Assembly/' + QC_folder[0] + '/' + file
    else:
        print('QC_and_Genome_Assembly subfolder not found, freaking folder structures...')
else:
    print('QC_and_Genome_Assembly folder not found, somewhere over the rainbow... la la la')
if file_found == 0:
    print('reads not found, go back to kindergarten')
else:
    print('reads found 1up')
```

```
    go.append(1)
QC_dir = subprocess.check_output('ls ' + './QC_and_Genome_Assembly', shell =
True).decode('UTF-8') #as a cr/lf at the end...
raw_dir = current_dir + 'QC_and_Genome_Assembly/' + QC_dir

#### find assembly files and save paths
#### verify folder and ORFs file
file_found = 0
if os.path.isdir('./IMG_Data') == True:
    for file in os.listdir('./IMG_Data'):
        if file[-3:] == 'faa':
            ass_num = file.split('.')[0]
            file_found = 1
            protass_file = current_dir + 'IMG_Data/' + file
else:
    print('IMG_Data folder nowhere to be found')
if file_found == 0:
    print('assembly.faa not found, sad day')
else:
    print('assembly.faa found, off to a good start')
    go.append(1)

#### verify COG file
if os.path.isfile('./IMG_Data/' + ass_num + '.assembled.faa.COG') == 1:
    print('assembled.faa.COG file found, yay I guess')
    go.append(1)
    COG_file = current_dir + 'IMG_Data/' + ass_num + '.assembled.faa.COG'
else:
    print('assembled.faa.COG file not found, who cares about COG anyway?')

#### verify KEGG file
if os.path.isfile('./IMG_Data/' + ass_num + '.assembled.faa.KO') == 1:
    print('assembled.faa.KO file found, ((((((((((()==<<<')
    go.append(1)
    KEGG_file = current_dir + 'IMG_Data/' + ass_num + '.assembled.faa.KO'
else:
    print('assembled.faa.KO file not found, :( thats probably going to leave a mark')
```

```
#### verify DNA assembly file
if os.path.isfile('./IMG_Data/' + ass_num + '.assembled.fna') == 1:
    print('assembled.fna file found this is getting good')
    go.append(1)
    DNAass_file = current_dir + 'IMG_Data/' + ass_num + '.assembled.fna'
else:
    print('assembled.fna file not found, so close yet so far away')

#### verify coverage file
if os.path.isfile('./IMG_Data/' + raw_num + '.scaffolds.cov') == 1:
    print('scaffolds.cov file found, wait for it...')
    go.append(1)
    COV_file = current_dir + 'IMG_Data/' + raw_num + '.scaffolds.cov'
else:
    print('scaffolds.cov file not found, da da da Im sorry; the caller you are trying to reach can not
be located. Please hang up and try your call again later.')

#### verify scaffold to contig mapping file
if os.path.isfile('./IMG_Data/' + ass_num + '.assembled.names_map') == 1:
    print('assembled.names_map file found this is getting good')
    go.append(1)
    MAP_file = current_dir + 'IMG_Data/' + ass_num + '.assembled.names_map'
else:
    print('assembled.names_map file not found, so close yet so far away')

#### verify resource files are available
if
os.path.isfile('/srv/scratch/jgi/Cavlab_pipeline_resources/v1/K00394_pathway_database_v1.fast
a') == 1:
    print('K00394_pathway_database_v1.fasta found')
    go.append(1)
else:
    print('K00394_pathway_database_v1.fasta not found')
if
os.path.isfile('/srv/scratch/jgi/Cavlab_pipeline_resources/v1/K00395_pathway_database_v1.fast
a') == 1:
```

```
    print('K00395_pathway_database_v1.fasta found')
    go.append(1)
else:
    print('K00395_pathway_database_v1.fasta not found')
if
os.path.isfile('/srv/scratch/jgi/Cavlab_pipeline_resources/v1/K11180_pathway_database_v1.fast
a') == 1:
    print('K11180_pathway_database_v1.fasta found')
    go.append(1)
else:
    print('K11180_pathway_database_v1.fasta not found')
if
os.path.isfile('/srv/scratch/jgi/Cavlab_pipeline_resources/v1/K11181_pathway_database_v1.fast
a') == 1:
    print('K11181_pathway_database_v1.fasta found')
    go.append(1)
else:
    print('K11181_pathway_database_v1.fasta not found')
if os.path.isfile('/srv/scratch/jgi/Cavlab_pipeline_resources/v1/COG_conversion_v1.csv') == 1:
    print('COG_conversion_v1.csv found')
    go.append(1)
else:
    print('COG_conversion_v1.csv not found')

#### decide if all files are found and proceed
if sum(go) == 12:
    print('all systems are GO for launch!')
else:
    print('HOLD HOLD HOLD')
    print('launch scrubbed, new launch window TBD')
    sys.exit()

#### make head folder and sub folders
now =date.today()
head_folder = 'Cav_' + str(now.year)[-2:] + str(now.month) + str(now.day)
subprocess.call('mkdir ' + head_folder, shell = True)
subprocess.call('mkdir ' + head_folder + '/resources', shell = True)
```

```
subprocess.call('mkdir ' + head_folder + '/metabat', shell = True)
#subprocess.call('cd ' + head_folder, shell = True)
head_dir = current_dir + head_folder + '/'
res_dir = head_dir + 'resources/'

#### write readme file
readme_text = """This is the head folder created from the Cav pipeline v1.2 on the %s day of the %s month of %s. It contains the results from and resources used by the pipeline.
/resources contains sulfur databases used in the processing of KEGG markers K00394, K00395, K11180, and K11181, and a COG conversion file. Jobs_log.txt has a record of the individual jobs created as part of the pipeline. Each entry corresponds to a SCRATCH######## file which contains the screen log of the job. Email reports are found in rcavlab@gmail.com. Error reports and output reports also correspond to the various jobs. All jobs were run on Katana, the UNSW science computing cluster.
/metabat contains files related to MetaBAT processing. Some were created by this run while others were created by other operations related to MetaBAT processing.
/phylosift is the output from PhyloSift and contains diversity related information.
.rma is a MEGAN file which is primarily used for taxonomy. However, this file has some COG information, but these assignments may differ from those used in this pipeline. This is because COG data used in this pipeline originates as a JGI file. The data in the MEGAN file comes from diamond (a faster version of BLAST).
COG_summary.csv is the summary of COG categories. The "by coverage" column is weighted by read depth of the ORF and "by count" is simply the fraction of counts in each COG category. The "total" row represents the total number of ORFs in the faa.COG file for the "by count" column and the sum of read depths for each ORF in the faa.COG file. These values were used to normalize each of the categories. The "issues" row represents ORFs that couldn't be placed in a category or didn't identify a coverage value.
KO_summary.csv is the KEGG pathways summary. It aggregates markers into pathways and the columns are the same as for COG. The "total" row is the same as COG. The "issues" row represents ORFs that couldn't identify a coverage value. Below the pathway rows, each marker is saved individually.
This is version 1.2 of the Cavlab pipeline and uses:
phylosift v1.0.1
 perl v5.20.1
 hmmer v3.1b2
 raxml v8.1.17
 fasttree v2.1.7
```

```
  pplacer v1.1.alpha16
diamond v0.8.4
  nr database ~June 1, 2016
MEGAN v6.4.5
  java v8u45
COG_sumarize_v2.py
  python v3.5.2
KOpathways_v8.py
  python v3.5.2
bbmap v35.82
assemblies_filter_v1.py
  python v3.5.2

At the time of writing this pipeline (October 2016) there is no documentation.
However, I intend to include a detailed description of the development and testing in my thesis.
I will try to add text to the bottom of the file as this information becomes available. The pipeline
was developed by James "Jay" Bevington on behalf of the Cavicchioli lab in the School of
Biotechnology and Biomolecular Sciences at the University of New South Wales. If you are still
reading, I assume you are in deep trouble... I (Jay) stand by my work and am happy to answer
questions long into the future. Please contact Dr. Rick Cavicchioli at r.cavicchioli@unsw.edu.au
(+612) 9385 3516 or Jay Bevington at jbevingt@gmail.com +61 401 096 241 or +1-985-789-
3511. Future amendments to be included below:
""" %(str(now.day), str(now.month), str(now.year))
with open(head_dir + 'readme.txt' , 'w') as readme_file:
    readme_file.write(readme_text)

#### write bash scripts and python codes
#### phylosift
phylosift_script = """#!/bin/bash
#PBS -N SCRATCH
#PBS -l nodes=1:ppn=1
#PBS -l vmem=24gb
#PBS -l walltime=200:00:00
#PBS -o %sPhylosift_Output_report_1
#PBS -o %sPhylosift_Error_report_1
#PBS -M rcavlab@gmail.com
#PBS -m ae
```

```
cd %s
module load perl/5.20.1
module load hmmer/3.1b2
module load raxml/8.1.17
module load fasttree/2.1.7
module load pplacer/1.1.alpha16
module load phylosift/1.0.1

phylosift all %s --out %sphylosift --paired
guppy fpd -o %sdiversity_table --theta 0.25,0.5 %sphylosift/%s.jplace
"""%(res_dir, res_dir, head_dir, read_file, head_dir, head_dir, head_dir, read_file.split('/')[-1])

with open(res_dir + 'phylosift.pbs', 'w') as phylosift_bash:
    phylosift_bash.write(phylosift_script)

#### diamond and MEGAN
#### python script to append coverage to ORF name
append_cov_code = """# -*- coding: utf-8 -*-
''' Created on Fri May 27 13:53:10 2016
@author: Jay2
Reverse adapted from Reads_cov_multi_v2 equivilant to Reads_cov_v3.py. Will append
coverage information to assembly ORFs for incorporating coverage info into MEGAN.'''

import csv
import Bio.SeqIO as SeqIO
#### read raw cov file
cov_name = []
cov = []
with open('%s', 'r') as rawfile:
    rawdata = csv.reader(rawfile, delimiter = '\t')
    for row in rawdata:
        cov_name.append(row[0])
        cov.append(row[1])
rawfile.close()
cov_name = cov_name[1:]
cov = cov[1:]
```

```
#### read map file
maps = {}
with open('%s', 'r') as map_file:
    map_csv = csv.reader(map_file, delimiter = '\t')
    for row in map_csv:
        maps[row[0]] = row[1]

#### build cov info w/ correct name
coverage = []
for i in range(len(cov_name)):
    coverage.append([maps[cov_name[i]], cov[i]])
name_len = len(coverage[0][0])

#### find cov info and write to file
index=0
with open('%s', 'r') as read_file:
    with open('%s.assembled_cov.faa', 'w') as newfile:
        for record in SeqIO.parse(read_file, "fasta"): # for each read
            for j in range(index,len(coverage)): #for each coverage line
                cov_namei = coverage[j][0]
                if record.id[0:name_len] == coverage[j][0]: #if the line is for the read
                    record.id = record.id + '|magnitude=' + coverage[j][1] #append wts value
                    SeqIO.write(record,newfile, 'fasta') #write to file
                    index = j #store index as new starting point to avoid scanning
                    break # skip the rest of the list

#### submit jobs as part of the Cavlab pipeline
import subprocess
command = 'qsub ' + '%s' + 'phylosift.pbs'
screen = subprocess.check_output(command,shell = True)
screen = screen.decode()[0:7]
with open('%sjob_log.txt', 'a') as job_log:
    job_csv = csv.writer(job_log)
    job_csv.writerow(['phylosift', screen])

command = 'qsub ' + '%s' + 'diamondp.pbs'
screen = subprocess.check_output(command, shell = True)
```

```
screen = screen.decode()[0:7]
with open('%sjob_log.txt', 'a') as job_log:
    job_csv = csv.writer(job_log)
    job_csv.writerow(['diamondp_MEGAN', screen])

command = 'qsub ' + '%s' + 'COGKEGG.pbs'
screen = subprocess.check_output(command, shell = True)
screen = screen.decode()[0:7]
with open('%sjob_log.txt', 'a') as job_log:
    job_csv = csv.writer(job_log)
    job_csv.writerow(['COG_KEGG', screen])

command = 'qsub ' + '%s' + 'sample2500_map.pbs'
screen = subprocess.check_output(command, shell = True)
screen = screen.decode()[0:7]
with open('%sjob_log.txt', 'a') as job_log:
    job_csv = csv.writer(job_log)
    job_csv.writerow(['metabat_sample_map', screen])""" %(COV_file, MAP_file, protass_file,
head_dir + ass_num, res_dir, res_dir, res_dir, res_dir, res_dir, res_dir, res_dir, res_dir)

with open(res_dir + 'append_cov2ORFs.py', 'w') as append_cov2ORFs_script:
    append_cov2ORFs_script.write(append_cov_code)
protass_file = head_dir + ass_num + '.assembled_cov.faa' # Use the file with cov appended from
now on

#### write diamond/MEGAN bash
megan_script = """#!/bin/bash
#PBS -N SCRATCH
#PBS -l nodes=1:ppn=8
#PBS -l vmem=63gb
#PBS -l walltime=48:00:00
#PBS -o %sDiamondp_Output_Report_1
#PBS -o %sDiamondp_Error_Report_1
#PBS -M rcavlab@gmail.com
#PBS -m ae
cd %s
module load diamond/0.8.4
```

```
diamond blastp -d /srv/scratch/jgi/Cavlab_pipeline_resources/v1/nr -q %s -a diamondp.daa -e
0.001 -p 8
diamond view -a diamondp.daa -o %s.diamondp.tab -f tab

module load java/8u45
module load megan/6.4.5
export _JAVA_OPTIONS="-Xmx55g"

blast2rma -r %s -i %s.diamondp.tab -o %s.diamondp.rma -g2t
/srv/scratch/jgi/Cavlab_pipeline_resources/v1/gi_taxid_prot.dmp.gz -a2eggnog
/srv/scratch/jgi/Cavlab_pipeline_resources/v1/acc2eggnog-June2016X.abin -f BlastTab -mag -
fun EGGNOG

rm diamondp.daa
""" %(res_dir, res_dir, head_dir, protass_file, ass_num, protass_file, ass_num, ass_num)

with open(res_dir + 'diamondp.pbs', 'w') as diamondp_bash:
    diamondp_bash.write(megan_script)

#### write COG and KEGG files
#### write COG script
COG_code = """# -*- coding: utf-8 -*-
''' Created on Fri Oct 14 17:35:49 2016
@author: jay3
Needs COG_Conversion.csv. Adds COG category to IMG data (.faa.COG) using the COG
number. Also, drops other columns reducing file size by ~half. Writes a new file with counts of
each category. This code should be placed in the directory with: the data file ___.faa.COG
(change file name below) and COG_Conversion.csv.
Some COG numbers have multiple category assignments. COG conversion assume that the first
assignment listed is the best. U was probably the most disturbed by this assumption.
v1
This script was reworked and updated from COG_conversion, added with statements for files.
v2:
Added inclusion of coverage information; fully tested (161025)'''

print('start COG')
```

```
import csv
import Bio.SeqIO as SeqIO
file_root = '%s'
assembly_file = '%s'
COG_file = '%s'

#### read ORF coverages from linemag.faa
coverage = []
cov_norm = 0
with open(assembly_file, 'r') as orf_file:
    for read_record in SeqIO.parse(orf_file, "fasta"):
        string = read_record.id
        string2 = string.split('|')
        coverage.append([string2[0], float(string2[1][10:])])
        cov_norm = cov_norm + float(string2[1][10:])
count_norm = len(coverage)
name_len = len(string2[0])

#### read in conversion file
reader =
csv.reader(open('/srv/scratch/jgi/Cavlab_pipeline_resources/v1/COG_conversion_v1.csv', 'r'))
d = {}
for row in reader:
   k, v = row
   d[k] = v

#### read in data COG numbers and make conversion
A = [0]
B = [0]
C = [0]
D = [0]
E = [0]
F = [0]
G = [0]
H = [0]
I = [0]
J = [0]
```

```
K = [0]
L = [0]
M = [0]
N = [0]
O = [0]
P = [0]
Q = [0]
R = [0]
S = [0]
T = [0]
U = [0]
V = [0]
Y = [0]
Z = [0]
other = []

index = 0
with open(COG_file, 'r') as DataRaw_file:
    DataRaw_csv = csv.reader(DataRaw_file, delimiter = '\t')
    for row in DataRaw_csv:
        ContigName = row[0]
        COGNum = row[1]
        COGCat = d[COGNum]
        err = []
        if COGCat == 'A':
            err = 1
            for j in range(index,len(coverage)):
                if row[0][0:name_len] == coverage[j][0]:
                    A.append(coverage[j][1]) #adds cov value to list
                    err = 0
                    break
        elif COGCat == 'B':
            err = 1
            for j in range(index,len(coverage)):
                if row[0][0:name_len] == coverage[j][0]:
                    B.append(coverage[j][1]) #adds cov value to list
                    err = 0
```

```
            break
    elif COGCat == 'C':
        err = 1
        for j in range(index,len(coverage)):
            if row[0][0:name_len] == coverage[j][0]:
                C.append(coverage[j][1]) #adds cov value to list
                err = 0
                break
    elif COGCat == 'D':
        err = 1
        for j in range(index,len(coverage)):
            if row[0][0:name_len] == coverage[j][0]:
                D.append(coverage[j][1]) #adds cov value to list
                err = 0
                break
    elif COGCat == 'E':
        err = 1
        for j in range(index,len(coverage)):
            if row[0][0:name_len] == coverage[j][0]:
                E.append(coverage[j][1]) #adds cov value to list
                err = 0
                break
    elif COGCat == 'F':
        err = 1
        for j in range(index,len(coverage)):
            if row[0][0:name_len] == coverage[j][0]:
                F.append(coverage[j][1]) #adds cov value to list
                err = 0
                break
    elif COGCat == 'G':
        err = 1
        for j in range(index,len(coverage)):
            if row[0][0:name_len] == coverage[j][0]:
                G.append(coverage[j][1]) #adds cov value to list
                err = 0
                break
    elif COGCat == 'H':
```

```
    err = 1
    for j in range(index,len(coverage)):
        if row[0][0:name_len] == coverage[j][0]:
            H.append(coverage[j][1]) #adds cov value to list
            err = 0
            break
elif COGCat == 'I':
    err = 1
    for j in range(index,len(coverage)):
        if row[0][0:name_len] == coverage[j][0]:
            I.append(coverage[j][1]) #adds cov value to list
            err = 0
            break
elif COGCat == 'J':
    err = 1
    for j in range(index,len(coverage)):
        if row[0][0:name_len] == coverage[j][0]:
            J.append(coverage[j][1]) #adds cov value to list
            err = 0
            break
elif COGCat == 'K':
    err = 1
    for j in range(index,len(coverage)):
        if row[0][0:name_len] == coverage[j][0]:
            K.append(coverage[j][1]) #adds cov value to list
            err = 0
            break
elif COGCat == 'L':
    err = 1
    for j in range(index,len(coverage)):
        if row[0][0:name_len] == coverage[j][0]:
            L.append(coverage[j][1]) #adds cov value to list
            err = 0
            break
elif COGCat == 'M':
    err = 1
    for j in range(index,len(coverage)):
```

```
            if row[0][0:name_len] == coverage[j][0]:
                M.append(coverage[j][1]) #adds cov value to list
                err = 0
                break
    elif COGCat == 'N':
        err = 1
        for j in range(index,len(coverage)):
            if row[0][0:name_len] == coverage[j][0]:
                N.append(coverage[j][1]) #adds cov value to list
                err = 0
                break
    elif COGCat == 'O':
        err=1
        for j in range(index,len(coverage)):
            if row[0][0:name_len] == coverage[j][0]:
                O.append(coverage[j][1]) #adds cov value to list
                err = 0
                break
    elif COGCat == 'P':
        err = 1
        for j in range(index,len(coverage)):
            if row[0][0:name_len] == coverage[j][0]:
                P.append(coverage[j][1]) #adds cov value to list
                err = 0
                break
    elif COGCat == 'Q':
        err = 1
        for j in range(index,len(coverage)):
            if row[0][0:name_len] == coverage[j][0]:
                Q.append(coverage[j][1]) #adds cov value to list
                err = 0
                break
    elif COGCat == 'R':
        err = 1
        for j in range(index,len(coverage)):
            if row[0][0:name_len] == coverage[j][0]:
                R.append(coverage[j][1]) #adds cov value to list
```

```
            err = 0
            break
    elif COGCat == 'S':
        err = 1
        for j in range(index,len(coverage)):
            if row[0][0:name_len] == coverage[j][0]:
                S.append(coverage[j][1]) #adds cov value to list
                err = 0
                break
    elif COGCat == 'T':
        err = 1
        for j in range(index,len(coverage)):
            if row[0][0:name_len] == coverage[j][0]:
                T.append(coverage[j][1]) #adds cov value to list
                err = 0
                break
    elif COGCat == 'U':
        err = 1
        for j in range(index,len(coverage)):
            if row[0][0:name_len] == coverage[j][0]:
                U.append(coverage[j][1]) #adds cov value to list
                err = 0
                break
    elif COGCat == 'V':
        err = 1
        for j in range(index,len(coverage)):
            if row[0][0:name_len] == coverage[j][0]:
                V.append(coverage[j][1]) #adds cov value to list
                err = 0
                break
    elif COGCat == 'Y':
        err = 1
        for j in range(index,len(coverage)):
            if row[0][0:name_len] == coverage[j][0]:
                Y.append(coverage[j][1]) #adds cov value to list
                err = 0
                break
```

```
        elif COGCat == 'Z':
            err = 1
            for j in range(index,len(coverage)):
                if row[0][0:name_len] == coverage[j][0]:
                    Z.append(coverage[j][1]) #adds cov value to list
                    err = 0
                    break
        else:
            other.append(COGCat)
        if err == 1:
            other.append(COGCat)
print('counting done')

#### use for cov info
Ac  =  sum(A)/cov_norm
Bc = sum(B)/cov_norm
Cc = sum(C)/cov_norm
Dc = sum(D)/cov_norm
Ec = sum(E)/cov_norm
Fc = sum(F)/cov_norm
Gc = sum(G)/cov_norm
Hc = sum(H)/cov_norm
Ic = sum(I)/cov_norm
Jc = sum(J)/cov_norm
Kc = sum(K)/cov_norm
Lc = sum(L)/cov_norm
Mc = sum(M)/cov_norm
Nc = sum(N)/cov_norm
Oc = sum(O)/cov_norm
Pc = sum(P)/cov_norm
Qc = sum(Q)/cov_norm
Rc = sum(R)/cov_norm
Sc = sum(S)/cov_norm
Tc = sum(T)/cov_norm
Uc = sum(U)/cov_norm
Vc = sum(V)/cov_norm
Yc = sum(Y)/cov_norm
```

```
Zc = sum(Z)/cov_norm

#### use for counts
An = (len(A)-1)/count_norm
Bn = (len(B)-1)/count_norm
Cn = (len(C)-1)/count_norm
Dn = (len(D)-1)/count_norm
En = (len(E)-1)/count_norm
Fn = (len(F)-1)/count_norm
Gn = (len(G)-1)/count_norm
Hn = (len(H)-1)/count_norm
In = (len(I)-1)/count_norm
Jn = (len(J)-1)/count_norm
Kn = (len(K)-1)/count_norm
Ln = (len(L)-1)/count_norm
Mn = (len(M)-1)/count_norm
Nn = (len(N)-1)/count_norm
On = (len(O)-1)/count_norm
Pn = (len(P)-1)/count_norm
Qn = (len(Q)-1)/count_norm
Rn = (len(R)-1)/count_norm
Sn = (len(S)-1)/count_norm
Tn = (len(T)-1)/count_norm
Un = (len(U)-1)/count_norm
Vn = (len(V)-1)/count_norm
Yn = (len(Y)-1)/count_norm
Zn = (len(Z)-1)/count_norm

#### write data to files

results_c = [Ac, Bc, Cc, Dc, Ec, Fc, Gc, Hc, Ic, Jc, Kc, Lc, Mc, Nc, Oc, Pc, Qc, Rc, Sc, Tc, Uc,
Vc, Yc, Zc]
results_n = [An, Bn, Cn, Dn, En, Fn, Gn, Hn, In, Jn, Kn, Ln, Mn, Nn, On, Pn, Qn, Rn, Sn, Tn,
Un, Vn, Yn, Zn]
header = ['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V',
'Y', 'Z', 'total', 'issues']
```

```
with open('%s' + file_root + '.assembled.faa.COG_summary.csv', 'w') as out_file:
    out_csv = csv.writer(out_file)
    out_csv.writerow(['COG category', 'by coverage', 'by count'])
    for i in range(len(header)-2):
        out_csv.writerow([header[i], results_c[i], results_n[i]])
    out_csv.writerow([header[-2], cov_norm, count_norm])
    out_csv.writerow([header[-1], len(other), len(other)])
print('COG done')
"""%(ass_num, head_dir + ass_num + '.assembled_cov.faa', COG_file, head_dir)

with open(res_dir + 'COG_summarize_v2.py', 'w') as COG_script:
    COG_script.write(COG_code)

#### write KEGG script
KEGG_code = """# -*- coding: utf-8 -*-
'''Created on Mon Jul 25 13:48:37 2016
@author: jay3
This script will find ORFs with given KO# and aggregate them to pathways. It is based on
Reads_cov_v3.py
v2:
Nothing really new, just to separate the working version from others. Added 0s to initiate Ko
vars to avoid issues with math calculations resulting from empty lists
added print statements.
v3:
Updated with new KEGG numbers. Added method to use counts instead of coverage
v4:
Need data to normalize. For counts, number of ORFS, and for cov, sum(cov*ORFS), use
linemag.faa to get this info.
v5:
Some changes made in v4 as debugging. The 5 designation is to set it apart.
v6:
Incorporated SulfurSub_on_real_data_v3.py.
v7:
Added capability to save both count and coverage versions. Added file write outs for results.
Added KOs from Yau13. Fully tested (161025).
v8:
Removed extra lines to reduce size. Added write out of all markers.
```

```
'''
import csv
import Bio.SeqIO as SeqIO
import numpy as np
from Bio import pairwise2
file_root = '%s'
assembly_file = '%s'
KEGG_file = '%s'
print('start KEGG')

#### read ORF coverages from linemag.faa
coverage = []
cov_norm = 0
with open(assembly_file, 'r') as orf_file:
    for read_record in SeqIO.parse(orf_file, "fasta"):
        string = read_record.id
        string2 = string.split('|')
        coverage.append([string2[0], float(string2[1][10:])])
        cov_norm = cov_norm + float(string2[1][10:])
count_norm = len(coverage)
name_len = len(string2[0])

#### read KO ORFS and append cov to KO_vars
K00362 = [0]
K00363 = [0]
K03385 = [0]
K15876 = [0]
K17877 = [0]
K00366 = [0]
K02305 = [0]
K04561 = [0]
K00376 = [0]
K00531 = [0]
K02586 = [0]
K02591 = [0]
K02588 = [0]
K10535 = [0]
```

K10944 = [0]
K01601 = [0]
K01602 = [0]
K00855 = [0]
K15230 = [0]
K15231 = [0]
K15234 = [0]
K15233 = [0]
K15232 = [0]
K00192 = [0]
K00198 = [0]
K03518 = [0]
K03519 = [0]
K03520 = [0]
K14138 = [0]
K02256 = [0]
K02262 = [0]
K02274 = [0]
K02276 = [0]
K00401 = [0]
K00400 = [0]
K16157 = [0]
K16158 = [0]
K16159 = [0]
K16161 = [0]
K00390 = [0]
K00392 = [0]
K00380 = [0]
K00381 = [0]
K00394 = [0]
K00394r = []
K00394o = []
K00395 = [0]
K00395r = []
K00395o = []
K11180 = [0]
K11180r = []

K11180o = []
K11181 = [0]
K11181r = []
K11181o = []
K17224 = [0]
K17227 = [0]
K17226 = [0]
K17222 = [0]
K17223 = [0]
K17225 = [0]
K05973 = [0]
K03821 = [0]
K15342 = [0]
K09951 = [0]
K07012 = [0]
K07475 = [0]
K19088 = [0]
K19123 = [0]
K19127 = [0]
K07016 = [0]
K19138 = [0]
K19141 = [0]
K09952 = [0]
K19137 = [0]
K07464 = [0]
K02703 = [0]
K02706 = [0]
K02705 = [0]
K02704 = [0]
K02707 = [0]
K02708 = [0]
K02689 = [0]
K02690 = [0]
K02691 = [0]
K02692 = [0]
K02693 = [0]
K02694 = [0]

K08928 = [0]
K08929 = [0]
K08940 = [0]
K08941 = [0]
K08942 = [0]
K08943 = [0]
K04643 = [0]
K04642 = [0]
K04641 = [0]
K04250 = [0]
K00909 = [0]
K01428 = [0]
K01429 = [0]
K01430 = [0]
K00111 = [0]
K00112 = [0]
K00113 = [0]
K00096 = [0]
K00518 = [0]
K04564 = [0]
K04565 = [0]
K16627 = [0]
K06164 = [0]
K05780 = [0]
K06165 = [0]
K06166 = [0]
K06167 = [0]
K06163 = [0]
K06162 = [0]
K08977 = [0]
K09836 = [0]
K15746 = [0]
K16953 = [0]
K17486 = [0]
K20452 = [0]
K07306 = [0]
K17218 = [0]

```
K03553 = [0]
K00394_ORFname = []
K00395_ORFname = []
K11180_ORFname = []
K11181_ORFname = []
no_cov = []
index = 0
with open(KEGG_file, 'r', newline = '') as read_file:
    KOs_csv = csv.reader(read_file, delimiter = '\t')
    KOs_all = []
    for row in KOs_csv:
        err = []
        KO_ID = row[2][3:]
        if KO_ID == 'K00362':
            err = 1
            for j in range(index,len(coverage)):
                if row[0][0:name_len] == coverage[j][0]:
                    K00362.append(coverage[j][1]) #adds cov value to list
                    err = 0
                    break
        elif KO_ID == 'K00363':
            err = 1
            for j in range(index,len(coverage)):
                if row[0][0:name_len] == coverage[j][0]:
                    K00363.append(coverage[j][1]) #adds cov value to list
                    err = 0
                    break
        elif KO_ID == 'K03385':
            err = 1
            for j in range(index,len(coverage)):
                if row[0][0:name_len] == coverage[j][0]:
                    K03385.append(coverage[j][1]) #adds cov value to list
                    err = 0
                    break
        elif KO_ID == 'K15876':
            err = 1
            for j in range(index,len(coverage)):
```

```
        if row[0][0:name_len] == coverage[j][0]:
            K15876.append(coverage[j][1]) #adds cov value to list
            err = 0
            break
elif KO_ID == 'K17877':
    err = 1
    for j in range(index,len(coverage)):
        if row[0][0:name_len] == coverage[j][0]:
            K17877.append(coverage[j][1]) #adds cov value to list
            err = 0
            break
elif KO_ID == 'K00366':
    err = 1
    for j in range(index,len(coverage)):
        if row[0][0:name_len] == coverage[j][0]:
            K00366.append(coverage[j][1]) #adds cov value to list
            err = 0
            break
elif KO_ID == 'K02305':
    err = 1
    for j in range(index,len(coverage)):
        if row[0][0:name_len] == coverage[j][0]:
            K02305.append(coverage[j][1]) #adds cov value to list
            err = 0
            break
elif KO_ID == 'K04561':
    err = 1
    for j in range(index,len(coverage)):
        if row[0][0:name_len] == coverage[j][0]:
            K04561.append(coverage[j][1]) #adds cov value to list
            err = 0
            break
elif KO_ID == 'K00376':
    err = 1
    for j in range(index,len(coverage)):
        if row[0][0:name_len] == coverage[j][0]:
            K00376.append(coverage[j][1]) #adds cov value to list
```

```
            err = 0
            break
    elif KO_ID == 'K00531':
        err = 1
        for j in range(index,len(coverage)):
            if row[0][0:name_len] == coverage[j][0]:
                K00531.append(coverage[j][1]) #adds cov value to list
                err = 0
                break
    elif KO_ID == 'K02586':
        err = 1
        for j in range(index,len(coverage)):
            if row[0][0:name_len] == coverage[j][0]:
                K02586.append(coverage[j][1]) #adds cov value to list
                err = 0
                break
    elif KO_ID == 'K02591':
        err = 1
        for j in range(index,len(coverage)):
            if row[0][0:name_len] == coverage[j][0]:
                K02591.append(coverage[j][1]) #adds cov value to list
                err = 0
                break
    elif KO_ID == 'K02588':
        err = 1
        for j in range(index,len(coverage)):
            if row[0][0:name_len] == coverage[j][0]:
                K02588.append(coverage[j][1]) #adds cov value to list
                err = 0
                break
    elif KO_ID == 'K10535':
        err = 1
        for j in range(index,len(coverage)):
            if row[0][0:name_len] == coverage[j][0]:
                K10535.append(coverage[j][1]) #adds cov value to list
                err = 0
                break
```

```
elif KO_ID == 'K10944':
    err = 1
    for j in range(index,len(coverage)):
        if row[0][0:name_len] == coverage[j][0]:
            K10944.append(coverage[j][1]) #adds cov value to list
            err = 0
            break
elif KO_ID == 'K01601':
    err = 1
    for j in range(index,len(coverage)):
        if row[0][0:name_len] == coverage[j][0]:
            K01601.append(coverage[j][1]) #adds cov value to list
            err = 0
            break
elif KO_ID == 'K01602':
    err = 1
    for j in range(index,len(coverage)):
        if row[0][0:name_len] == coverage[j][0]:
            K01602.append(coverage[j][1]) #adds cov value to list
            err = 0
            break
elif KO_ID == 'K00855':
    err = 1
    for j in range(index,len(coverage)):
        if row[0][0:name_len] == coverage[j][0]:
            K00855.append(coverage[j][1]) #adds cov value to list
            err = 0
            break
elif KO_ID == 'K15230':
    err = 1
    for j in range(index,len(coverage)):
        if row[0][0:name_len] == coverage[j][0]:
            K15230.append(coverage[j][1]) #adds cov value to list
            err = 0
            break
elif KO_ID == 'K15231':
    err = 1
```

```
    for j in range(index,len(coverage)):
        if row[0][0:name_len] == coverage[j][0]:
            K15231.append(coverage[j][1]) #adds cov value to list
            err = 0
            break
elif KO_ID == 'K15234':
    err = 1
    for j in range(index,len(coverage)):
        if row[0][0:name_len] == coverage[j][0]:
            K15234.append(coverage[j][1]) #adds cov value to list
            err = 0
            break
elif KO_ID == 'K15233':
    err = 1
    for j in range(index,len(coverage)):
        if row[0][0:name_len] == coverage[j][0]:
            K15233.append(coverage[j][1]) #adds cov value to list
            err = 0
            break
elif KO_ID == 'K15232':
    err = 1
    for j in range(index,len(coverage)):
        if row[0][0:name_len] == coverage[j][0]:
            K15232.append(coverage[j][1]) #adds cov value to list
            err = 0
            break
elif KO_ID == 'K00192':
    err = 1
    for j in range(index,len(coverage)):
        if row[0][0:name_len] == coverage[j][0]:
            K00192.append(coverage[j][1]) #adds cov value to list
            err = 0
            break
elif KO_ID == 'K00198':
    err = 1
    for j in range(index,len(coverage)):
        if row[0][0:name_len] == coverage[j][0]:
```

```
            K00198.append(coverage[j][1]) #adds cov value to list
            err = 0
            break
    elif KO_ID == 'K03518':
        err = 1
        for j in range(index,len(coverage)):
            if row[0][0:name_len] == coverage[j][0]:
                K03518.append(coverage[j][1]) #adds cov value to list
                err = 0
                break
    elif KO_ID == 'K03519':
        err = 1
        for j in range(index,len(coverage)):
            if row[0][0:name_len] == coverage[j][0]:
                K03519.append(coverage[j][1]) #adds cov value to list
                err = 0
                break
    elif KO_ID == 'K03520':
        err = 1
        for j in range(index,len(coverage)):
            if row[0][0:name_len] == coverage[j][0]:
                K03520.append(coverage[j][1]) #adds cov value to list
                err = 0
                break
    elif KO_ID == 'K14138':
        err = 1
        for j in range(index,len(coverage)):
            if row[0][0:name_len] == coverage[j][0]:
                K14138.append(coverage[j][1]) #adds cov value to list
                err = 0
                break
    elif KO_ID == 'K02256':
        err = 1
        for j in range(index,len(coverage)):
            if row[0][0:name_len] == coverage[j][0]:
                K02256.append(coverage[j][1]) #adds cov value to list
                err = 0
```

```
            break
    elif KO_ID == 'K02262':
        err = 1
        for j in range(index,len(coverage)):
            if row[0][0:name_len] == coverage[j][0]:
                K02262.append(coverage[j][1]) #adds cov value to list
                err = 0
                break
    elif KO_ID == 'K02274':
        err = 1
        for j in range(index,len(coverage)):
            if row[0][0:name_len] == coverage[j][0]:
                K02274.append(coverage[j][1]) #adds cov value to list
                err = 0
                break
    elif KO_ID == 'K02276':
        err = 1
        for j in range(index,len(coverage)):
            if row[0][0:name_len] == coverage[j][0]:
                K02276.append(coverage[j][1]) #adds cov value to list
                err = 0
                break
    elif KO_ID == 'K00401':
        err = 1
        for j in range(index,len(coverage)):
            if row[0][0:name_len] == coverage[j][0]:
                K00401.append(coverage[j][1]) #adds cov value to list
                err = 0
                break
    elif KO_ID == 'K00400':
        err = 1
        for j in range(index,len(coverage)):
            if row[0][0:name_len] == coverage[j][0]:
                K00400.append(coverage[j][1]) #adds cov value to list
                err = 0
                break
    elif KO_ID == 'K16157':
```

```
        err = 1
        for j in range(index,len(coverage)):
            if row[0][0:name_len] == coverage[j][0]:
                K16157.append(coverage[j][1]) #adds cov value to list
                err = 0
                break
    elif KO_ID == 'K16158':
        err = 1
        for j in range(index,len(coverage)):
            if row[0][0:name_len] == coverage[j][0]:
                K16158.append(coverage[j][1]) #adds cov value to list
                err = 0
                break
    elif KO_ID == 'K16159':
        err = 1
        for j in range(index,len(coverage)):
            if row[0][0:name_len] == coverage[j][0]:
                K16159.append(coverage[j][1]) #adds cov value to list
                err = 0
                break
    elif KO_ID == 'K16161':
        err = 1
        for j in range(index,len(coverage)):
            if row[0][0:name_len] == coverage[j][0]:
                K16161.append(coverage[j][1]) #adds cov value to list
                err = 0
                break
    elif KO_ID == 'K00390':
        err = 1
        for j in range(index,len(coverage)):
            if row[0][0:name_len] == coverage[j][0]:
                K00390.append(coverage[j][1]) #adds cov value to list
                err = 0
                break
    elif KO_ID == 'K00392':
        err = 1
        for j in range(index,len(coverage)):
```

```
            if row[0][0:name_len] == coverage[j][0]:
                K00392.append(coverage[j][1]) #adds cov value to list
                err = 0
                break
        elif KO_ID == 'K00380':
            err = 1
            for j in range(index,len(coverage)):
                if row[0][0:name_len] == coverage[j][0]:
                    K00380.append(coverage[j][1]) #adds cov value to list
                    err = 0
                    break
        elif KO_ID == 'K00381':
            err = 1
            for j in range(index,len(coverage)):
                if row[0][0:name_len] == coverage[j][0]:
                    K00381.append(coverage[j][1]) #adds cov value to list
                    err = 0
                    break

#### sulfur assimilatory and dissimilatory
        elif KO_ID == 'K00394':
            err = 1
            for j in range(index,len(coverage)):
                if row[0][0:name_len] == coverage[j][0]:
                    K00394_ORFname.append(row[0][0:name_len])
                    K00394.append(coverage[j][1]) #adds cov value to list
                    err = 0
                    break
        elif KO_ID == 'K00395':
            err = 1
            for j in range(index,len(coverage)):
                if row[0][0:name_len] == coverage[j][0]:
                    K00395_ORFname.append(row[0][0:name_len])
                    K00395.append(coverage[j][1]) #adds cov value to list
                    err = 0
                    break
        elif KO_ID == 'K11180':
```

```
            err = 1
            for j in range(index,len(coverage)):
                if row[0][0:name_len] == coverage[j][0]:
                    K11180_ORFname.append(row[0][0:name_len])
                    K11180.append(coverage[j][1]) #adds cov value to list
                    err = 0
                    break
        elif KO_ID == 'K11181':
            err = 1
            for j in range(index,len(coverage)):
                if row[0][0:name_len] == coverage[j][0]:
                    K11181_ORFname.append(row[0][0:name_len])
                    K11181.append(coverage[j][1]) #adds cov value to list
                    err = 0
                    break

#### others
        elif KO_ID == 'K17224':
            err = 1
            for j in range(index,len(coverage)):
                if row[0][0:name_len] == coverage[j][0]:
                    K17224.append(coverage[j][1]) #adds cov value to list
                    err = 0
                    break
        elif KO_ID == 'K17227':
            err = 1
            for j in range(index,len(coverage)):
                if row[0][0:name_len] == coverage[j][0]:
                    K17227.append(coverage[j][1]) #adds cov value to list
                    err = 0
                    break
        elif KO_ID == 'K17226':
            err = 1
            for j in range(index,len(coverage)):
                if row[0][0:name_len] == coverage[j][0]:
                    K17226.append(coverage[j][1]) #adds cov value to list
                    err = 0
```

```
            break
    elif KO_ID == 'K17222':
        err = 1
        for j in range(index,len(coverage)):
            if row[0][0:name_len] == coverage[j][0]:
                K17222.append(coverage[j][1]) #adds cov value to list
                err = 0
                break
    elif KO_ID == 'K17223':
        err = 1
        for j in range(index,len(coverage)):
            if row[0][0:name_len] == coverage[j][0]:
                K17223.append(coverage[j][1]) #adds cov value to list
                err = 0
                break
    elif KO_ID == 'K17225':
        err = 1
        for j in range(index,len(coverage)):
            if row[0][0:name_len] == coverage[j][0]:
                K17225.append(coverage[j][1]) #adds cov value to list
                err = 0
                break
    elif KO_ID == 'K05973':
        err = 1
        for j in range(index,len(coverage)):
            if row[0][0:name_len] == coverage[j][0]:
                K05973.append(coverage[j][1]) #adds cov value to list
                err = 0
                break
    elif KO_ID == 'K03821':
        err = 1
        for j in range(index,len(coverage)):
            if row[0][0:name_len] == coverage[j][0]:
                K03821.append(coverage[j][1]) #adds cov value to list
                err = 0
                break
    elif KO_ID == 'K15342':
```

```
    err = 1
    for j in range(index,len(coverage)):
        if row[0][0:name_len] == coverage[j][0]:
            K15342.append(coverage[j][1]) #adds cov value to list
            err = 0
            break
elif KO_ID == 'K09951':
    err = 1
    for j in range(index,len(coverage)):
        if row[0][0:name_len] == coverage[j][0]:
            K09951.append(coverage[j][1]) #adds cov value to list
            err = 0
            break
elif KO_ID == 'K07012':
    err = 1
    for j in range(index,len(coverage)):
        if row[0][0:name_len] == coverage[j][0]:
            K07012.append(coverage[j][1]) #adds cov value to list
            err = 0
            break
elif KO_ID == 'K07475':
    err = 1
    for j in range(index,len(coverage)):
        if row[0][0:name_len] == coverage[j][0]:
            K07475.append(coverage[j][1]) #adds cov value to list
            err = 0
            break
elif KO_ID == 'K19088':
    err = 1
    for j in range(index,len(coverage)):
        if row[0][0:name_len] == coverage[j][0]:
            K19088.append(coverage[j][1]) #adds cov value to list
            err = 0
            break
elif KO_ID == 'K19123':
    err = 1
    for j in range(index,len(coverage)):
```

```
        if row[0][0:name_len] == coverage[j][0]:
            K19123.append(coverage[j][1]) #adds cov value to list
            err = 0
            break
elif KO_ID == 'K19127':
    err = 1
    for j in range(index,len(coverage)):
        if row[0][0:name_len] == coverage[j][0]:
            K19127.append(coverage[j][1]) #adds cov value to list
            err = 0
            break
elif KO_ID == 'K07016':
    err = 1
    for j in range(index,len(coverage)):
        if row[0][0:name_len] == coverage[j][0]:
            K07016.append(coverage[j][1]) #adds cov value to list
            err = 0
            break
elif KO_ID == 'K19138':
    err = 1
    for j in range(index,len(coverage)):
        if row[0][0:name_len] == coverage[j][0]:
            K19138.append(coverage[j][1]) #adds cov value to list
            err = 0
            break
elif KO_ID == 'K19141':
    err = 1
    for j in range(index,len(coverage)):
        if row[0][0:name_len] == coverage[j][0]:
            K19141.append(coverage[j][1]) #adds cov value to list
            err = 0
            break
elif KO_ID == 'K09952':
    err = 1
    for j in range(index,len(coverage)):
        if row[0][0:name_len] == coverage[j][0]:
            K09952.append(coverage[j][1]) #adds cov value to list
```

```
            err = 0
            break
    elif KO_ID == 'K19137':
        err = 1
        for j in range(index,len(coverage)):
            if row[0][0:name_len] == coverage[j][0]:
                K19137.append(coverage[j][1]) #adds cov value to list
                err = 0
                break
    elif KO_ID == 'K07464':
        err = 1
        for j in range(index,len(coverage)):
            if row[0][0:name_len] == coverage[j][0]:
                K07464.append(coverage[j][1]) #adds cov value to list
                err = 0
                break
    elif KO_ID == 'K02703':
        err = 1
        for j in range(index,len(coverage)):
            if row[0][0:name_len] == coverage[j][0]:
                K02703.append(coverage[j][1]) #adds cov value to list
                err = 0
                break
    elif KO_ID == 'K02706':
        err = 1
        for j in range(index,len(coverage)):
            if row[0][0:name_len] == coverage[j][0]:
                K02706.append(coverage[j][1]) #adds cov value to list
                err = 0
                break
    elif KO_ID == 'K02705':
        err = 1
        for j in range(index,len(coverage)):
            if row[0][0:name_len] == coverage[j][0]:
                K02705.append(coverage[j][1]) #adds cov value to list
                err = 0
                break
```

```
elif KO_ID == 'K02704':
    err = 1
    for j in range(index,len(coverage)):
        if row[0][0:name_len] == coverage[j][0]:
            K02704.append(coverage[j][1]) #adds cov value to list
            err = 0
            break
elif KO_ID == 'K02707':
    err = 1
    for j in range(index,len(coverage)):
        if row[0][0:name_len] == coverage[j][0]:
            K02707.append(coverage[j][1]) #adds cov value to list
            err = 0
            break
elif KO_ID == 'K02708':
    err = 1
    for j in range(index,len(coverage)):
        if row[0][0:name_len] == coverage[j][0]:
            K02708.append(coverage[j][1]) #adds cov value to list
            err = 0
            break
elif KO_ID == 'K02689':
    err = 1
    for j in range(index,len(coverage)):
        if row[0][0:name_len] == coverage[j][0]:
            K02689.append(coverage[j][1]) #adds cov value to list
            err = 0
            break
elif KO_ID == 'K02690':
    err = 1
    for j in range(index,len(coverage)):
        if row[0][0:name_len] == coverage[j][0]:
            K02690.append(coverage[j][1]) #adds cov value to list
            err = 0
            break
elif KO_ID == 'K02691':
    err = 1
```

```
    for j in range(index,len(coverage)):
        if row[0][0:name_len] == coverage[j][0]:
            K02691.append(coverage[j][1]) #adds cov value to list
            err = 0
            break
elif KO_ID == 'K02692':
    err = 1
    for j in range(index,len(coverage)):
        if row[0][0:name_len] == coverage[j][0]:
            K02692.append(coverage[j][1]) #adds cov value to list
            err = 0
            break
elif KO_ID == 'K02693':
    err = 1
    for j in range(index,len(coverage)):
        if row[0][0:name_len] == coverage[j][0]:
            K02693.append(coverage[j][1]) #adds cov value to list
            err = 0
            break
elif KO_ID == 'K02694':
    err = 1
    for j in range(index,len(coverage)):
        if row[0][0:name_len] == coverage[j][0]:
            K02694.append(coverage[j][1]) #adds cov value to list
            err = 0
            break
elif KO_ID == 'K08928':
    err = 1
    for j in range(index,len(coverage)):
        if row[0][0:name_len] == coverage[j][0]:
            K08928.append(coverage[j][1]) #adds cov value to list
            err = 0
            break
elif KO_ID == 'K08929':
    err = 1
    for j in range(index,len(coverage)):
        if row[0][0:name_len] == coverage[j][0]:
```

```
            K08929.append(coverage[j][1]) #adds cov value to list
            err = 0
            break
    elif KO_ID == 'K08940':
        err = 1
        for j in range(index,len(coverage)):
            if row[0][0:name_len] == coverage[j][0]:
                K08940.append(coverage[j][1]) #adds cov value to list
                err = 0
                break
    elif KO_ID == 'K08941':
        err = 1
        for j in range(index,len(coverage)):
            if row[0][0:name_len] == coverage[j][0]:
                K08941.append(coverage[j][1]) #adds cov value to list
                err = 0
                break
    elif KO_ID == 'K08942':
        err = 1
        for j in range(index,len(coverage)):
            if row[0][0:name_len] == coverage[j][0]:
                K08942.append(coverage[j][1]) #adds cov value to list
                err = 0
                break
    elif KO_ID == 'K08943':
        err = 1
        for j in range(index,len(coverage)):
            if row[0][0:name_len] == coverage[j][0]:
                K08943.append(coverage[j][1]) #adds cov value to list
                err = 0
                break
    elif KO_ID == 'K04643':
        err = 1
        for j in range(index,len(coverage)):
            if row[0][0:name_len] == coverage[j][0]:
                K04643.append(coverage[j][1]) #adds cov value to list
                err = 0
```

```
            break
    elif KO_ID == 'K04642':
        err = 1
        for j in range(index,len(coverage)):
            if row[0][0:name_len] == coverage[j][0]:
                K04642.append(coverage[j][1]) #adds cov value to list
                err = 0
                break
    elif KO_ID == 'K04641':
        err = 1
        for j in range(index,len(coverage)):
            if row[0][0:name_len] == coverage[j][0]:
                K04641.append(coverage[j][1]) #adds cov value to list
                err = 0
                break
    elif KO_ID == 'K04250':
        err = 1
        for j in range(index,len(coverage)):
            if row[0][0:name_len] == coverage[j][0]:
                K04250.append(coverage[j][1]) #adds cov value to list
                err = 0
                break
    elif KO_ID == 'K00909':
        err = 1
        for j in range(index,len(coverage)):
            if row[0][0:name_len] == coverage[j][0]:
                K00909.append(coverage[j][1]) #adds cov value to list
                err = 0
                break
    elif KO_ID == 'K01428':
        err = 1
        for j in range(index,len(coverage)):
            if row[0][0:name_len] == coverage[j][0]:
                K01428.append(coverage[j][1]) #adds cov value to list
                err = 0
                break
    elif KO_ID == 'K01429':
```

```
    err = 1
    for j in range(index,len(coverage)):
        if row[0][0:name_len] == coverage[j][0]:
            K01429.append(coverage[j][1]) #adds cov value to list
            err = 0
            break
elif KO_ID == 'K01430':
    err = 1
    for j in range(index,len(coverage)):
        if row[0][0:name_len] == coverage[j][0]:
            K01430.append(coverage[j][1]) #adds cov value to list
            err = 0
            break
elif KO_ID == 'K00111':
    err = 1
    for j in range(index,len(coverage)):
        if row[0][0:name_len] == coverage[j][0]:
            K00111.append(coverage[j][1]) #adds cov value to list
            err = 0
            break
elif KO_ID == 'K00112':
    err = 1
    for j in range(index,len(coverage)):
        if row[0][0:name_len] == coverage[j][0]:
            K00112.append(coverage[j][1]) #adds cov value to list
            err = 0
            break
elif KO_ID == 'K00113':
    err = 1
    for j in range(index,len(coverage)):
        if row[0][0:name_len] == coverage[j][0]:
            K00113.append(coverage[j][1]) #adds cov value to list
            err = 0
            break
elif KO_ID == 'K00096':
    err = 1
    for j in range(index,len(coverage)):
```

```
            if row[0][0:name_len] == coverage[j][0]:
                K00096.append(coverage[j][1]) #adds cov value to list
                err = 0
                break
    elif KO_ID == 'K00518':
        err = 1
        for j in range(index,len(coverage)):
            if row[0][0:name_len] == coverage[j][0]:
                K00518.append(coverage[j][1]) #adds cov value to list
                err = 0
                break
    elif KO_ID == 'K04564':
        err = 1
        for j in range(index,len(coverage)):
            if row[0][0:name_len] == coverage[j][0]:
                K04564.append(coverage[j][1]) #adds cov value to list
                err = 0
                break
    elif KO_ID == 'K04565':
        err = 1
        for j in range(index,len(coverage)):
            if row[0][0:name_len] == coverage[j][0]:
                K04565.append(coverage[j][1]) #adds cov value to list
                err = 0
                break
    elif KO_ID == 'K16627':
        err = 1
        for j in range(index,len(coverage)):
            if row[0][0:name_len] == coverage[j][0]:
                K16627.append(coverage[j][1]) #adds cov value to list
                err = 0
                break
    elif KO_ID == 'K06164':
        err = 1
        for j in range(index,len(coverage)):
            if row[0][0:name_len] == coverage[j][0]:
                K06164.append(coverage[j][1]) #adds cov value to list
```

```
            err = 0
            break
    elif KO_ID == 'K05780':
        err = 1
        for j in range(index,len(coverage)):
            if row[0][0:name_len] == coverage[j][0]:
                K05780.append(coverage[j][1]) #adds cov value to list
                err = 0
                break
    elif KO_ID == 'K06165':
        err = 1
        for j in range(index,len(coverage)):
            if row[0][0:name_len] == coverage[j][0]:
                K06165.append(coverage[j][1]) #adds cov value to list
                err = 0
                break
    elif KO_ID == 'K06166':
        err = 1
        for j in range(index,len(coverage)):
            if row[0][0:name_len] == coverage[j][0]:
                K06166.append(coverage[j][1]) #adds cov value to list
                err = 0
                break
    elif KO_ID == 'K06167':
        err = 1
        for j in range(index,len(coverage)):
            if row[0][0:name_len] == coverage[j][0]:
                K06167.append(coverage[j][1]) #adds cov value to list
                err = 0
                break
    elif KO_ID == 'K06163':
        err = 1
        for j in range(index,len(coverage)):
            if row[0][0:name_len] == coverage[j][0]:
                K06163.append(coverage[j][1]) #adds cov value to list
                err = 0
                break
```

```
elif KO_ID == 'K06162':
    err = 1
    for j in range(index,len(coverage)):
        if row[0][0:name_len] == coverage[j][0]:
            K06162.append(coverage[j][1]) #adds cov value to list
            err = 0
            break
elif KO_ID == 'K08977':
    err = 1
    for j in range(index,len(coverage)):
        if row[0][0:name_len] == coverage[j][0]:
            K08977.append(coverage[j][1]) #adds cov value to list
            err = 0
            break
elif KO_ID == 'K09836':
    err = 1
    for j in range(index,len(coverage)):
        if row[0][0:name_len] == coverage[j][0]:
            K09836.append(coverage[j][1]) #adds cov value to list
            err = 0
            break
elif KO_ID == 'K15746':
    err = 1
    for j in range(index,len(coverage)):
        if row[0][0:name_len] == coverage[j][0]:
            K15746.append(coverage[j][1]) #adds cov value to list
            err = 0
            break
elif KO_ID == 'K16953':
    err = 1
    for j in range(index,len(coverage)):
        if row[0][0:name_len] == coverage[j][0]:
            K16953.append(coverage[j][1]) #adds cov value to list
            err = 0
            break
elif KO_ID == 'K17486':
    err = 1
```

```
        for j in range(index,len(coverage)):
            if row[0][0:name_len] == coverage[j][0]:
                K17486.append(coverage[j][1]) #adds cov value to list
                err = 0
                break
    elif KO_ID == 'K20452':
        err = 1
        for j in range(index,len(coverage)):
            if row[0][0:name_len] == coverage[j][0]:
                K20452.append(coverage[j][1]) #adds cov value to list
                err = 0
                break
    elif KO_ID == 'K07306':
        err = 1
        for j in range(index,len(coverage)):
            if row[0][0:name_len] == coverage[j][0]:
                K07306.append(coverage[j][1]) #adds cov value to list
                err = 0
                break
    elif KO_ID == 'K17218':
        err = 1
        for j in range(index,len(coverage)):
            if row[0][0:name_len] == coverage[j][0]:
                K17218.append(coverage[j][1]) #adds cov value to list
                err = 0
                break
    elif KO_ID == 'K03553':
        err = 1
        for j in range(index,len(coverage)):
            if row[0][0:name_len] == coverage[j][0]:
                K03553.append(coverage[j][1]) #adds cov value to list
                err = 0
                break
    if err == 1:
        no_cov.append(KO_ID)
print('counting done')
```

```
#### SulfurSub for assimilatory vs dissimilatory
limit=14
#### split K00394
#### get seqs for the marker
K00394_ORFseq = []
with open(assembly_file, 'r') as orf_file:
    for read_record in SeqIO.parse(orf_file, "fasta"):
        string = read_record.id
        string2 = string.split('|')
        for ORF in K00394_ORFname:
            #print(ORF)
            #print(string2[0])
            if ORF == string2[0]:
                K00394_ORFseq.append(read_record)
marker = K00394_ORFseq ###ORFS
#### get the database seqs
db = []
with open('/srv/scratch/jgi/Cavlab_pipeline_resources/v1/K00394_pathway_database_v1.fasta',
'r') as dsrAB_file:
    db = list(SeqIO.parse(dsrAB_file, "fasta"))
#### make a prediction
for ORF in marker:
    scorei = []
    for seq in db:
        align = pairwise2.align.localms(ORF.seq, seq.seq, 2, -1, -.5, -.1, score_only = 1)
        scorei.append(align)
        seq.description = align
    keep = max(scorei)
    cat = []
    for seq in db:
        if seq.description >= keep:
            cat.append(seq.id.split('$')[0])
    dis = 0
    ox = 0
    oth = 0
    un = 0
    for obs in cat:
```

```
        if obs == 'Reductive':
            dis = dis + 1
        elif obs == 'Oxidative':
            ox = ox + 1
        elif obs == 'Other':
            oth = oth + 1
        else:
            un = un + 1
    if keep > limit:
        #assign
        if dis > ox and dis > oth:
            assignment = 'Reductive'
            K00394r.append(float(ORF.id.split('|')[1][10:]))
        elif ox > dis and ox > oth:
            assignment = 'Oxidative'
            K00394o.append(float(ORF.id.split('|')[1][10:]))
        elif oth > dis and oth > ox:
            assignment = 'Other'
        else:
            assignment = 'Unknown'
print('K00394 done')

#### split K00395
#### get seqs for the marker
K00395_ORFseq = []
with open(assembly_file,'r') as orf_file:
    for read_record in SeqIO.parse(orf_file, "fasta"):
        string = read_record.id
        string2 = string.split('|')
        for ORF in K00395_ORFname:
            #print(ORF)
            #print(string2[0])
            if ORF == string2[0]:
                K00395_ORFseq.append(read_record)
marker = K00395_ORFseq ###ORFS
#### get the database seqs
db = []
```

```
with
open('/srv/scratch/jgi/Cavlab_pipeline_resources/v1/K00395_pathway_database_v1.fasta','r') as
dsrAB_file:
    db = list(SeqIO.parse(dsrAB_file, "fasta"))
#### make a prediction
for ORF in marker:
    scorei = []
    for seq in db:
        align = pairwise2.align.localms(ORF.seq, seq.seq, 2, -1, -.5, -.1, score_only = 1)
        scorei.append(align)
        seq.description = align
    keep = max(scorei)
    cat = []
    for seq in db:
        if seq.description >= keep:
            cat.append(seq.id.split('$')[0])
    dis = 0
    ox = 0
    oth = 0
    un = 0
    for obs in cat:
        if obs == 'Reductive':
            dis = dis + 1
        elif obs == 'Oxidative':
            ox = ox + 1
        elif obs == 'Other':
            oth = oth + 1
        else:
            un = un + 1
    if keep > limit:
        #assign
        if dis > ox and dis > oth:
            assignment = 'Reductive'
            K00395r.append(float(ORF.id.split('|')[1][10:]))
        elif ox > dis and ox > oth:
            assignment = 'Oxidative'
            K00395o.append(float(ORF.id.split('|')[1][10:]))
```

```
        elif oth > dis and oth > ox:
            assignment = 'Other'
        else:
            assignment = 'Unknown'
print('K00395 done')

#### split K11180
#### get seqs for the marker
K11180_ORFseq = []
with open(assembly_file, 'r') as orf_file:
    for read_record in SeqIO.parse(orf_file, "fasta"):
        string = read_record.id
        string2 = string.split('|')
        for ORF in K11180_ORFname:
            #print(ORF)
            #print(string2[0])
            if ORF == string2[0]:
                K11180_ORFseq.append(read_record)
marker = K11180_ORFseq ###ORFS
#### get the database seqs
db = []
with open('/srv/scratch/jgi/Cavlab_pipeline_resources/v1/K11180_pathway_database_v1.fasta',
'r') as dsrAB_file:
    db = list(SeqIO.parse(dsrAB_file, "fasta"))
#### make a prediction
for ORF in marker:
    scorei = []
    for seq in db:
        align = pairwise2.align.localms(ORF.seq, seq.seq, 2, -1, -.5, -.1, score_only = 1)
        scorei.append(align)
        seq.description = align
    keep = max(scorei)
    cat = []
    for seq in db:
        if seq.description >= keep:
            cat.append(seq.id.split('$')[0])
    dis = 0
```

```
    ox = 0
    oth = 0
    un = 0
    for obs in cat:
        if obs == 'Reductive':
            dis = dis + 1
        elif obs == 'Oxidative':
            ox = ox + 1
        elif obs == 'Other':
            oth = oth + 1
        else:
            un = un + 1
    if keep > limit:
        #assign
        if dis > ox and dis > oth:
            assignment = 'Reductive'
            K11180r.append(float(ORF.id.split('|')[1][10:]))
        elif ox > dis and ox > oth:
            assignment = 'Oxidative'
            K11180o.append(float(ORF.id.split('|')[1][10:]))
        elif oth > dis and oth > ox:
            assignment = 'Other'
        else:
            assignment = 'Unknown'
print('K11180 done')

#### split K11181
#### get seqs for the marker
K11181_ORFseq = []
with open(assembly_file, 'r') as orf_file:
    for read_record in SeqIO.parse(orf_file, "fasta"):
        string = read_record.id
        string2 = string.split('|')
        for ORF in K11181_ORFname:
            #print(ORF)
            #print(string2[0])
            if ORF == string2[0]:
```

```
            K11181_ORFseq.append(read_record)
marker = K11181_ORFseq ###ORFS
#### get the database seqs
db = []
with open('/srv/scratch/jgi/Cavlab_pipeline_resources/v1/K11181_pathway_database_v1.fasta',
'r') as dsrAB_file:
    db = list(SeqIO.parse(dsrAB_file, "fasta"))
#### make a prediction
for ORF in marker:
    scorei = []
    for seq in db:
        align = pairwise2.align.localms(ORF.seq, seq.seq, 2, -1, -.5, -.1, score_only = 1)
        scorei.append(align)
        seq.description = align
    keep = max(scorei)
    cat = []
    for seq in db:
        if seq.description >= keep:
            cat.append(seq.id.split('$')[0])
    dis = 0
    ox = 0
    oth = 0
    un = 0
    for obs in cat:
        if obs == 'Reductive':
            dis = dis + 1
        elif obs == 'Oxidative':
            ox = ox + 1
        elif obs == 'Other':
            oth = oth + 1
        else:
            un = un + 1
    if keep > limit:
        #assign
        if dis > ox and dis > oth:
            assignment = 'Reductive'
            K11181r.append(float(ORF.id.split('|')[1][10:]))
```

```
        elif ox > dis and ox > oth:
            assignment = 'Oxidative'
            K11181o.append(float(ORF.id.split('|')[1][10:]))
        elif oth > dis and oth > ox:
            assignment = 'Other'
        else:
            assignment = 'Unknown'
print('K11181 done')

#### use for coverage info
K00362c = sum(K00362)
K00363c = sum(K00363)
K03385c = sum(K03385)
K15876c = sum(K15876)
K17877c = sum(K17877)
K00366c = sum(K00366)
K02305c = sum(K02305)
K04561c = sum(K04561)
K00376c = sum(K00376)
K00531c = sum(K00531)
K02586c = sum(K02586)
K02591c = sum(K02591)
K02588c = sum(K02588)
K10535c = sum(K10535)
K10944c = sum(K10944)
K01601c = sum(K01601)
K01602c = sum(K01602)
K00855c = sum(K00855)
K15230c = sum(K15230)
K15231c = sum(K15231)
K15234c = sum(K15234)
K15233c = sum(K15233)
K15232c = sum(K15232)
K00192c = sum(K00192)
K00198c = sum(K00198)
K03518c = sum(K03518)
K03519c = sum(K03519)
```

K03520c = sum(K03520)
K14138c = sum(K14138)
K02256c = sum(K02256)
K02262c = sum(K02262)
K02274c = sum(K02274)
K02276c = sum(K02276)
K00401c = sum(K00401)
K00400c = sum(K00400)
K16157c = sum(K16157)
K16158c = sum(K16158)
K16159c = sum(K16159)
K16161c = sum(K16161)
K00390c = sum(K00390)
K00392c = sum(K00392)
K00380c = sum(K00380)
K00381c = sum(K00381)
K00394rc = sum(K00394r)
K00395rc = sum(K00395r)
K11180rc = sum(K11180r)
K11181rc = sum(K11181r)
K00394oc = sum(K00394o)
K00395oc = sum(K00395o)
K11180oc = sum(K11180o)
K11181oc = sum(K11181o)
K17224c = sum(K17224)
K17227c = sum(K17227)
K17226c = sum(K17226)
K17222c = sum(K17222)
K17223c = sum(K17223)
K17225c = sum(K17225)
K05973c = sum(K05973)
K03821c = sum(K03821)
K15342c = sum(K15342)
K09951c = sum(K09951)
K07012c = sum(K07012)
K07475c = sum(K07475)
K19088c = sum(K19088)

K19123c = sum(K19123)

K19127c = sum(K19127)

K07016c = sum(K07016)

K19138c = sum(K19138)

K19141c = sum(K19141)

K09952c = sum(K09952)

K19137c = sum(K19137)

K07464c = sum(K07464)

K02703c = sum(K02703)

K02706c = sum(K02706)

K02705c = sum(K02705)

K02704c = sum(K02704)

K02707c = sum(K02707)

K02708c = sum(K02708)

K02689c = sum(K02689)

K02690c = sum(K02690)

K02691c = sum(K02691)

K02692c = sum(K02692)

K02693c = sum(K02693)

K02694c = sum(K02694)

K08928c = sum(K08928)

K08929c = sum(K08929)

K08940c = sum(K08940)

K08941c = sum(K08941)

K08942c = sum(K08942)

K08943c = sum(K08943)

K04643c = sum(K04643)

K04642c = sum(K04642)

K04641c = sum(K04641)

K04250c = sum(K04250)

K00909c = sum(K00909)

K01428c = sum(K01428)

K01429c = sum(K01429)

K01430c = sum(K01430)

K00111c = sum(K00111)

K00112c = sum(K00112)

K00113c = sum(K00113)

```
K00096c = sum(K00096)
K00518c = sum(K00518)
K04564c = sum(K04564)
K04565c = sum(K04565)
K16627c = sum(K16627)
K06164c = sum(K06164)
K05780c = sum(K05780)
K06165c = sum(K06165)
K06166c = sum(K06166)
K06167c = sum(K06167)
K06163c = sum(K06163)
K06162c = sum(K06162)
K08977c = sum(K08977)
K09836c = sum(K09836)
K15746c = sum(K15746)
K16953c = sum(K16953)
K17486c = sum(K17486)
K20452c = sum(K20452)
K07306c = sum(K07306)
K17218c = sum(K17218)
K03553c = sum(K03553)

#### calculate pathways
#N cycle
dissimilatory_nitrogen_reduction_c = np.average([K00362c, K00363c, K03385c, K15876c])
assimilatory_nitrogen_reduction_c = np.average([K17877c, K00366c])
denitrification_c = np.average([K02305c, K04561c, K00376c])
nitrogen_fixation_c = np.average([K00531c, K02586c, K02591c, K02588c])
nitrification_c = np.average([K10535c, K10944c])
#C cycle
calvin_cycle_c = np.average([K01601c, K01602c, K00855c])
rTCA_c = np.average([K15230c, K15231c, K15234c, K15233c, K15232c])
WL_c = np.average([K00192c, K00198c, K14138c])
carbon_fixation_c = calvin_cycle_c + rTCA_c + WL_c
respiration_c = np.average([K02256c, K02262c, K02274c, K02276c])
CO_oxidation_c = np.average([K03520c, K03519c, K03518c])
methanogenesis_c = np.average([K00401c, K00400c])
```

```
methane_oxidation_c = np.average([K16157c, K16158c, K16159c, K16161c])
#S cycle
assimilatory_sulfur_reduction_c = np.average([K00390c, K00392c, K00380c, K00381c])
dissimilatory_sulfur_reduction_c = np.average([K00394rc, K00395rc, K11180rc, K11181rc])
dissimilatory_sulfur_oxidation_c = np.average([K00394oc, K00395oc, K11180oc, K11181oc])
sox_c = np.average([K17224c, K17227c, K17226c, K17222c, K17223c, K17225c])
#PHA storage
PHA_bioynthesis_c = np.average([K05973c, K03821c])
#CRISPR
CRISPR_overall_c = np.average([K15342c, K09951c])
CRISPR_1I_c = np.average([K07012c, K07475c])
CRISPR_1IA_c = np.average([K19088c])
CRISPR_1IE_c = np.average([K19123c])
CRISPR_1IF_c = np.average([K19127c])
CRISPR_1III_c = np.average([K07016c])
CRISPR_1IIIA_c = np.average([K19138c])
CRISPR_1IIIB_c = np.average([K19141c])
CRISPR_2II_c = np.average([K09952c])
CRISPR_2IIA_c = np.average([K19137c])
CRISPR_2IIB_c = np.average([K07464c])
#photosynthesis
photosystem_II_c = np.average([K02703c, K02706c, K02705c, K02704c, K02707c, K02708c])
photosystem_I_c = np.average([K02689c, K02690c, K02691c, K02692c, K02693c, K02694c])
anoxygenic_photosystem_II_c = np.average([K08928c, K08929c])
anoxygenic_photosystem_I_c = np.average([K08940c, K08941c, K08942c, K08943c])
#rhodopsins
rhodopsins_c = np.average([K04643c, K04642c, K04641c, K04250c, K00909c])
#urea
urea_c = np.average([K01428c, K01429c, K01430c])
#glycerol
glycerol_c = np.average([K00111c, K00112c, K00113c, K00096c])
#O2 related
superoxidedismutase_c = np.average([K00518c, K04564c, K04565c, K16627c])
#phosphonate catabolism
phosphonate_catabolism_c = np.average([K06164c, K05780c, K06165c, K06166c, K06167c,
K06163c, K06162c])
#pigments
```

```
bacterioruberin_c = K08977c
bacteriorhodopsin_c = K04641c #also included in rhodopsins
astaxanthin_c = np.average([K09836c, K15746c])
#sulfur genes from organic
DMSO_reduction_c = np.average([K16953c, K17486c, K20452c, K07306c])
sqrA_c = K17218c
#normalization
recA_c = K03553c

#### use for counts
K00362n = (len(K00362)-1)
K00363n = (len(K00363)-1)
K03385n = (len(K03385)-1)
K15876n = (len(K15876)-1)
K17877n = (len(K17877)-1)
K00366n = (len(K00366)-1)
K02305n = (len(K02305)-1)
K04561n = (len(K04561)-1)
K00376n = (len(K00376)-1)
K00531n = (len(K00531)-1)
K02586n = (len(K02586)-1)
K02591n = (len(K02591)-1)
K02588n = (len(K02588)-1)
K10535n = (len(K10535)-1)
K10944n = (len(K10944)-1)
K01601n = (len(K01601)-1)
K01602n = (len(K01602)-1)
K00855n = (len(K00855)-1)
K15230n = (len(K15230)-1)
K15231n = (len(K15231)-1)
K15234n = (len(K15234)-1)
K15233n = (len(K15233)-1)
K15232n = (len(K15232)-1)
K00192n = (len(K00192)-1)
K00198n = (len(K00198)-1)
K03518n = (len(K03518)-1)
K03519n = (len(K03519)-1)
```

K03520n = (len(K03520)-1)
K14138n = (len(K14138)-1)
K02256n = (len(K02256)-1)
K02262n = (len(K02262)-1)
K02274n = (len(K02274)-1)
K02276n = (len(K02276)-1)
K00401n = (len(K00401)-1)
K00400n = (len(K00400)-1)
K16157n = (len(K16157)-1)
K16158n = (len(K16158)-1)
K16159n = (len(K16159)-1)
K16161n = (len(K16161)-1)
K00390n = (len(K00390)-1)
K00392n = (len(K00392)-1)
K00380n = (len(K00380)-1)
K00381n = (len(K00381)-1)
K00394rn = (len(K00394r))
K00395rn = (len(K00395r))
K11180rn = (len(K11180r))
K11181rn = (len(K11181r))
K00394on = (len(K00394o))
K00395on = (len(K00395o))
K11180on = (len(K11180o))
K11181on = (len(K11181o))
K17224n = (len(K17224)-1)
K17227n = (len(K17227)-1)
K17226n = (len(K17226)-1)
K17222n = (len(K17222)-1)
K17223n = (len(K17223)-1)
K17225n = (len(K17225)-1)
K05973n = (len(K05973)-1)
K03821n = (len(K03821)-1)
K15342n = (len(K15342)-1)
K09951n = (len(K09951)-1)
K07012n = (len(K07012)-1)
K07475n = (len(K07475)-1)
K19088n = (len(K19088)-1)

K19123n = (len(K19123)-1)
K19127n = (len(K19127)-1)
K07016n = (len(K07016)-1)
K19138n = (len(K19138)-1)
K19141n = (len(K19141)-1)
K09952n = (len(K09952)-1)
K19137n = (len(K19137)-1)
K07464n = (len(K07464)-1)
K02703n = (len(K02703)-1)
K02706n = (len(K02706)-1)
K02705n = (len(K02705)-1)
K02704n = (len(K02704)-1)
K02707n = (len(K02707)-1)
K02708n = (len(K02708)-1)
K02689n = (len(K02689)-1)
K02690n = (len(K02690)-1)
K02691n = (len(K02691)-1)
K02692n = (len(K02692)-1)
K02693n = (len(K02693)-1)
K02694n = (len(K02694)-1)
K08928n = (len(K08928)-1)
K08929n = (len(K08929)-1)
K08940n = (len(K08940)-1)
K08941n = (len(K08941)-1)
K08942n = (len(K08942)-1)
K08943n = (len(K08943)-1)
K04643n = (len(K04643)-1)
K04642n = (len(K04642)-1)
K04641n = (len(K04641)-1)
K04250n = (len(K04250)-1)
K00909n = (len(K00909)-1)
K01428n = (len(K01428)-1)
K01429n = (len(K01429)-1)
K01430n = (len(K01430)-1)
K00111n = (len(K00111)-1)
K00112n = (len(K00112)-1)
K00113n = (len(K00113)-1)

```
K00096n = (len(K00096)-1)
K00518n = (len(K00518)-1)
K04564n = (len(K04564)-1)
K04565n = (len(K04565)-1)
K16627n = (len(K16627)-1)
K06164n = (len(K06164)-1)
K05780n = (len(K05780)-1)
K06165n = (len(K06165)-1)
K06166n = (len(K06166)-1)
K06167n = (len(K06167)-1)
K06163n = (len(K06163)-1)
K06162n = (len(K06162)-1)
K08977n = (len(K08977)-1)
K09836n = (len(K09836)-1)
K15746n = (len(K15746)-1)
K16953n = (len(K16953)-1)
K17486n = (len(K17486)-1)
K20452n = (len(K20452)-1)
K07306n = (len(K07306)-1)
K17218n = (len(K17218)-1)
K03553n = (len(K03553)-1)

#### calculate pathways
#N cycle
dissimilatory_nitrogen_reduction_n = np.average([K00362n, K00363n, K03385n, K15876n])
assimilatory_nitrogen_reduction_n = np.average([K17877n, K00366n])
denitrification_n = np.average([K02305n, K04561n, K00376n])
nitrogen_fixation_n = np.average([K00531n, K02586n, K02591n, K02588n])
nitrification_n = np.average([K10535n, K10944n])
#C cycle
calvin_cycle_n = np.average([K01601n, K01602n, K00855n])
rTCA_n = np.average([K15230n, K15231n, K15234n, K15233n, K15232n])
WL_n = np.average([K00192n, K00198n, K14138n])
carbon_fixation_n = calvin_cycle_n+rTCA_n+WL_n
respiration_n = np.average([K02256n, K02262n, K02274n, K02276n])
CO_oxidation_n = np.average([K03520n, K03519n, K03518n])
methanogenesis_n = np.average([K00401n, K00400n])
```

```
methane_oxidation_n = np.average([K16157n, K16158n, K16159n, K16161n])
#S cycle
assimilatory_sulfur_reduction_n = np.average([K00390n, K00392n, K00380n, K00381n])
dissimilatory_sulfur_reduction_n = np.average([K00394rn, K00395rn, K11180rn, K11181rn])
dissimilatory_sulfur_oxidation_n = np.average([K00394on, K00395on, K11180on, K11181on])
sox_n = np.average([K17224n, K17227n, K17226n, K17222n, K17223n, K17225n])
#PHA storage
PHA_bioynthesis_n = np.average([K05973n, K03821n])
#CRISPR
CRISPR_overall_n = np.average([K15342n, K09951n])
CRISPR_1I_n = np.average([K07012n, K07475n])
CRISPR_1IA_n = np.average([K19088n])
CRISPR_1IE_n = np.average([K19123n])
CRISPR_1IF_n = np.average([K19127n])
CRISPR_1III_n = np.average([K07016n])
CRISPR_1IIIA_n = np.average([K19138n])
CRISPR_1IIIB_n = np.average([K19141n])
CRISPR_2II_n = np.average([K09952n])
CRISPR_2IIA_n = np.average([K19137n])
CRISPR_2IIB_n = np.average([K07464n])
#photosynthesis
photosystem_II_n = np.average([K02703n, K02706n, K02705n, K02704n, K02707n,
K02708n])
photosystem_I_n = np.average([K02689n, K02690n, K02691n, K02692n, K02693n, K02694n])
anoxygenic_photosystem_II_n = np.average([K08928n, K08929n])
anoxygenic_photosystem_I_n = np.average([K08940n, K08941n, K08942n, K08943n])
#rhodopsins
rhodopsins_n = np.average([K04643n, K04642n, K04641n, K04250n, K00909n])
#urea
urea_n = np.average([K01428n, K01429n, K01430n])
#glycerol
glycerol_n = np.average([K00111n, K00112n, K00113n, K00096n])
#O2 related
superoxidedismutase_n = np.average([K00518n, K04564n, K04565n, K16627n])
#phosphonate catabolism
phosphonate_catabolism_n = np.average([K06164n, K05780n, K06165n, K06166n, K06167n,
K06163n, K06162n])
```

```
#pigments
bacterioruberin_n = K08977n
bacteriorhodopsin_n = K04641n #also included in rhodopsins
astaxanthin_n = np.average([K09836n, K15746n])
#sulfur genes from organic
DMSO_reduction_n = np.average([K16953n, K17486n, K20452n, K07306n])
sqrA_n = K17218n
#normalization
recA_n = K03553n

#### write data out
header = ['dissimilatory_nitrogen_reduction', 'assimilatory_nitrogen_reduction', 'denitrification',
'nitrogen_fixation', 'nitrification', 'calvin_cycle', 'rTCA', 'WL', 'carbon_fixation', 'respiration',
'CO_oxidation', 'methanogenesis', 'methane_oxidation', 'assimilatory_sulfur_reduction',
'dissimilatory_sulfur_reduction', 'dissimilatory_sulfur_oxidation', 'sox', 'PHA_bioynthesis',
'CRISPR_overall', 'CRISPR_1I', 'CRISPR_1IA', 'CRISPR_1IE', 'CRISPR_1IF', 'CRISPR_1III',
'CRISPR_1IIIA', 'CRISPR_1IIIB', 'CRISPR_2II', 'CRISPR_2IIA', 'CRISPR_2IIB',
'photosystem_II', 'photosystem_I', 'anoxygenic_photosystem_II', 'anoxygenic_photosystem_I',
'rhodopsins', 'urea', 'glycerol', 'superoxidedismutase', 'phosphonate_catabolism',
'bacterioruberin', 'bacteriorhodopsin', 'astaxanthin', 'DMSO_reduction', 'sqrA', 'recA', 'total',
'issues']
results_c = [dissimilatory_nitrogen_reduction_c, assimilatory_nitrogen_reduction_c,
denitrification_c, nitrogen_fixation_c, nitrification_c, calvin_cycle_c, rTCA_c, WL_c,
carbon_fixation_c, respiration_c, CO_oxidation_c, methanogenesis_c, methane_oxidation_c,
assimilatory_sulfur_reduction_c, dissimilatory_sulfur_reduction_c,
dissimilatory_sulfur_oxidation_c, sox_c, PHA_bioynthesis_c, CRISPR_overall_c,
CRISPR_1I_c, CRISPR_1IA_c, CRISPR_1IE_c, CRISPR_1IF_c, CRISPR_1III_c,
CRISPR_1IIIA_c, CRISPR_1IIIB_c, CRISPR_2II_c, CRISPR_2IIA_c, CRISPR_2IIB_c,
photosystem_II_c, photosystem_I_c, anoxygenic_photosystem_II_c,
anoxygenic_photosystem_I_c, rhodopsins_c, urea_c, glycerol_c, superoxidedismutase_c,
phosphonate_catabolism_c, bacterioruberin_c, bacteriorhodopsin_c, astaxanthin_c,
DMSO_reduction_c, sqrA_c, recA_c]
results_n = [dissimilatory_nitrogen_reduction_n, assimilatory_nitrogen_reduction_n,
denitrification_n, nitrogen_fixation_n, nitrification_n, calvin_cycle_n, rTCA_n, WL_n,
carbon_fixation_n, respiration_n, CO_oxidation_n, methanogenesis_n, methane_oxidation_n,
assimilatory_sulfur_reduction_n, dissimilatory_sulfur_reduction_n,
dissimilatory_sulfur_oxidation_n, sox_n, PHA_bioynthesis_n, CRISPR_overall_n,
```

CRISPR_1I_n, CRISPR_1IA_n, CRISPR_1IE_n, CRISPR_1IF_n, CRISPR_1III_n, CRISPR_1IIIA_n, CRISPR_1IIIB_n, CRISPR_2II_n, CRISPR_2IIA_n, CRISPR_2IIB_n, photosystem_II_n, photosystem_I_n, anoxygenic_photosystem_II_n, anoxygenic_photosystem_I_n, rhodopsins_n, urea_n, glycerol_n, superoxidedismutase_n, phosphonate_catabolism_n, bacterioruberin_n, bacteriorhodopsin_n, astaxanthin_n, DMSO_reduction_n, sqrA_n, recA_n]

all_header = ['K00362', 'K00363', 'K03385', 'K15876', 'K17877', 'K00366', 'K02305', 'K04561', 'K00376', 'K00531', 'K02586', 'K02591', 'K02588', 'K10535', 'K10944', 'K01601', 'K01602', 'K00855', 'K15230', 'K15231', 'K15234', 'K15233', 'K15232', 'K00192', 'K00198', 'K14138', 'K02256', 'K02262', 'K02274', 'K02276', 'K03520', 'K03519', 'K03518', 'K00401', 'K00400', 'K16157', 'K16158', 'K16159', 'K16161', 'K00390', 'K00392', 'K00380', 'K00381', 'K17224', 'K17227', 'K17226', 'K17222', 'K17223', 'K17225', 'K05973', 'K03821', 'K15342', 'K09951', 'K07012', 'K07475', 'K19088', 'K19123', 'K19127', 'K07016', 'K19138', 'K19141', 'K09952', 'K19137', 'K07464', 'K02703', 'K02706', 'K02705', 'K02704', 'K02707', 'K02708', 'K02689', 'K02690', 'K02691', 'K02692', 'K02693', 'K02694', 'K08928', 'K08929', 'K08940', 'K08941', 'K08942', 'K08943', 'K04643', 'K04642', 'K04641', 'K04250', 'K00909', 'K01428', 'K01429', 'K01430', 'K00111', 'K00112', 'K00113', 'K00096', 'K00518', 'K04564', 'K04565', 'K16627', 'K06164', 'K05780', 'K06165', 'K06166', 'K06167', 'K06163', 'K06162', 'K08977', 'K09836', 'K15746', 'K16953', 'K17486', 'K20452', 'K07306', 'K17218', 'K03553', 'K00394r', 'K00395r', 'K11180r', 'K11181r', 'K00394o', 'K00395o', 'K11180o', 'K11181o']

all_c = [K00362c, K00363c, K03385c, K15876c, K17877c, K00366c, K02305c, K04561c, K00376c, K00531c, K02586c, K02591c, K02588c, K10535c, K10944c, K01601c, K01602c, K00855c, K15230c, K15231c, K15234c, K15233c, K15232c, K00192c, K00198c, K14138c, K02256c, K02262c, K02274c, K02276c, K03520c, K03519c, K03518c, K00401c, K00400c, K16157c, K16158c, K16159c, K16161c, K00390c, K00392c, K00380c, K00381c, K17224c, K17227c, K17226c, K17222c, K17223c, K17225c, K05973c, K03821c, K15342c, K09951c, K07012c, K07475c, K19088c, K19123c, K19127c, K07016c, K19138c, K19141c, K09952c, K19137c, K07464c, K02703c, K02706c, K02705c, K02704c, K02707c, K02708c, K02689c, K02690c, K02691c, K02692c, K02693c, K02694c, K08928c, K08929c, K08940c, K08941c, K08942c, K08943c, K04643c, K04642c, K04641c, K04250c, K00909c, K01428c, K01429c, K01430c, K00111c, K00112c, K00113c, K00096c, K00518c, K04564c, K04565c, K16627c, K06164c, K05780c, K06165c, K06166c, K06167c, K06163c, K06162c, K08977c, K09836c, K15746c, K16953c, K17486c, K20452c, K07306c, K17218c, K03553c, K00394rc, K00395rc, K11180rc, K11181rc, K00394oc, K00395oc, K11180oc, K11181oc]

all_n = [K00362n, K00363n, K03385n, K15876n, K17877n, K00366n, K02305n, K04561n, K00376n, K00531n, K02586n, K02591n, K02588n, K10535n, K10944n, K01601n, K01602n, K00855n, K15230n, K15231n, K15234n, K15233n, K15232n, K00192n, K00198n, K14138n,

```
K02256n, K02262n, K02274n, K02276n, K03520n, K03519n, K03518n, K00401n, K00400n,
K16157n, K16158n, K16159n, K16161n, K00390n, K00392n, K00380n, K00381n, K17224n,
K17227n, K17226n, K17222n, K17223n, K17225n, K05973n, K03821n, K15342n, K09951n,
K07012n, K07475n, K19088n, K19123n, K19127n, K07016n, K19138n, K19141n, K09952n,
K19137n, K07464n, K02703n, K02706n, K02705n, K02704n, K02707n, K02708n, K02689n,
K02690n, K02691n, K02692n, K02693n, K02694n, K08928n, K08929n, K08940n, K08941n,
K08942n, K08943n, K04643n, K04642n, K04641n, K04250n, K00909n, K01428n, K01429n,
K01430n, K00111n, K00112n, K00113n, K00096n, K00518n, K04564n, K04565n, K16627n,
K06164n, K05780n, K06165n, K06166n, K06167n, K06163n, K06162n, K08977n, K09836n,
K15746n, K16953n, K17486n, K20452n, K07306n, K17218n, K03553n, K00394rn, K00395rn,
K11180rn, K11181rn, K00394on, K00395on, K11180on, K11181on]
with open('%s' + file_root + '.assembled.faa.KO_summary.csv', 'w') as count_file:
    count_csv = csv.writer(count_file)
    count_csv.writerow(['pathway', 'by coverage', 'by count'])
    for i in range(len(header)-2):
        count_csv.writerow([header[i], results_c[i],cov_norm,results_n[i],count_norm])
    count_csv.writerow([header[-2], cov_norm, count_norm])
    count_csv.writerow([header[-1], len(no_cov), len(no_cov)])
    count_csv.writerow(['single markers below here'])
    for i in range(len(all_header)):
        count_csv.writerow([all_header[i], all_c[i], all_n[i]])
print('KEGG done')
"""%(ass_num, head_dir + ass_num + '.assembled_cov.faa', KEGG_file, head_dir)

with open(res_dir + 'KOpathways_v8.py', 'w') as KEGG_script:
    KEGG_script.write(KEGG_code)

## write COG and KEGG bash
KEGG_bash = '''#!/bin/bash
#PBS -N SCRATCH
#PBS -l nodes=1:ppn=1
#PBS -l vmem=12gb
#PBS -l walltime=48:00:00
#PBS -o %sCOGKEGG_Output_Report_1
#PBS -o %sCOGKEGG_Error_Report_1
#PBS -M rcavlab@gmail.com
#PBS -m ae
```

```
module load python/3.5.2
cd %s

python3 COG_summarize_v2.py
python3 KOpathways_v8.py
'''%(res_dir, res_dir, res_dir)

with open(res_dir + 'COGKEGG.pbs', 'w') as KEGG_pbs:
    KEGG_pbs.write(KEGG_bash)

#### write metabat mapping
#### write 2500 filter python script
filter_script = """# -*- coding: utf-8 -*-
'''Created on Tue Jul 12 15:54:25 2016
@author: jay3
Based on comb_assemlies_v3.py. Filters assemblies for contigs less than 2500
for use with MetaBAT.'''

import Bio.SeqIO as SeqIO
seqs = []
file_name='%s'
with open(file_name, 'r') as read_file:
    for record in SeqIO.parse(read_file, "fasta"):
        if len(record.seq) >= 2500:
            seqs.append(record)
with open('%s' + '.assembled_2500.fna', 'w') as comb_file:
    SeqIO.write(seqs, comb_file, "fasta")
"""%(DNAass_file, head_dir + 'metabat/' + ass_num)

with open(res_dir + 'assemblies_filter_v1.py', 'w') as filter_py:
    filter_py.write(filter_script)

#### write mapping bash script
metabat_script = '''#!/bin/bash
#PBS -N SCRATCH
#PBS -l nodes=1:ppn=4
#PBS -l vmem=31gb
```

```
#PBS -l walltime=12:00:00
#PBS -o %sSample2500_map_Output_Report_1
#PBS -o %sSample2500_map_Error_Report_1
#PBS -M rcavlab@gmail.com
#PBS -m ae
cd %s
module load python/3.5.2
python3 assemblies_filter_v1.py

module load bbmap/35.82
export _JAVA_OPTIONS="-Xmx28g"
cd %s
bbmap.sh ref=%s.assembled_2500.fna

cd %s
bbmap.sh ref=%s.assembled_2500.fna in='%s' outm=%s outu=%s idfilter=.9 threads=4
'''%(res_dir, res_dir, res_dir, head_dir + 'metabat/', ass_num, raw_dir, head_dir + 'metabat/' +
ass_num, read_file.split('/')[-1], head_dir + 'metabat/reads_in_' + ass_num +
'_assembled2500.fna', head_dir + 'metabat/reads_not_in_' + ass_num + '_assembled2500.fna')
#'../../'+read_file.split('/')[-1]

with open(res_dir + 'sample2500_map.pbs', 'w') as metabat_pbs:
    metabat_pbs.write(metabat_script)

#### write preprocess bash
preprocess_script = '''#!/bin/bash
#PBS -N SCRATCH
#PBS -l nodes=1:ppn=1
#PBS -l vmem=8gb
#PBS -l walltime=12:00:00
#PBS -o %sPreprocess_Output_Report_1
#PBS -o %sPreprocess_Error_Report_1
#PBS -M rcavlab@gmail.com
#PBS -m ae
module load python/3.5.2
cd %s
python3 append_cov2ORFs.py
```

```
'''%(res_dir, res_dir, res_dir)


with open(res_dir + 'preprocess.pbs', 'w') as preprocess_pbs:
    preprocess_pbs.write(preprocess_script)


#### submit first job
command = '''cd %s
qsub %s'''%(res_dir, res_dir + 'preprocess.pbs') #the change dir is so that the SCRATCH######
report is placed in resources
screen = subprocess.check_output(command, shell = True)
screen = screen.decode()[0:7]
with open(res_dir + 'job_log.txt', 'a') as job_log:
    job_csv = csv.writer(job_log)
    job_csv.writerow(['Preprocess', screen])
print('done')
```

Appendix C

Cavlab pipeline v4.1 — the latest metagenome analysis pipeline

Code C1. Python code for Cavlab pipeline v4.1. This code was developed for the analysis of metagenomes annotated by JGI's IMG system. It verifies input resource files and file paths (Resource file verification component), creates an output folder structure (Output folder structure preparation component), and performs contig taxonomic assignment and species abundance calculations (Contig taxonomy and abundance estimation component), COG functional potential analyses (COG functional potential analysis component), KEGG functional potential analyses (KEGG functional potential analysis component), and protein taxonomic assignments using MEGAN (DIAMOND and MEGAN6 protein taxonomy component). Each of these analyses is described in details in Chapter 2.

```
""" Latest version of Cavlab pipeline.
@author: Pratibha Panwar
This is the main script for the metagenome analysis pipeline, which should be run
from the JGI IMG metagenome folder. The pipeline depends on a consistent folder structure -
IMG_Data and QC_and_Genome_Assembly/QC_Filtered_Raw_Data.

v1.2:
Decided to keep databases in a single folder rather than rewrite on every run. Specified # of
threads in DIAMOND and BBMap lines. Minimum request job time has been set to 12 hrs.

@Jay3
v1.3a:
Changed the version of PhyloSift that is used - direct it to use the copy of PhyloSift located in
Katana scratch so that the database version is not updated without our knowledge - for Version
3a Michelle Allen added the --config flag to force PhyloSift to use the config file phylosiftrc
(which contains a flag not to update the database). @Michelle (23rd March 2017)

v1.3b:
Modified the output head_folder naming format. The folder will now be named as
Cav_YYMMDD e.g. Cav_170330 for 30 March 2017. @Pratibha (30 March, 2017)

v1.3b.1:
Runs the pipeline without running the PhyloSift component. @Pratibha (18 April, 2017)

v1.4.1:
The output and error files have been merged. KEGG process wall-time has been increased from
48 to 60 h. @Pratibha (20 April, 2017)

v1.4.2
Modification in the code for selecting protein sequence file, to ensure that '.assembled.faa'
protein file would be picked and not the '.unassembled_illumina.faa' protein file.
```

In addition, the wall-time for COGKEGG has been increased to 96 h.
@Pratibha (17 May, 2017)

v1.4.2a
The wall-time for COGKEGG has been increased to 120 h.
@Pratibha (25 May, 2017)

v1.5
Separated the COG and KEGG processes. COG wall-time is still 120 h, but KEGG has been given only 12 h wall-time.
@Pratibha (28 May, 2017)

v2.0
Major updates to the COG process to reduce its process time. COG process has now been allotted 12 h wall-time, in place of 120 h. Moreover, the COG conversion file has been updated.
PhyloSift component has been removed.
CRISPR component has been added.
In MEGAN, the KEGG mapping file has been included.
@Pratibha (3 July, 2017)

v2.1
Some metagenomes do not have the raw read file (fastq) file in the usual IMG metagenome folder (QC_Filtered_Raw_Data). Therefore, added new commands to search for fastq files in other relevant folders.
Associated changes were made in MetaBATt section.
Also, made changes to KEGG code.
@Pratibha (13 July, 2017)

v2.2
This version resolves the issue with the use of deprecated mapping files for MEGAN6. The MEGAN6 component now uses the latest v6.8.18 and the associated updated mapping files. The NCBI-NR database has also been updated, which is why the mapping files had to be updated. The only drawback of this update is that the KEGG mapping file is no longer valid. Hence, this MEGAN6 output file will have taxonomy and eggNOG data, but no KEGG data.
@Pratibha (7 August, 2017)

v2.2a
This version includes a minor addition to the CRISPR code. Apart from creating a CSV of CRISPRs and their corresponding reads, the script will also create a FASTA file

containing the CRISPR spacer sequences as records and corresponding contig name (along with position of CRISPR spacer) as record.id. As each spacer can be present in more than one contig and each contig can contain more than one spacer, mentioning the position in the record.id should help to distinguish between such records. @Pratibha (8 August, 2017)

v2.2b

This version includes a minor addition to the MEGAN6 process: Interpro mapping file has been included. The MEGAN6 output will now include Interpro2GO data.
KEGG mapping has been removed from MEGAN6 component, as there are no accession to kegg mapping files available for MEGAN6 community edition. @Pratibha (10 August, 2017)

v3.0

This version includes major updates to MEGAN6 process, with addition of contig taxonomic mapping and read-based relative abundance estimation.
The folder structure has also been revised to make it more comprehensive.
The protein sequence file pre-processing has been changed, so that the record ID includes protein name instead of associated contig coverage.
The CRISPR analysis and early steps of MetaBAT have been removed from this version. The MetaBAT pipeline will be separate from this pipeline and will be run by Michelle Allen. @Pratibha (2 February, 2018)

v3.1

The taxonomic abundance component has been removed from this version of the pipeline, as a new method for calculating abundances has been developed. @Pratibha (15 February, 2018)

v3.1a

Slight modifications were made in the way coverage and mapping files were read and used. @Pratibha (19 February, 2018)

v3.2

Includes modifications to the code for reading the contig file. An extra step has been added to ensure that the correct contig file (scaffold file) is selected from the QC_and_Assembly folder. @Pratibha (27 February, 2018)

v3.3

Changes have been made to KEGG and COG to acquire the total coverage of proteins that do not fall in any of the decided categories.
The COG output will no longer be a fraction of the total coverage of all proteins in the metagenome, although this information will still be provided in the output file.
The output head folder name will now include the pipeline version. @Pratibha (1 March, 2018)

```
v4
Output folder names have been changed.
LAST and MEGAN-LR for contig taxonomy were removed.
A script for IMG protein taxonomy-based contig taxonomy analysis and abundance estimation was added.
Updated the list of KEGG numbers and pathways/enzymes analysed in KEGG analysis component. Updated the list of KEGG database files and their paths.
Updated the MEGAN mapping files for protein taxonomy and function section. @Pratibha (28-29 May, 2020)

v4.1
Added the new JGI file nomenclatures to the script.
Fixed an issue with the latest COG files downloaded from JGI IMG, by removing an unnecessary blank column, if present. @Pratibha (13 June, 2020)
"""

from datetime import date
import os
import subprocess
from Bio import SeqIO
import sys
import csv
current_dir = subprocess.check_output('pwd', shell = True).decode().strip() + '/' # get current dir

#### Resource file verification component ####
go = []
prot, phylo, cog, kegg, mapf, product, cov = 0, 0, 0, 0, 0, 0, 0
if os.path.isdir('./IMG_Data') == True:
    for file in os.listdir('./IMG_Data'):
        if file[-13:] == 'assembled.faa' or (file[-12:] == 'proteins.faa' and file.split('_')[1] != 'prodigal' and file.split('_')[1] != 'genemark'): # find protein sequence file
            assembly_num = file.split('.')[0]
            prot = 1
            PROTEIN_file = current_dir + 'IMG_Data/' + file
        if len(file) > 19: # to avoid python index error
            if file[-19:] == 'assembled.phylodist' or file[-18:] == 'gene_phylogeny.tsv': # find phylodist annotation file
```

```
            phylo = 1
            protTAXA_file = current_dir + 'IMG_Data/' + file
        if file[-16:] == 'assembled.faa.KO' or file[-12:] == 'assembled.KO' or file[-6:] == 'ko.tsv': #
find KEGG annotation file
            kegg = 1
            KEGG_file = current_dir + 'IMG_Data/' + file
        if len(file) > 24: # to avoid python index error
            if file[-19:] == 'assembled.names_map' or file[-24:] == 'contig_names_mapping.tsv': #
find scaffold to conig mapping file
                mapf = 1
                MAP_file = current_dir + 'IMG_Data/' + file
        if len(file) > 23: # to avoid python index error
            if file[-23:] == 'assembled.product.names' or file[-23:] == 'assembled.product_names' or
file[-17:] == 'product_names.tsv': # find protein annotation file with product name
                product = 1
                PRODUCT_file = current_dir + 'IMG_Data/' + file
else:
    print('Error: IMG_Data folder does not exist.')

if os.path.isdir('./IMG_Data') == True:
    for file in os.listdir('./IMG_Data'):
        if file[-17:] == 'assembled.faa.COG' or file[-13:] == 'assembled.COG': # find COG
annotation file
            cog = 1
            COG_file = current_dir + 'IMG_Data/' + file
        elif file[-7:] == 'cog.gff':
            with open(current_dir + 'IMG_Data/' + file, 'r') as inf: # to check if there is an empty
column between protein ID and COG number and remove it
                infc = csv.reader(inf, delimiter = '\t')
                test = next(infc)
                if test[1] == '':
                    with open(current_dir + 'COGfile-mod.txt', 'w', newline = '') as outf:
                        outfc = csv.writer(outf, delimiter = '\t')
                        for row in infc:
                            outfc.writerow([row[0], row[2]])
                    cog = 1
                    COG_file = current_dir + 'COGfile-mod.txt'
```

```
            else:
                cog = 1
                COG_file = current_dir + 'IMG_Data/' + file

if os.path.isdir('./IMG_Data') == True:
    for file in os.listdir('./IMG_Data'):
        if file[-13:] == 'scaffolds.cov' or file[0:17] == 'seq_coverage_file' or file[-14:] ==
'sorted.bam.cov': # find contig coverage file
            cov = 1
            COV_file = current_dir + 'IMG_Data/' + file

if cov == 0: # if coverage file not in IMG_Data folder, check in QC_and_Genome_Assembly
folder
    if os.path.isdir('./QC_and_Genome_Assembly') == True:
        with os.scandir('./QC_and_Genome_Assembly') as direcs:
            for direc in direcs:
                if direc.is_dir():
                    for file in os.listdir(direc):
                        if file == 'covstats.txt':
                            cov = 1
                            COV_file = current_dir + 'QC_and_Genome_Assembly/' + direc.name + '/' +
file

if cov == 0: # if coverage file still not found, look in Assembled_data folder
    if os.path.isdir('./Assembled_data') == True:
        for file in os.listdir('./Assembled_data'):
            if file[-12:] == 'coverage.txt':
                cov = 1
                COV_file = current_dir + 'Assembled_data/' + file

if prot == 0:
    print('Error: Protein sequence file not found.')
else:
    print('Protein sequence file: ', PROTEIN_file)
    go.append(1)
if phylo == 0:
    print('Error: Protein taxonomy file not found.')
```

```
else:
    print('Protein taxonomy file: ', protTAXA_file)
    go.append(1)
if cog == 0:
    print('Error: COG file not found.')
else:
    print('COG file: ', COG_file)
    go.append(1)
if kegg == 0:
    print('Error: KEGG file not found.')
else:
    print('KEGG file: ', KEGG_file)
    go.append(1)
if mapf == 0:
    print('Error: Scaffold to contig mapping file not found.')
else:
    print('Scaffold to contig mapping file: ', MAP_file)
    go.append(1)
if product == 0:
    print('Error: Protein product name file not found.')
else:
    print('Protein product name file: ', PRODUCT_file)
    go.append(1)

if cov == 0:
    print('Error: Contig coverage file not found.')
else:
    print('Contig coverage file: ', COV_file)
    go.append(1)

if os.path.isdir('./QC_and_Genome_Assembly') == True:
    for file in os.listdir('./QC_and_Genome_Assembly'):
        if file[-13:] == 'contigs.fasta': # find scaffold sequence file
            contf = current_dir + 'QC_and_Genome_Assembly/' + file
    maps = {}
    with open(MAP_file, 'r') as mapfile:
        mapcsv = csv.reader(mapfile, delimiter = '\t')
```

```
        for row in mapcsv:
            maps[row[0]] = row[1]
    mapfile.close()
    with open(contf, 'r') as contigs: # prepare contig sequence file from sacffold file
        with open('./final.contigs-mod.fna', 'w') as newcontigs:
            for record in SeqIO.parse(contigs, "fasta"):
                record.id = maps[record.id]
                SeqIO.write(record, newcontigs, "fasta")
    newcontigs.close()
    contigs.close()
    go.append(1)
    CONTIG_file = current_dir + 'final.contigs-mod.fna'
    print('New contig file created: ', CONTIG_file)
elif os.path.isdir('./IMG_Data') == True:
    for file in os.listdir('./IMG_Data'):
        if file[-13:] == 'assembled.fna' or file[-11:] == 'contigs.fna': # if no scaffold file present,
search for contig sequence file
            go.append(1)
            CONTIG_file = current_dir + 'IMG_Data/' + file
            print('Contig sequence file: ', CONTIG_file)
else:
    print('Error: Contig sequence file not found.')

#### Verify COG and KEGG database files
if
os.path.isfile('/srv/scratch/jgi/Cavlab_pipeline_resources/v4/K00394_pathway_database_v1.fast
a') == 1:
    print('K00394_pathway_database_v1.fasta found.')
    go.append(1)
else:
    print('Error: K00394_pathway_database_v1.fasta not found.')

if
os.path.isfile('/srv/scratch/jgi/Cavlab_pipeline_resources/v4/K00395_pathway_database_v1.fast
a') == 1:
    print('K00395_pathway_database_v1.fasta found.')
    go.append(1)
```

```
else:
    print('Error: K00395_pathway_database_v1.fasta not found.')

if
os.path.isfile('/srv/scratch/jgi/Cavlab_pipeline_resources/v4/K11180_pathway_database_v1.fast
a') == 1:
    print('K11180_pathway_database_v1.fasta found.')
    go.append(1)
else:
    print('Error: K11180_pathway_database_v1.fasta not found.')

if
os.path.isfile('/srv/scratch/jgi/Cavlab_pipeline_resources/v4/K11181_pathway_database_v1.fast
a') == 1:
    print('K11181_pathway_database_v1.fasta found.')
    go.append(1)
else:
    print('Error: K11181_pathway_database_v1.fasta not found.')

if
os.path.isfile('/srv/scratch/jgi/Cavlab_pipeline_resources/v4/K00958_pathway_database_v1.fast
a') == 1:
    print('K00958_pathway_database_v1.fasta found.')
    go.append(1)
else:
    print('Error: K00958_pathway_database_v1.fasta not found.')

if
os.path.isfile('/srv/scratch/jgi/Cavlab_pipeline_resources/v4/K10944_pathway_database_v1.fast
a') == 1:
    print('K10944_pathway_database_v1.fasta found.')
    go.append(1)
else:
    print('Error: K10944_pathway_database_v1.fasta not found.')
```

```
if
os.path.isfile('/srv/scratch/jgi/Cavlab_pipeline_resources/v4/K10945_pathway_database_v1.fast
a') == 1:
    print('K10945_pathway_database_v1.fasta found.')
    go.append(1)
else:
    print('Error: K10945_pathway_database_v1.fasta not found.')

if
os.path.isfile('/srv/scratch/jgi/Cavlab_pipeline_resources/v4/K10946_pathway_database_v1.fast
a') == 1:
    print('K10946_pathway_database_v1.fasta found.')
    go.append(1)
else:
    print('Error: K10946_pathway_database_v1.fasta not found.')

if os.path.isfile('/srv/scratch/jgi/Cavlab_pipeline_resources/v4/COG_conversion_v2.csv') == 1:
    print('COG_conversion_v2.csv found.')
    go.append(1)
else:
    print('Error: COG_conversion_v2.csv not found.')

#### Verify all files present
if sum(go) == 17:
    print('All files are present. Good to go.')
else:
    print('Error: Some files are missing. Launch scrubbed.')
    sys.exit()

#### Output folder structure preparation component ####
now = date.today()
if now.month < 10:
    month = '0' + str(now.month)
else:
    month = now.month

if now.day < 10:
```

```
    day = '0' + str(now.day)
else:
    day = now.day

head_folder = 'Cavlab_v4.1_' + str(now.year)[-2:] + str(month) + str(day)
subprocess.call('mkdir ' + head_folder, shell=True)
subprocess.call('mkdir ' + head_folder + '/Resources', shell=True)
subprocess.call('mkdir ' + head_folder + '/Contig_taxonomy_and_abundance', shell=True)
subprocess.call('mkdir ' + head_folder + '/Protein_taxonomy_and_function', shell=True)
subprocess.call('mkdir ' + head_folder + '/COG_KEGG_functions', shell=True)

head_dir = current_dir + head_folder + '/'
res_dir = head_dir + 'Resources/'
contTaxa_dir = head_dir + 'Contig_taxonomy_and_abundance/'
protTaxa_dir = head_dir + 'Protein_taxonomy_and_function/'
cogkegg_dir = head_dir + 'COG_KEGG_functions/'

#### Readme file preparation
readme_text = """This is the head folder for the Cavlab metagenome analysis pipeline v4.1,
created on %s.%s.%s (DDMMYYYY format).

The pipeline covers 3 main analyses:
1. Contig taxonomic classification and abundance estimation [output in
Contig_taxonomy_and_abundance subfolder].
2. Protein taxonomic classification and function analysis [output in
Protein_taxonomy_and_function subfolder].
3. Functional potential analysis [output in COG_KEGG_functions subfolder].

Resources subfolder contains the python and bash scripts, along with the output log/report files.
Jobs_log.txt has a record of the individual jobs created as a part of the pipeline.
Each entry corresponds to a job running on Katana and mentions the JobID.
Email reports can be found in rcavlab@gmail.com.

Contig_taxonomy_and_abundance subfolder contains the phylodist-based contig taxonomy
output, along with species abundance estimation file.
The outputs are plain text files (.txt) generated through a python script.
```

Protein_taxonomy_and_function subfolder contains the compressesed DAA alignment file (.daa.bz2) from Diamond and the RMA taxonomy file from MEGAN6.
It also has a modified protein sequence file with product names added to protein headers. This file is used as an input for Diamond alignment.
The RMA file also has COG (based on eggNOG database) and GO terms (based on InterPro database) information.
The MEGAN COG comes from the eggNOG database and might differ from that produced in the COG section of the pipeline, which is based on IMG COG annotations.

COG_KEGG_functions subfolder contains COG and KEGG outputs.
COG.csv is the summary of the COG categories.
The "by coverage" column is weighted by average fold proteins and "by count" is simply the fraction of counts of each COG category.
KEGG.csv is the KEGG pathways summary and its columns are the same as those for COG. It aggregates markers into pathways using specific formulae used in the pipeline.

This is version 4 of the Cavlab pipeline and uses:
append_name2proteins.py
    python/v3.8.2
phylodist-to-contigSpeciesAbn.py
    python/v3.8.2
diamond/0.9.31
    nr_Jul2019 database prepared on July 11, 2019
megan/6.15.1
    java/8u121
COG_categorisation.py
    python/v3.8.2
KEGG_pathways.py
    python/v3.8.2

The pipeline was initiated by James Bevington on behalf of the Cavicchioli lab from BABS, UNSW, and he worked on the pipeline uptil v1.2.
Michelle Allen tried resolving technical issues in PyloSift runs; the software runs were stalled in v1.3b.1 and removed in v3.0.
Pratibha Panwar continued working on the pipeline and prepared the lates v4.1 as part of her thesis.

```
For further information or clarification contact Pratibha Panwar
(p.panwar@student.unsw.edu.au).
"""%(str(now.day), str(now.month), str(now.year))

with open(head_dir + 'Readme.txt', 'w') as readme_file:
    readme_file.write(readme_text)

#### Submit jobs as part of the Cavlab pipeline
jobprep_py = """import subprocess
import csv

command = 'qsub ' + '%s' + 'ContigTaxa_and_Abn.pbs'
screen = subprocess.check_output(command, shell = True)
screen = screen.decode()[0:6]
with open('%sjob_log.txt', 'a') as job_log:
    job_csv = csv.writer(job_log, delimiter = '\t')
    job_csv.writerow(['Phylodist to Contig taxonomy & abundance', screen])
print('Contig taxonomy and abundance estimation job submitted.')

command = 'qsub ' + '%s' + 'protein_function.pbs'
screen = subprocess.check_output(command, shell = True)
screen = screen.decode()[0:6]
with open('%sjob_log.txt', 'a') as job_log:
    job_csv = csv.writer(job_log, delimiter = '\t')
    job_csv.writerow(['Diamond and MEGAN', screen])
print('Protein function job submitted.')

command = 'qsub ' + '%s' + 'COG.pbs'
screen = subprocess.check_output(command, shell = True)
screen = screen.decode()[0:6]
with open('%sjob_log.txt', 'a') as job_log:
    job_csv = csv.writer(job_log, delimiter = '\t')
    job_csv.writerow(['COG', screen])
print('COG function job submitted.')

command = 'qsub ' + '%s' + 'KEGG.pbs'
screen = subprocess.check_output(command, shell = True)
```

```
screen = screen.decode()[0:6]
with open('%sjob_log.txt', 'a') as job_log:
    job_csv = csv.writer(job_log, delimiter = '\t')
    job_csv.writerow(['KEGG', screen])
print('KEGG function job submitted.')
"""%(res_dir, res_dir, res_dir, res_dir, res_dir, res_dir, res_dir, res_dir)

with open(res_dir + 'jobsubmission.py', 'w') as jobsub_script:
    jobsub_script.write(jobprep_py)

#### Pre-processing component ####
prtbin_py = """import csv
import Bio.SeqIO as SeqIO

#### Reading protein product name file
names = {}
with open('%s', 'r') as namef:
    namefc = csv.reader(namef, delimiter = '\t')
    for row in namefc:
        names[row[0]] = ('_').join(row[1].split(' '))

#### add protein product names to protein sequence headers
with open('%s', 'r') as prtfile:
    with open('%s.assembled_names.faa', 'w') as newfile:
        for record in SeqIO.parse(prtfile, "fasta"):
            if record.id in names.keys():
                record.id = record.id + '|' + names[record.id]
            else:
                record.id = record.id + '|Uncharacterized_predicted_protein'
            SeqIO.write(record, newfile, 'fasta')
"""%(PRODUCT_file, PROTEIN_file, protTaxa_dir + assembly_num)

with open(res_dir + 'append_name2proteins.py', 'w') as name2ORFs_script:
    name2ORFs_script.write(prtbin_py)
PROTEIN_file = protTaxa_dir + assembly_num + '.assembled_names.faa'

#### 1. Contig taxonomy and abundance estimation component ####
```

```
# Write python script
phylodist2contigTaxa_py = """import csv
from collections import Counter
import operator

def checkEqual(lst):
    return len(set(lst)) == 1 # if frequency of occurrence of all taxonomies is same, return True

scaflen, scafavgfold, allAbn, totcont = {}, {}, 0, 0
with open('%s', 'r') as covfile:
    covfilec = csv.reader(covfile,delimiter = '\t')
    next(covfilec) # skip header line
    for row in covfilec:
        scaflen[row[0]] = float(row[2]) # scaflen = {scaffoldID:length of scaffold}
        scafavgfold[row[0]] = float(row[1]) # scafavgfold = {scaffoldID: avg fold of scaffold}
        allAbn += float(row[1])*float(row[2])
        totcont += 1
print('Coverage file read.')

#### Preparing Contig ID to scaffold length and scaffold average fold dictionaries
contlen, contavgfold = {}, {}
with open('%s', 'r') as mapfile:
    mapfilec = csv.reader(mapfile,delimiter = '\t')
    for row in mapfilec:
        contlen[row[1]] = scaflen[row[0]] # contlen = {contigID:scaffold length}
        contavgfold[row[1]] = scafavgfold[row[0]]# contavgfold = {contigID: avg fold of
scaffold}
    contigIDlen = len(row[1]) # contig header lengths in a metagenome are constant, so length of
any header can be used
print('Contig to scaffold mapping file read.')

with open('%s', 'r') as prtname:
    prtnamec = csv.reader(prtname,delimiter = '\t')
    temp = {}
    for row in prtnamec:
```

```
        temp.setdefault(row[0][0:contigIDlen],[]).append(int(1)) # temp =
{contigID:[1,1,1,1,1,1...]}; the number of 1's in the list correspond to each gene identified on the
contig
    prtnum = {}
    for k, v in temp.items():
        prtnum[k] = sum(v) # prtnum = {contigID:number of proteins}
print('Protein names file read.')

#### Creating dictionary of contigs and the taxonomies of the proteins on them
contig, contigtaxonomy, contigidentity = [], {}, {}
with open('%s', 'r') as phylof:
    phylofc = csv.reader(phylof, delimiter = '\t')
    for row in phylofc:
        contig.append(row[0][0:contigIDlen]) # create a list of contigs with protein taxonomies in
phylodist file
        taxonomy = ';'.join(row[4].split(';')[0:7]) # exclude strain information
        contigtaxonomy.setdefault(row[0][0:contigIDlen],[]).append(taxonomy) # contigtaxonomy
= {contigID:taxonomy}
        #contigidentity.setdefault(row[0][0:contigIDlen],[]).append(row[3]+'---'+taxonomy)#
contigidentity = {contigID:identity---taxonomy}
    contprt = {}
    for k, v in contigtaxonomy.items():
        contprt[k] = len(v) # contprt = {contigID:number of proteins with taxonomy}
contig = sorted(list(set(contig))) # sorted list of contigs with at least one protein taxonomy
print('Phylodist file read.')

#### Writing contig taxonomy and metadata to output file
with open('%s' + '_contigtaxa.txt', 'w', newline = '') as contfile:
    contfilec = csv.writer(contfile, delimiter = '\t')
    contfilec.writerow(['Total metagenome abundance', allAbn])
    contfilec.writerow(['Total contigs in metagenome', totcont])
    contfilec.writerow(['ContigID', 'Average fold', 'Length', 'Taxonomy'])
    for i in range(len(contig)):
        if contig[i] in scafavgfold.keys(): # for coverage files with ContigIDs in place of
ScaffoldIDs
            avgfold = scafavgfold[contig[i]]
            length = scaflen[contig[i]]
```

```
        else:
            avgfold = contavgfold[contig[i]]
            length = contlen[contig[i]]
        taxacount = []
        taxa = sorted(Counter(contigtaxonomy[contig[i]]).items(),key = operator.itemgetter(1)) #
taxa = [[taxonomy1,frequency of occurrence],...,[taxonomyN,frequency of occurence]], where
taxonomy1 has lowest frequency and taxonomyN has highest frequency
        if contprt[contig[i]] >= 0.30 * prtnum[contig[i]]: # 30 percent prt taxa check criteria
            for j in range(len(taxa)):
                taxacount.append(taxa[j][1]) # taxacount = [list of frequencies of occurrence]
            if len(taxacount) > 1 and checkEqual(taxacount) == True:
                contfilec.writerow([contig[i], avgfold, length, 'Unclassified'])
            else:
                contfilec.writerow([contig[i], avgfold, length, taxa[-1][0]])
        else:
            contfilec.writerow([contig[i], avgfold, length, 'Unclassified'])
contfile.close()
print('Finished writing contig taxonomy file.')

#### Calculating species abundance using contig taxonomies
with open('%s' + '_speciesAbn.txt', 'w' , newline = '') as outf:
    outfc = csv.writer(outf, delimiter = '\t')
    totabn, contcount, addcount, species, contigspecies, allAbun, totcontigs = 0, 0, 0, [], {}, 0, 0
    with open('%s' + '_contigtaxa.txt','r') as metdat:
        metdatc = csv.reader(metdat, delimiter = '\t')
        allAbun = float(next(metdatc)[1])
        totcontigs = float(next(metdatc)[1])
        next(metdatc)
        for row in metdatc:
            addcount += 1 # count number of contigs with protein taxonomies
            totabn += float(row[1]) * float(row[2]) # calculate total abundance of assigned contigs,
including 'Unclassified' contigs
            # for unclassified contigs
            if row[3] == 'Unclassified':
                species.append('Unclassified contigs')
                contigspecies.setdefault('Unclassified contigs',[]).append(float(row[1])*float(row[2]))
# contigspecies = {species name:[list of abundances of species contigs]}
```

```
        # for contigs with only 'sp.' for species name; the genus name needs to be added
separately
        elif row[3].split(';')[6] == 'sp.':
            species.append(row[3].split(';')[5] + ' ' + row[3].split(';')[6])
            contcount += 1
            contigspecies.setdefault(row[3].split(';')[5] + ' ' +
row[3].split(';')[6],[]).append(float(row[1])*float(row[2]))
        # all other contigs with taxonomies
        else:
            species.append(row[3].split(';')[6])
            contcount += 1
            contigspecies.setdefault(row[3].split(';')[6],[]).append(float(row[1])*float(row[2]))
    species = list(set(species)) # removing duplicate species names

    for j in range(len(species)):
        speciescov = sum(contigspecies[species[j]]) # calculating species abundance
        outfc.writerow([species[j], speciescov])
    outfc.writerow(['Assigned contigs abundance', totabn]) # includes 'Unclassifed' contigs
    outfc.writerow(['Total metagenome abundance', allAbun])
    outfc.writerow(['Assigned contigs', round((contcount/totcontigs)*100,2)]) # percentage of
metagenome contigs with protein taxonomies
    outfc.writerow(['Unclassified contigs', round(((addcount-contcount)/totcontigs)*100,2)]) #
percentage of metagenome contigs that were 'Unclassified'
    outfc.writerow(['Unassigned contigs', round(100-((addcount/totcontigs)*100),2)]) #
percentage of metagenome contigs that had no protein taxonomies
print('Finished writing species abundance file.')
"""%(COV_file, MAP_file, PRODUCT_file, protTAXA_file, contTaxa_dir + assembly_num,
contTaxa_dir + assembly_num, contTaxa_dir + assembly_num)

with open(res_dir + 'phylodist-to-contigSpeciesAbn.py', 'w') as contTaxa_script:
    contTaxa_script.write(phylodist2contigTaxa_py)

# Write bash script
phylodist2contigTaxa_bash = """#!/bin/bash
#PBS -N Cavlab-ContigTaxonomy
#PBS -l select=1:ncpus=1:mem=96gb
#PBS -l walltime=12:00:00
```

```
#PBS -j oe
#PBS -o %sContigTaxonomy_report
#PBS -M rcavlab@gmail.com
#PBS -m ae

cd %s
module load python/3.8.2
python3 phylodist-to-contigSpeciesAbn.py
"""%(res_dir, res_dir)

with open(res_dir + 'ContigTaxa_and_Abn.pbs', 'w') as contbin_script:
    contbin_script.write(phylodist2contigTaxa_bash)

#### 2. DIAMOND and MEGAN6 protein taxonomy component ####
prtbin_bash = """#!/bin/bash
#PBS -N Cavlab-DIAMOND_MEGAN
#PBS -l select=1:ncpus=16:mem=120gb
#PBS -l walltime=48:00:00
#PBS -j oe
#PBS -o %sFunctionalBinning_report
#PBS -M rcavlab@gmail.com
#PBS -m ae

cd %s
module load diamond/0.9.31
diamond blastp --more-sensitive -d
/srv/scratch/jgi/Cavlab_pipeline_resources/v4/DiamondDB/nr_Jul2019 -q %s -o %s.daa -f 100 -
-algo 0 --index-mode 1 -p 16 -v

module load java/8u121
module load megan/6.15.1
export _JAVA_OPTIONS="-Xmx96g"
daa2rma -i %s.daa -o %s_prtFunction.rma -a2t
/srv/scratch/jgi/Cavlab_pipeline_resources/v4/prot_acc2tax-Jul2019X1.abin -a2eggnog
/srv/scratch/jgi/Cavlab_pipeline_resources/v4/acc2eggnog-Jul2019X.abin -a2interpro2go
/srv/scratch/jgi/Cavlab_pipeline_resources/v4/acc2interpro-Jul2019X.abin -v
```

```
bzip2 %s.daa
"""%(res_dir, protTaxa_dir, PROTEIN_file, assembly_num, assembly_num, assembly_num,
assembly_num)

with open(res_dir + 'protein_function.pbs', 'w') as prtbin_script:
    prtbin_script.write(prtbin_bash)

#### 3. COG functional potential analysis component ####
# Write python script
COG_py = """import csv
import Bio.SeqIO as SeqIO

#### Initialize COG category
A = [0]
B = [0]
C = [0]
D = [0]
E = [0]
F = [0]
G = [0]
H = [0]
I = [0]
J = [0]
K = [0]
L = [0]
M = [0]
N = [0]
O = [0]
P = [0]
Q = [0]
R = [0]
S = [0]
T = [0]
U = [0]
V = [0]
W = [0]
X = [0]
```

```
Y = [0]
Z = [0]
other = [0]
no_cov = []
print('COG categories initialised.')

#### Build coverage to protein map
coverage = {}
with open('%s', ‘r’) as covf:
    covfc = csv.reader(covf, delimiter = ‘\t’)
    next(covfc)
    for row in covfc:
        coverage[row[0]] = float(row[1])
print('Coverage file read.')

maps = {}
with open('%s',’r’) as mapf:
    mapfc = csv.reader(mapf, delimiter = ‘\t’)
    for row in mapfc:
        maps[row[0]] = row[1]
mapk = list(maps.keys())
print('Contig to scaffold mapping file read.')

covmap = {}
for i in range(len(mapk)):
    covmap[maps[mapk[i]]] = coverage[mapk[i]]
contname_len = len(list(covmap.keys())[0])
print('Contig to coverage mapping complete.')

prtcov = {}
with open('%s', ‘r’) as prtf:
    for record in SeqIO.parse(prtf, “fasta”):
        prtname = record.id.split('|')[0]
        if prtname[0:contname_len] in coverage.keys():
            prtcov[prtname] = coverage[prtname[0:contname_len]]
        else:
            prtcov[prtname] = covmap[prtname[0:contname_len]]
```

```
print('Protein to coverage mapping complete.')

foldCount_total = sum(prtcov.values())
directCount_total = len(prtcov)

#### Reading COG conversion file
reader =
csv.reader(open('/srv/scratch/jgi/Cavlab_pipeline_resources/v4/COG_conversion_v2.csv', ‘r’))
d = {}
for row in reader:
  k, v = row
  d[k] = v
print('Conversion dictionary prepared.')

#### Reading COG numbers from COG file
cogs = []
with open('%s', ‘r’) as cogf:
   cogfc = csv.reader(cogf, delimiter = ‘\t’)
   for row in cogfc:
      err = 1
      if row[0] in prtcov.keys():
         cogs.append([row[1], prtcov[row[0]]])
         err = 0
      if err == 1:
         no_cov.append(row[1])
print('COG file read.')

for j in range(len(cogs)):
   COGCat = d[cogs[j][0]]
   if COGCat == 'A':
      A.append(cogs[j][1])
   elif COGCat == 'B':
      B.append(cogs[j][1])
   elif COGCat == 'C':
      C.append(cogs[j][1])
   elif COGCat == 'D':
      D.append(cogs[j][1])
```

```
elif COGCat == 'E':
    E.append(cogs[j][1])
elif COGCat == 'F':
    F.append(cogs[j][1])
elif COGCat == 'G':
    G.append(cogs[j][1])
elif COGCat == 'H':
    H.append(cogs[j][1])
elif COGCat == 'I':
    I.append(cogs[j][1])
elif COGCat == 'J':
    J.append(cogs[j][1])
elif COGCat == 'K':
    K.append(cogs[j][1])
elif COGCat == 'L':
    L.append(cogs[j][1])
elif COGCat == 'M':
    M.append(cogs[j][1])
elif COGCat == 'N':
    N.append(cogs[j][1])
elif COGCat == 'O':
    O.append(cogs[j][1])
elif COGCat == 'P':
    P.append(cogs[j][1])
elif COGCat == 'Q':
    Q.append(cogs[j][1])
elif COGCat == ‘r’:
    R.append(cogs[j][1])
elif COGCat == 'S':
    S.append(cogs[j][1])
elif COGCat == 'T':
    T.append(cogs[j][1])
elif COGCat == 'U':
    U.append(cogs[j][1])
elif COGCat == 'V':
    V.append(cogs[j][1])
elif COGCat == ‘w’:
```

```
        W.append(cogs[j][1])
    elif COGCat == 'X':
        X.append(cogs[j][1])
    elif COGCat == 'Y':
        Y.append(cogs[j][1])
    elif COGCat == 'Z':
        Z.append(cogs[j][1])
    else:
        other.append(cogs[j][1])
print('COG numbers grouped under respective COG categories.')

#### Normalizing by coverage
Ac = sum(A)
Bc = sum(B)
Cc = sum(C)
Dc = sum(D)
Ec = sum(E)
Fc = sum(F)
Gc = sum(G)
Hc = sum(H)
Ic = sum(I)
Jc = sum(J)
Kc = sum(K)
Lc = sum(L)
Mc = sum(M)
Nc = sum(N)
Oc = sum(O)
Pc = sum(P)
Qc = sum(Q)
Rc = sum(R)
Sc = sum(S)
Tc = sum(T)
Uc = sum(U)
Vc = sum(V)
Wc = sum(W)
Xc = sum(X)
Yc = sum(Y)
```

```
Zc = sum(Z)
otherc = sum(other)
print('COG category coverages calculated.')

#### Normalizing by count
An = len(A)-1
Bn = len(B)-1
Cn = len(C)-1
Dn = len(D)-1
En = len(E)-1
Fn = len(F)-1
Gn = len(G)-1
Hn = len(H)-1
In = len(I)-1
Jn = len(J)-1
Kn = len(K)-1
Ln = len(L)-1
Mn = len(M)-1
Nn = len(N)-1
On = len(O)-1
Pn = len(P)-1
Qn = len(Q)-1
Rn = len(R)-1
Sn = len(S)-1
Tn = len(T)-1
Un = len(U)-1
Vn = len(V)-1
Wn = len(W)-1
Xn = len(X)-1
Yn = len(Y)-1
Zn = len(Z)-1
othern = len(other)-1
print('COG category counts calculated.')

#### Writing data to files
results_c = [Ac, Bc, Cc, Dc, Ec, Fc, Gc, Hc, Ic, Jc, Kc, Lc, Mc, Nc, Oc, Pc, Qc, Rc, Sc, Tc, Uc,
Vc, Wc, Xc, Yc, Zc, otherc]
```

```
results_n = [An, Bn, Cn, Dn, En, Fn, Gn, Hn, In, Jn, Kn, Ln, Mn, Nn, On, Pn, Qn, Rn, Sn, Tn,
Un, Vn, Wn, Xn, Yn, Zn, othern]
header = ['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'r', 'S', 'T', 'U', 'V',
'w', 'X', 'Y', 'Z', 'Other categories', 'Issues', 'Total ORF coverage']

with open('%s' + '_COG.txt', 'w', newline = '') as outfile:
    outcsv = csv.writer(outfile, delimiter = '\t')
    outcsv.writerow(['COG category', 'By coverage', 'By count'])
    for i in range(len(header)-2):
        outcsv.writerow([header[i], results_c[i], results_n[i]])
    outcsv.writerow([header[-2], len(no_cov), len(no_cov)])
    outcsv.writerow([header[-1], foldCount_total, directCount_total])
print('Finished writing COG output to file.')
"""%(COV_file, MAP_file, PROTEIN_file, COG_file, cogkegg_dir + assembly_num)

with open(res_dir + 'COG_categorisation.py','w') as COG_script:
    COG_script.write(COG_py)

# Write bash script
COG_bash = """#!/bin/bash
#PBS -N Cavlab-COG
#PBS -l select=1:ncpus=1:mem=64gb
#PBS -l walltime=12:00:00
#PBS -j oe
#PBS -o %sCOG_report
#PBS -M rcavlab@gmail.com
#PBS -m ae

cd %s
module load python/3.8.2
python3 COG_categorisation.py
"""%(res_dir, res_dir)

with open(res_dir + 'COG.pbs', 'w') as COG_pbs:
    COG_pbs.write(COG_bash)

#### 4. KEGG functional potential analysis component ####
```

```
KEGG_py = """import csv
import Bio.SeqIO as SeqIO
import numpy as np
from Bio import pairwise2

#### Initializing KEGG number variables
K00437 = [0]
K00436 = [0]
K18332 = [0]
K17997 = [0]
K00532 = [0]
K00533 = [0]
K05922 = [0]
K18016 = [0]
K14068 = [0]
K00440 = [0]
K13942 = [0]
K14126 = [0]
K01915 = [0]
K00264 = [0]
K00265 = [0]
K00266 = [0]
K00284 = [0]
K00864 = [0]
K00005 = [0]
K00169 = [0]
K00170 = [0]
K00456 = [0]
K01011 = [0]
K00860 = [0]
K00956 = [0]
K00957 = [0]
K00016 = [0]
K20932 = [0]
K20933 = [0]
K20934 = [0]
K20935 = [0]
```

K00174 = [0]
K00175 = [0]
K00360 = [0]
K00367 = [0]
K00244 = [0]
K03385 = [0]
K17877 = [0]
K00366 = [0]
K02305 = [0]
K04561 = [0]
K00376 = [0]
K02586 = [0]
K02591 = [0]
K10535 = [0]
K10944a = []
K10944m = []
K10945a = []
K10945m = []
K10946a = []
K10946m = []
K01602 = [0]
K00855 = [0]
K15230 = [0]
K15231 = [0]
K15234 = [0]
K15233 = [0]
K15232 = [0]
K00197 = [0]
K00194 = [0]
K03518 = [0]
K03519 = [0]
K03520 = [0]
K02256 = [0]
K02262 = [0]
K02274 = [0]
K02276 = [0]
K00401 = [0]

K00400 = [0]

K16157 = [0]

K16158 = [0]

K16159 = [0]

K16161 = [0]

K00390 = [0]

K00392 = [0]

K00380 = [0]

K00381 = [0]

K00394r = []

K00394o = []

K00395r = []

K00395o = []

K11180r = []

K11180o = []

K11181r = []

K11181o = []

K17224 = [0]

K17227 = [0]

K17226 = [0]

K17222 = [0]

K17223 = [0]

K17225 = [0]

K03821 = [0]

K15342 = [0]

K09951 = [0]

K07012 = [0]

K07475 = [0]

K19088 = [0]

K19087 = [0]

K19117 = [0]

K19123 = [0]

K19046 = [0]

K19127 = [0]

K19128 = [0]

K19129 = [0]

K07016 = [0]

K19138 = [0]
K19141 = [0]
K09952 = [0]
K19137 = [0]
K07464 = [0]
K02703 = [0]
K02706 = [0]
K02705 = [0]
K02704 = [0]
K02707 = [0]
K02708 = [0]
K02689 = [0]
K02690 = [0]
K02691 = [0]
K02692 = [0]
K02693 = [0]
K02694 = [0]
K08928 = [0]
K08929 = [0]
K08940 = [0]
K08941 = [0]
K08942 = [0]
K08943 = [0]
K04643 = [0]
K04642 = [0]
K04641 = [0]
K04250 = [0]
K00909 = [0]
K01428 = [0]
K01429 = [0]
K01430 = [0]
K00111 = [0]
K00112 = [0]
K00113 = [0]
K00096 = [0]
K00518 = [0]
K04564 = [0]

K04565 = [0]
K16627 = [0]
K06164 = [0]
K05780 = [0]
K06165 = [0]
K06166 = [0]
K06163 = [0]
K08977 = [0]
K09836 = [0]
K15746 = [0]
K16953 = [0]
K17486 = [0]
K07306 = [0]
K17218 = [0]
K03553 = [0]
K00370 = [0]
K00368 = [0]
K10944_ORFname = []
K10945_ORFname = []
K10946_ORFname = []
K00394_ORFname = []
K00395_ORFname = []
K11180_ORFname = []
K11181_ORFname = []
K11959 = [0]
K11960 = [0]
K11961 = [0]
K11962 = [0]
K11963 = [0]
K02048 = [0]
K02046 = [0]
K02047 = [0]
K02045 = [0]
K15576 = [0]
K15577 = [0]
K15578 = [0]
K15579 = [0]

K11950 = [0]
K11951 = [0]
K11952 = [0]
K11953 = [0]
K15551 = [0]
K15552 = [0]
K10831 = [0]
K15553 = [0]
K15554 = [0]
K15555 = [0]
K11069 = [0]
K11070 = [0]
K11071 = [0]
K11072 = [0]
K11073 = [0]
K11074 = [0]
K11075 = [0]
K11076 = [0]
K02040 = [0]
K02037 = [0]
K02038 = [0]
K02036 = [0]
K02044 = [0]
K02042 = [0]
K02041 = [0]
K11081 = [0]
K11082 = [0]
K11083 = [0]
K11084 = [0]
K02002 = [0]
K02001 = [0]
K02000 = [0]
K05845 = [0]
K05846 = [0]
K05847 = [0]
K10108 = [0]
K10109 = [0]

K10110 = [0]
K15770 = [0]
K15771 = [0]
K15772 = [0]
K10117 = [0]
K10118 = [0]
K10119 = [0]
K10232 = [0]
K10233 = [0]
K10234 = [0]
K10235 = [0]
K10196 = [0]
K10197 = [0]
K10198 = [0]
K10199 = [0]
K17315 = [0]
K17316 = [0]
K17317 = [0]
K10236 = [0]
K10237 = [0]
K10238 = [0]
K17311 = [0]
K17312 = [0]
K17313 = [0]
K17314 = [0]
K10200 = [0]
K10201 = [0]
K10202 = [0]
K10240 = [0]
K10241 = [0]
K10242 = [0]
K17329 = [0]
K17330 = [0]
K17331 = [0]
K17244 = [0]
K17245 = [0]
K17246 = [0]

K10537 = [0]
K10538 = [0]
K10539 = [0]
K10188 = [0]
K10189 = [0]
K10190 = [0]
K10191 = [0]
K10543 = [0]
K10544 = [0]
K10545 = [0]
K17326 = [0]
K17327 = [0]
K17328 = [0]
K10546 = [0]
K10547 = [0]
K10548 = [0]
K10552 = [0]
K10553 = [0]
K10554 = [0]
K10559 = [0]
K10560 = [0]
K10561 = [0]
K10562 = [0]
K10439 = [0]
K10440 = [0]
K10441 = [0]
K17202 = [0]
K17203 = [0]
K17204 = [0]
K10120 = [0]
K10121 = [0]
K10122 = [0]
K17321 = [0]
K17322 = [0]
K17323 = [0]
K17324 = [0]
K17325 = [0]

K02027 = [0]

K02025 = [0]

K02026 = [0]

K02058 = [0]

K02057 = [0]

K02056 = [0]

K10013 = [0]

K10015 = [0]

K10016 = [0]

K10017 = [0]

K10014 = [0]

K10036 = [0]

K10037 = [0]

K10038 = [0]

K09996 = [0]

K09997 = [0]

K09998 = [0]

K09999 = [0]

K10000 = [0]

K10001 = [0]

K10002 = [0]

K10003 = [0]

K10004 = [0]

K10039 = [0]

K10040 = [0]

K10041 = [0]

K10018 = [0]

K10019 = [0]

K10020 = [0]

K10021 = [0]

K09969 = [0]

K09970 = [0]

K09971 = [0]

K09972 = [0]

K10005 = [0]

K10006 = [0]

K10007 = [0]

K10008 = [0]
K02424 = [0]
K10009 = [0]
K10010 = [0]
K16956 = [0]
K16957 = [0]
K16958 = [0]
K16959 = [0]
K16960 = [0]
K10022 = [0]
K10023 = [0]
K10024 = [0]
K10025 = [0]
K23059 = [0]
K17077 = [0]
K23060 = [0]
K01999 = [0]
K01997 = [0]
K01998 = [0]
K01995 = [0]
K01996 = [0]
K11954 = [0]
K11955 = [0]
K11956 = [0]
K11957 = [0]
K11958 = [0]
K02073 = [0]
K02072 = [0]
K02071 = [0]
K15580 = [0]
K15581 = [0]
K15582 = [0]
K15583 = [0]
K10823 = [0]
K12368 = [0]
K12369 = [0]
K12370 = [0]

K12371 = [0]
K12372 = [0]
K16199 = [0]
K16200 = [0]
K16201 = [0]
K16202 = [0]
K01216 = [0]
K01199 = [0]
K19891 = [0]
K19892 = [0]
K19893 = [0]
K01190 = [0]
K12111 = [0]
K12308 = [0]
K12309 = [0]
K01188 = [0]
K05349 = [0]
K05350 = [0]
K01198 = [0]
K15920 = [0]
K22268 = [0]
K01179 = [0]
K19357 = [0]
K20542 = [0]
K01180 = [0]
K20846 = [0]
K20850 = [0]
K01219 = [0]
K20851 = [0]
K01200 = [0]
K21575 = [0]
K01177 = [0]
K01208 = [0]
K05992 = [0]
K22253 = [0]
K01178 = [0]
K12047 = [0]

K21574 = [0]
K07024 = [0]
K01193 = [0]
K00064 = [0]
K17993 = [0]
K02567 = [0]
K03778 = [0]
K00955 = [0]
K05907 = [0]
K17229 = [0]
K00958r = []
K00958o = []
K00958_ORFname = []
K01225 = [0]
K19668 = [0]
K08688 = [0]
K00301 = [0]
K00302 = [0]
K00303 = [0]
K00304 = [0]
K00305 = [0]
K03851 = [0]
K03852 = [0]
K01130 = [0]
K15923 = [0]
K00879 = [0]
K01628 = [0]
K00848 = [0]
K01629 = [0]
K01183 = [0]
K13381 = [0]
K14083 = [0]
K16178 = [0]
K16176 = [0]
K00702 = [0]
K16149 = [0]
K00975 = [0]

```
K00703 = [0]
K16146 = [0]
K16147 = [0]
K01176 = [0]
K05973 = [0]
K03430 = [0]
K05306 = [0]
K11472 = [0]
K01941 = [0]
other = [0]
no_cov = []
print('KEGG number variables initialized.')

#### Build coverage to protein map
coverage = {}
with open('%s', ‘r’) as covf:
    covfc = csv.reader(covf, delimiter = ‘\t’)
    next(covfc)
    for row in covfc:
        coverage[row[0]] = float(row[1])
print('Coverage file read.')

maps = {}
with open('%s',’r’) as mapf:
    mapfc = csv.reader(mapf, delimiter = ‘\t’)
    for row in mapfc:
        maps[row[0]] = row[1]
mapk = list(maps.keys())
print('Contig to scaffold mapping file read.')

covmap = {}
covmap = {}
for i in range(len(mapk)):
    covmap[maps[mapk[i]]] = coverage[mapk[i]]
contname_len = len(list(covmap.keys())[0])
print('Contig to coverage mapping complete.')
```

```
prtcov = {}
with open('%s', ‘r’) as prtf:
    for record in SeqIO.parse(prtf, “fasta”):
        prtname = record.id.split('|')[0]
        if prtname[0:contname_len] in coverage.keys():
            prtcov[prtname] = coverage[prtname[0:contname_len]]
        else:
            prtcov[prtname] = covmap[prtname[0:contname_len]]
print('Protein to coverage mapping complete.')

foldCount_total = sum(prtcov.values())
directCount_total = len(prtcov)

#### extracting KEGG
kegg = []
with open('%s', ‘r’) as keggf:
    keggfc = csv.reader(keggf, delimiter = ‘\t’)
    for row in keggfc:
        err = 1
        if row[0] in prtcov.keys():
            kegg.append([row[2][3:],prtcov[row[0]],row[0]])
            err=0
        if err==1:
            no_cov.append(row[2][3:])
print('KEGG file read.')

for j in range(len(kegg)):
    ##sulfur-assimilatory and dissimilatory
    if kegg[j][0] == 'K00394':
        K00394_ORFname.append(kegg[j][2])
    elif kegg[j][0] == 'K00395':
        K00395_ORFname.append(kegg[j][2])
    elif kegg[j][0] == 'K11180':
        K11180_ORFname.append(kegg[j][2])
    elif kegg[j][0] == 'K11181':
        K11181_ORFname.append(kegg[j][2])
    elif kegg[j][0] == 'K00958':
```

```
        K00958_ORFname.append(kegg[j][2])
    ##ammonia/methane monooxygenase
    elif kegg[j][0] == 'K10944':
        K10944_ORFname.append(kegg[j][2])
    elif kegg[j][0] == 'K10945':
        K10945_ORFname.append(kegg[j][2])
    elif kegg[j][0] == 'K10946':
        K10946_ORFname.append(kegg[j][2])
    ##others
    elif kegg[j][0] == 'K00437':
        K00437.append(kegg[j][1])
    elif kegg[j][0] == 'K00436':
        K00436.append(kegg[j][1])
    elif kegg[j][0] == 'K18332':
        K18332.append(kegg[j][1])
    elif kegg[j][0] == 'K17997':
        K17997.append(kegg[j][1])
    elif kegg[j][0] == 'K00532':
        K00532.append(kegg[j][1])
    elif kegg[j][0] == 'K00533':
        K00533.append(kegg[j][1])
    elif kegg[j][0] == 'K05922':
        K05922.append(kegg[j][1])
    elif kegg[j][0] == 'K18016':
        K18016.append(kegg[j][1])
    elif kegg[j][0] == 'K14068':
        K14068.append(kegg[j][1])
    elif kegg[j][0] == 'K00440':
        K00440.append(kegg[j][1])
    elif kegg[j][0] == 'K13942':
        K13942.append(kegg[j][1])
    elif kegg[j][0] == 'K14126':
        K14126.append(kegg[j][1])
    elif kegg[j][0] == 'K01915':
        K01915.append(kegg[j][1])
    elif kegg[j][0] == 'K00264':
        K00264.append(kegg[j][1])
```

```
elif kegg[j][0] == 'K00265':
    K00265.append(kegg[j][1])
elif kegg[j][0] == 'K00266':
    K00266.append(kegg[j][1])
elif kegg[j][0] == 'K00284':
    K00284.append(kegg[j][1])
elif kegg[j][0] == 'K00864':
    K00864.append(kegg[j][1])
elif kegg[j][0] == 'K00005':
    K00005.append(kegg[j][1])
elif kegg[j][0] == 'K19117':
    K19117.append(kegg[j][1])
elif kegg[j][0] == 'K19128':
    K19128.append(kegg[j][1])
elif kegg[j][0] == 'K00169':
    K00169.append(kegg[j][1])
elif kegg[j][0] == 'K00170':
    K00170.append(kegg[j][1])
elif kegg[j][0] == 'K00016':
    K00016.append(kegg[j][1])
elif kegg[j][0] == 'K00174':
    K00174.append(kegg[j][1])
elif kegg[j][0] == 'K00175':
    K00175.append(kegg[j][1])
elif kegg[j][0] == 'K00244':
    K00244.append(kegg[j][1])
elif kegg[j][0] == 'K00194':
    K00194.append(kegg[j][1])
elif kegg[j][0] == 'K00197':
    K00197.append(kegg[j][1])
elif kegg[j][0] == 'K00360':
    K00360.append(kegg[j][1])
elif kegg[j][0] == 'K00367':
    K00367.append(kegg[j][1])
elif kegg[j][0] == 'K20932':
    K20932.append(kegg[j][1])
elif kegg[j][0] == 'K20933':
```

```
        K20933.append(kegg[j][1])
    elif kegg[j][0] == 'K20934':
        K20934.append(kegg[j][1])
    elif kegg[j][0] == 'K20935':
        K20935.append(kegg[j][1])
    elif kegg[j][0] == 'K00456':
        K00456.append(kegg[j][1])
    elif kegg[j][0] == 'K01011':
        K01011.append(kegg[j][1])
    elif kegg[j][0] == 'K00860':
        K00860.append(kegg[j][1])
    elif kegg[j][0] == 'K00956':
        K00956.append(kegg[j][1])
    elif kegg[j][0] == 'K00957':
        K00957.append(kegg[j][1])
    elif kegg[j][0] == 'K19087':
        K19087.append(kegg[j][1])
    elif kegg[j][0] == 'K19046':
        K19046.append(kegg[j][1])
    elif kegg[j][0] == 'K19127':
        K19127.append(kegg[j][1])
    elif kegg[j][0] == 'K19129':
        K19129.append(kegg[j][1])
    elif kegg[j][0] == 'K03385':
        K03385.append(kegg[j][1])
    elif kegg[j][0] == 'K17877':
        K17877.append(kegg[j][1])
    elif kegg[j][0] == 'K00366':
        K00366.append(kegg[j][1])
    elif kegg[j][0] == 'K02305':
        K02305.append(kegg[j][1])
    elif kegg[j][0] == 'K04561':
        K04561.append(kegg[j][1])
    elif kegg[j][0] == 'K00376':
        K00376.append(kegg[j][1])
    elif kegg[j][0] == 'K02586':
        K02586.append(kegg[j][1])
```

```
elif kegg[j][0] == 'K02591':
    K02591.append(kegg[j][1])
elif kegg[j][0] == 'K10535':
    K10535.append(kegg[j][1])
elif kegg[j][0] == 'K01602':
    K01602.append(kegg[j][1])
elif kegg[j][0] == 'K00855':
    K00855.append(kegg[j][1])
elif kegg[j][0] == 'K15230':
    K15230.append(kegg[j][1])
elif kegg[j][0] == 'K15231':
    K15231.append(kegg[j][1])
elif kegg[j][0] == 'K15234':
    K15234.append(kegg[j][1])
elif kegg[j][0] == 'K15233':
    K15233.append(kegg[j][1])
elif kegg[j][0] == 'K15232':
    K15232.append(kegg[j][1])
elif kegg[j][0] == 'K03518':
    K03518.append(kegg[j][1])
elif kegg[j][0] == 'K03519':
    K03519.append(kegg[j][1])
elif kegg[j][0] == 'K03520':
    K03520.append(kegg[j][1])
elif kegg[j][0] == 'K02256':
    K02256.append(kegg[j][1])
elif kegg[j][0] == 'K02262':
    K02262.append(kegg[j][1])
elif kegg[j][0] == 'K02274':
    K02274.append(kegg[j][1])
elif kegg[j][0] == 'K02276':
    K02276.append(kegg[j][1])
elif kegg[j][0] == 'K00401':
    K00401.append(kegg[j][1])
elif kegg[j][0] == 'K00400':
    K00400.append(kegg[j][1])
elif kegg[j][0] == 'K16157':
```

```
        K16157.append(kegg[j][1])
    elif kegg[j][0] == 'K16158':
        K16158.append(kegg[j][1])
    elif kegg[j][0] == 'K16159':
        K16159.append(kegg[j][1])
    elif kegg[j][0] == 'K16161':
        K16161.append(kegg[j][1])
    elif kegg[j][0] == 'K00390':
        K00390.append(kegg[j][1])
    elif kegg[j][0] == 'K00392':
        K00392.append(kegg[j][1])
    elif kegg[j][0] == 'K00380':
        K00380.append(kegg[j][1])
    elif kegg[j][0] == 'K00381':
        K00381.append(kegg[j][1])
    elif kegg[j][0] == 'K17224':
        K17224.append(kegg[j][1])
    elif kegg[j][0] == 'K17227':
        K17227.append(kegg[j][1])
    elif kegg[j][0] == 'K17226':
        K17226.append(kegg[j][1])
    elif kegg[j][0] == 'K17222':
        K17222.append(kegg[j][1])
    elif kegg[j][0] == 'K17223':
        K17223.append(kegg[j][1])
    elif kegg[j][0] == 'K17225':
        K17225.append(kegg[j][1])
    elif kegg[j][0] == 'K03821':
        K03821.append(kegg[j][1])
    elif kegg[j][0] == 'K15342':
        K15342.append(kegg[j][1])
    elif kegg[j][0] == 'K09951':
        K09951.append(kegg[j][1])
    elif kegg[j][0] == 'K07012':
        K07012.append(kegg[j][1])
    elif kegg[j][0] == 'K07475':
        K07475.append(kegg[j][1])
```

```
elif kegg[j][0] == 'K19088':
    K19088.append(kegg[j][1])
elif kegg[j][0] == 'K19123':
    K19123.append(kegg[j][1])
elif kegg[j][0] == 'K19127':
    K19127.append(kegg[j][1])
elif kegg[j][0] == 'K07016':
    K07016.append(kegg[j][1])
elif kegg[j][0] == 'K19138':
    K19138.append(kegg[j][1])
elif kegg[j][0] == 'K19141':
    K19141.append(kegg[j][1])
elif kegg[j][0] == 'K09952':
    K09952.append(kegg[j][1])
elif kegg[j][0] == 'K19137':
    K19137.append(kegg[j][1])
elif kegg[j][0] == 'K07464':
    K07464.append(kegg[j][1])
elif kegg[j][0] == 'K02703':
    K02703.append(kegg[j][1])
elif kegg[j][0] == 'K02706':
    K02706.append(kegg[j][1])
elif kegg[j][0] == 'K02705':
    K02705.append(kegg[j][1])
elif kegg[j][0] == 'K02704':
    K02704.append(kegg[j][1])
elif kegg[j][0] == 'K02707':
    K02707.append(kegg[j][1])
elif kegg[j][0] == 'K02708':
    K02708.append(kegg[j][1])
elif kegg[j][0] == 'K02689':
    K02689.append(kegg[j][1])
elif kegg[j][0] == 'K02690':
    K02690.append(kegg[j][1])
elif kegg[j][0] == 'K02691':
    K02691.append(kegg[j][1])
elif kegg[j][0] == 'K02692':
```

```
        K02692.append(kegg[j][1])
    elif kegg[j][0] == 'K02693':
        K02693.append(kegg[j][1])
    elif kegg[j][0] == 'K02694':
        K02694.append(kegg[j][1])
    elif kegg[j][0] == 'K08928':
        K08928.append(kegg[j][1])
    elif kegg[j][0] == 'K08929':
        K08929.append(kegg[j][1])
    elif kegg[j][0] == 'K08940':
        K08940.append(kegg[j][1])
    elif kegg[j][0] == 'K08941':
        K08941.append(kegg[j][1])
    elif kegg[j][0] == 'K08942':
        K08942.append(kegg[j][1])
    elif kegg[j][0] == 'K08943':
        K08943.append(kegg[j][1])
    elif kegg[j][0] == 'K04643':
        K04643.append(kegg[j][1])
    elif kegg[j][0] == 'K04642':
        K04642.append(kegg[j][1])
    elif kegg[j][0] == 'K04641':
        K04641.append(kegg[j][1])
    elif kegg[j][0] == 'K04250':
        K04250.append(kegg[j][1])
    elif kegg[j][0] == 'K00909':
        K00909.append(kegg[j][1])
    elif kegg[j][0] == 'K01428':
        K01428.append(kegg[j][1])
    elif kegg[j][0] == 'K01429':
        K01429.append(kegg[j][1])
    elif kegg[j][0] == 'K00111':
        K00111.append(kegg[j][1])
    elif kegg[j][0] == 'K00112':
        K00112.append(kegg[j][1])
    elif kegg[j][0] == 'K00113':
        K00113.append(kegg[j][1])
```

```
elif kegg[j][0] == 'K00096':
    K00096.append(kegg[j][1])
elif kegg[j][0] == 'K00518':
    K00518.append(kegg[j][1])
elif kegg[j][0] == 'K04564':
    K04564.append(kegg[j][1])
elif kegg[j][0] == 'K04565':
    K04565.append(kegg[j][1])
elif kegg[j][0] == 'K16627':
    K16627.append(kegg[j][1])
elif kegg[j][0] == 'K06164':
    K06164.append(kegg[j][1])
elif kegg[j][0] == 'K05780':
    K05780.append(kegg[j][1])
elif kegg[j][0] == 'K06165':
    K06165.append(kegg[j][1])
elif kegg[j][0] == 'K06166':
    K06166.append(kegg[j][1])
elif kegg[j][0] == 'K06163':
    K06163.append(kegg[j][1])
elif kegg[j][0] == 'K08977':
    K08977.append(kegg[j][1])
elif kegg[j][0] == 'K09836':
    K09836.append(kegg[j][1])
elif kegg[j][0] == 'K15746':
    K15746.append(kegg[j][1])
elif kegg[j][0] == 'K16953':
    K16953.append(kegg[j][1])
elif kegg[j][0] == 'K17486':
    K17486.append(kegg[j][1])
elif kegg[j][0] == 'K07306':
    K07306.append(kegg[j][1])
elif kegg[j][0] == 'K17218':
    K17218.append(kegg[j][1])
elif kegg[j][0] == 'K03553':
    K03553.append(kegg[j][1])
elif kegg[j][0] == 'K00370':
```

```
        K00370.append(kegg[j][1])
    elif kegg[j][0] == 'K00368':
        K00368.append(kegg[j][1])
    elif kegg[j][0] == 'K11959':
        K11959.append(kegg[j][1])
    elif kegg[j][0] == 'K11960':
        K11960.append(kegg[j][1])
    elif kegg[j][0] == 'K11961':
        K11961.append(kegg[j][1])
    elif kegg[j][0] == 'K11962':
        K11962.append(kegg[j][1])
    elif kegg[j][0] == 'K11963':
        K11963.append(kegg[j][1])
    elif kegg[j][0] == 'K02048':
        K02048.append(kegg[j][1])
    elif kegg[j][0] == 'K02046':
        K02046.append(kegg[j][1])
    elif kegg[j][0] == 'K02047':
        K02047.append(kegg[j][1])
    elif kegg[j][0] == 'K02045':
        K02045.append(kegg[j][1])
    elif kegg[j][0] == 'K15576':
        K15576.append(kegg[j][1])
    elif kegg[j][0] == 'K15577':
        K15577.append(kegg[j][1])
    elif kegg[j][0] == 'K15578':
        K15578.append(kegg[j][1])
    elif kegg[j][0] == 'K15579':
        K15579.append(kegg[j][1])
    elif kegg[j][0] == 'K11950':
        K11950.append(kegg[j][1])
    elif kegg[j][0] == 'K11951':
        K11951.append(kegg[j][1])
    elif kegg[j][0] == 'K11952':
        K11952.append(kegg[j][1])
    elif kegg[j][0] == 'K11953':
        K11953.append(kegg[j][1])
```

```
elif kegg[j][0] == 'K15551':
    K15551.append(kegg[j][1])
elif kegg[j][0] == 'K15552':
    K15552.append(kegg[j][1])
elif kegg[j][0] == 'K10831':
    K10831.append(kegg[j][1])
elif kegg[j][0] == 'K15553':
    K15553.append(kegg[j][1])
elif kegg[j][0] == 'K15554':
    K15554.append(kegg[j][1])
elif kegg[j][0] == 'K15555':
    K15555.append(kegg[j][1])
elif kegg[j][0] == 'K11069':
    K11069.append(kegg[j][1])
elif kegg[j][0] == 'K11070':
    K11070.append(kegg[j][1])
elif kegg[j][0] == 'K11071':
    K11071.append(kegg[j][1])
elif kegg[j][0] == 'K11072':
    K11072.append(kegg[j][1])
elif kegg[j][0] == 'K11073':
    K11073.append(kegg[j][1])
elif kegg[j][0] == 'K11074':
    K11074.append(kegg[j][1])
elif kegg[j][0] == 'K11075':
    K11075.append(kegg[j][1])
elif kegg[j][0] == 'K11076':
    K11076.append(kegg[j][1])
elif kegg[j][0] == 'K02040':
    K02040.append(kegg[j][1])
elif kegg[j][0] == 'K02037':
    K02037.append(kegg[j][1])
elif kegg[j][0] == 'K02038':
    K02038.append(kegg[j][1])
elif kegg[j][0] == 'K02036':
    K02036.append(kegg[j][1])
elif kegg[j][0] == 'K02044':
```

```
        K02044.append(kegg[j][1])
    elif kegg[j][0] == 'K02042':
        K02042.append(kegg[j][1])
    elif kegg[j][0] == 'K02041':
        K02041.append(kegg[j][1])
    elif kegg[j][0] == 'K11081':
        K11081.append(kegg[j][1])
    elif kegg[j][0] == 'K11082':
        K11082.append(kegg[j][1])
    elif kegg[j][0] == 'K11083':
        K11083.append(kegg[j][1])
    elif kegg[j][0] == 'K11084':
        K11084.append(kegg[j][1])
    elif kegg[j][0] == 'K02002':
        K02002.append(kegg[j][1])
    elif kegg[j][0] == 'K02001':
        K02001.append(kegg[j][1])
    elif kegg[j][0] == 'K02000':
        K02000.append(kegg[j][1])
    elif kegg[j][0] == 'K05845':
        K05845.append(kegg[j][1])
    elif kegg[j][0] == 'K05846':
        K05846.append(kegg[j][1])
    elif kegg[j][0] == 'K05847':
        K05847.append(kegg[j][1])
    elif kegg[j][0] == 'K10108':
        K10108.append(kegg[j][1])
    elif kegg[j][0] == 'K10109':
        K10109.append(kegg[j][1])
    elif kegg[j][0] == 'K10110':
        K10110.append(kegg[j][1])
    elif kegg[j][0] == 'K15770':
        K15770.append(kegg[j][1])
    elif kegg[j][0] == 'K15771':
        K15771.append(kegg[j][1])
    elif kegg[j][0] == 'K15772':
        K15772.append(kegg[j][1])
```

```
elif kegg[j][0] == 'K10117':
    K10117.append(kegg[j][1])
elif kegg[j][0] == 'K10118':
    K10118.append(kegg[j][1])
elif kegg[j][0] == 'K10119':
    K10119.append(kegg[j][1])
elif kegg[j][0] == 'K10232':
    K10232.append(kegg[j][1])
elif kegg[j][0] == 'K10233':
    K10233.append(kegg[j][1])
elif kegg[j][0] == 'K10234':
    K10234.append(kegg[j][1])
elif kegg[j][0] == 'K10235':
    K10235.append(kegg[j][1])
elif kegg[j][0] == 'K10196':
    K10196.append(kegg[j][1])
elif kegg[j][0] == 'K10197':
    K10197.append(kegg[j][1])
elif kegg[j][0] == 'K10198':
    K10198.append(kegg[j][1])
elif kegg[j][0] == 'K10199':
    K10199.append(kegg[j][1])
elif kegg[j][0] == 'K17315':
    K17315.append(kegg[j][1])
elif kegg[j][0] == 'K17316':
    K17316.append(kegg[j][1])
elif kegg[j][0] == 'K17317':
    K17317.append(kegg[j][1])
elif kegg[j][0] == 'K10236':
    K10236.append(kegg[j][1])
elif kegg[j][0] == 'K10237':
    K10237.append(kegg[j][1])
elif kegg[j][0] == 'K10238':
    K10238.append(kegg[j][1])
elif kegg[j][0] == 'K17311':
    K17311.append(kegg[j][1])
elif kegg[j][0] == 'K17312':
```

```
        K17312.append(kegg[j][1])
    elif kegg[j][0] == 'K17313':
        K17313.append(kegg[j][1])
    elif kegg[j][0] == 'K17314':
        K17314.append(kegg[j][1])
    elif kegg[j][0] == 'K10200':
        K10200.append(kegg[j][1])
    elif kegg[j][0] == 'K10201':
        K10201.append(kegg[j][1])
    elif kegg[j][0] == 'K10202':
        K10202.append(kegg[j][1])
    elif kegg[j][0] == 'K10240':
        K10240.append(kegg[j][1])
    elif kegg[j][0] == 'K10241':
        K10241.append(kegg[j][1])
    elif kegg[j][0] == 'K10242':
        K10242.append(kegg[j][1])
    elif kegg[j][0] == 'K17329':
        K17329.append(kegg[j][1])
    elif kegg[j][0] == 'K17330':
        K17330.append(kegg[j][1])
    elif kegg[j][0] == 'K17331':
        K17331.append(kegg[j][1])
    elif kegg[j][0] == 'K17244':
        K17244.append(kegg[j][1])
    elif kegg[j][0] == 'K17245':
        K17245.append(kegg[j][1])
    elif kegg[j][0] == 'K17246':
        K17246.append(kegg[j][1])
    elif kegg[j][0] == 'K10537':
        K10537.append(kegg[j][1])
    elif kegg[j][0] == 'K10538':
        K10538.append(kegg[j][1])
    elif kegg[j][0] == 'K10539':
        K10539.append(kegg[j][1])
    elif kegg[j][0] == 'K10188':
        K10188.append(kegg[j][1])
```

```
elif kegg[j][0] == 'K10189':
    K10189.append(kegg[j][1])
elif kegg[j][0] == 'K10190':
    K10190.append(kegg[j][1])
elif kegg[j][0] == 'K10191':
    K10191.append(kegg[j][1])
elif kegg[j][0] == 'K10543':
    K10543.append(kegg[j][1])
elif kegg[j][0] == 'K10544':
    K10544.append(kegg[j][1])
elif kegg[j][0] == 'K10545':
    K10545.append(kegg[j][1])
elif kegg[j][0] == 'K17326':
    K17326.append(kegg[j][1])
elif kegg[j][0] == 'K17327':
    K17327.append(kegg[j][1])
elif kegg[j][0] == 'K17328':
    K17328.append(kegg[j][1])
elif kegg[j][0] == 'K10546':
    K10546.append(kegg[j][1])
elif kegg[j][0] == 'K10547':
    K10547.append(kegg[j][1])
elif kegg[j][0] == 'K10548':
    K10548.append(kegg[j][1])
elif kegg[j][0] == 'K10552':
    K10552.append(kegg[j][1])
elif kegg[j][0] == 'K10553':
    K10553.append(kegg[j][1])
elif kegg[j][0] == 'K10554':
    K10554.append(kegg[j][1])
elif kegg[j][0] == 'K10559':
    K10559.append(kegg[j][1])
elif kegg[j][0] == 'K10560':
    K10560.append(kegg[j][1])
elif kegg[j][0] == 'K10561':
    K10561.append(kegg[j][1])
elif kegg[j][0] == 'K10562':
```

```
        K10562.append(kegg[j][1])
    elif kegg[j][0] == 'K10439':
        K10439.append(kegg[j][1])
    elif kegg[j][0] == 'K10440':
        K10440.append(kegg[j][1])
    elif kegg[j][0] == 'K10441':
        K10441.append(kegg[j][1])
    elif kegg[j][0] == 'K17202':
        K17202.append(kegg[j][1])
    elif kegg[j][0] == 'K17203':
        K17203.append(kegg[j][1])
    elif kegg[j][0] == 'K17204':
        K17204.append(kegg[j][1])
    elif kegg[j][0] == 'K10120':
        K10120.append(kegg[j][1])
    elif kegg[j][0] == 'K10121':
        K10121.append(kegg[j][1])
    elif kegg[j][0] == 'K10122':
        K10122.append(kegg[j][1])
    elif kegg[j][0] == 'K17321':
        K17321.append(kegg[j][1])
    elif kegg[j][0] == 'K17322':
        K17322.append(kegg[j][1])
    elif kegg[j][0] == 'K17323':
        K17323.append(kegg[j][1])
    elif kegg[j][0] == 'K17324':
        K17324.append(kegg[j][1])
    elif kegg[j][0] == 'K17325':
        K17325.append(kegg[j][1])
    elif kegg[j][0] == 'K02027':
        K02027.append(kegg[j][1])
    elif kegg[j][0] == 'K02025':
        K02025.append(kegg[j][1])
    elif kegg[j][0] == 'K02026':
        K02026.append(kegg[j][1])
    elif kegg[j][0] == 'K02058':
        K02058.append(kegg[j][1])
```

```
elif kegg[j][0] == 'K02057':
    K02057.append(kegg[j][1])
elif kegg[j][0] == 'K02056':
    K02056.append(kegg[j][1])
elif kegg[j][0] == 'K10013':
    K10013.append(kegg[j][1])
elif kegg[j][0] == 'K10015':
    K10015.append(kegg[j][1])
elif kegg[j][0] == 'K10016':
    K10016.append(kegg[j][1])
elif kegg[j][0] == 'K10017':
    K10017.append(kegg[j][1])
elif kegg[j][0] == 'K10014':
    K10014.append(kegg[j][1])
elif kegg[j][0] == 'K10036':
    K10036.append(kegg[j][1])
elif kegg[j][0] == 'K10037':
    K10037.append(kegg[j][1])
elif kegg[j][0] == 'K10038':
    K10038.append(kegg[j][1])
elif kegg[j][0] == 'K09996':
    K09996.append(kegg[j][1])
elif kegg[j][0] == 'K09997':
    K09997.append(kegg[j][1])
elif kegg[j][0] == 'K09998':
    K09998.append(kegg[j][1])
elif kegg[j][0] == 'K09999':
    K09999.append(kegg[j][1])
elif kegg[j][0] == 'K10000':
    K10000.append(kegg[j][1])
elif kegg[j][0] == 'K10001':
    K10001.append(kegg[j][1])
elif kegg[j][0] == 'K10002':
    K10002.append(kegg[j][1])
elif kegg[j][0] == 'K10003':
    K10003.append(kegg[j][1])
elif kegg[j][0] == 'K10004':
```

```
        K10004.append(kegg[j][1])
    elif kegg[j][0] == 'K10039':
        K10039.append(kegg[j][1])
    elif kegg[j][0] == 'K10040':
        K10040.append(kegg[j][1])
    elif kegg[j][0] == 'K10041':
        K10041.append(kegg[j][1])
    elif kegg[j][0] == 'K10018':
        K10018.append(kegg[j][1])
    elif kegg[j][0] == 'K10019':
        K10019.append(kegg[j][1])
    elif kegg[j][0] == 'K10020':
        K10020.append(kegg[j][1])
    elif kegg[j][0] == 'K10021':
        K10021.append(kegg[j][1])
    elif kegg[j][0] == 'K09969':
        K09969.append(kegg[j][1])
    elif kegg[j][0] == 'K09970':
        K09970.append(kegg[j][1])
    elif kegg[j][0] == 'K09971':
        K09971.append(kegg[j][1])
    elif kegg[j][0] == 'K09972':
        K09972.append(kegg[j][1])
    elif kegg[j][0] == 'K10005':
        K10005.append(kegg[j][1])
    elif kegg[j][0] == 'K10006':
        K10006.append(kegg[j][1])
    elif kegg[j][0] == 'K10007':
        K10007.append(kegg[j][1])
    elif kegg[j][0] == 'K10008':
        K10008.append(kegg[j][1])
    elif kegg[j][0] == 'K02424':
        K02424.append(kegg[j][1])
    elif kegg[j][0] == 'K10009':
        K10009.append(kegg[j][1])
    elif kegg[j][0] == 'K10010':
        K10010.append(kegg[j][1])
```

```
elif kegg[j][0] == 'K16956':
    K16956.append(kegg[j][1])
elif kegg[j][0] == 'K16957':
    K16957.append(kegg[j][1])
elif kegg[j][0] == 'K16958':
    K16958.append(kegg[j][1])
elif kegg[j][0] == 'K16959':
    K16959.append(kegg[j][1])
elif kegg[j][0] == 'K16960':
    K16960.append(kegg[j][1])
elif kegg[j][0] == 'K10022':
    K10022.append(kegg[j][1])
elif kegg[j][0] == 'K10023':
    K10023.append(kegg[j][1])
elif kegg[j][0] == 'K10024':
    K10024.append(kegg[j][1])
elif kegg[j][0] == 'K10025':
    K10025.append(kegg[j][1])
elif kegg[j][0] == 'K23059':
    K23059.append(kegg[j][1])
elif kegg[j][0] == 'K17077':
    K17077.append(kegg[j][1])
elif kegg[j][0] == 'K23060':
    K23060.append(kegg[j][1])
elif kegg[j][0] == 'K01999':
    K01999.append(kegg[j][1])
elif kegg[j][0] == 'K01997':
    K01997.append(kegg[j][1])
elif kegg[j][0] == 'K01998':
    K01998.append(kegg[j][1])
elif kegg[j][0] == 'K01995':
    K01995.append(kegg[j][1])
elif kegg[j][0] == 'K01996':
    K01996.append(kegg[j][1])
elif kegg[j][0] == 'K11954':
    K11954.append(kegg[j][1])
elif kegg[j][0] == 'K11955':
```

```
        K11955.append(kegg[j][1])
    elif kegg[j][0] == 'K11956':
        K11956.append(kegg[j][1])
    elif kegg[j][0] == 'K11957':
        K11957.append(kegg[j][1])
    elif kegg[j][0] == 'K11958':
        K11958.append(kegg[j][1])
    elif kegg[j][0] == 'K02073':
        K02073.append(kegg[j][1])
    elif kegg[j][0] == 'K02072':
        K02072.append(kegg[j][1])
    elif kegg[j][0] == 'K02071':
        K02071.append(kegg[j][1])
    elif kegg[j][0] == 'K15580':
        K15580.append(kegg[j][1])
    elif kegg[j][0] == 'K15581':
        K15581.append(kegg[j][1])
    elif kegg[j][0] == 'K15582':
        K15582.append(kegg[j][1])
    elif kegg[j][0] == 'K15583':
        K15583.append(kegg[j][1])
    elif kegg[j][0] == 'K10823':
        K10823.append(kegg[j][1])
    elif kegg[j][0] == 'K12368':
        K12368.append(kegg[j][1])
    elif kegg[j][0] == 'K12369':
        K12369.append(kegg[j][1])
    elif kegg[j][0] == 'K12370':
        K12370.append(kegg[j][1])
    elif kegg[j][0] == 'K12371':
        K12371.append(kegg[j][1])
    elif kegg[j][0] == 'K12372':
        K12372.append(kegg[j][1])
    elif kegg[j][0] == 'K16199':
        K16199.append(kegg[j][1])
    elif kegg[j][0] == 'K16200':
        K16200.append(kegg[j][1])
```

```
elif kegg[j][0] == 'K16201':
    K16201.append(kegg[j][1])
elif kegg[j][0] == 'K16202':
    K16202.append(kegg[j][1])
elif kegg[j][0] == 'K01216':
    K01216.append(kegg[j][1])
elif kegg[j][0] == 'K01199':
    K01199.append(kegg[j][1])
elif kegg[j][0] == 'K19891':
    K19891.append(kegg[j][1])
elif kegg[j][0] == 'K19892':
    K19892.append(kegg[j][1])
elif kegg[j][0] == 'K19893':
    K19893.append(kegg[j][1])
elif kegg[j][0] == 'K12111':
    K12111.append(kegg[j][1])
elif kegg[j][0] == 'K12308':
    K12308.append(kegg[j][1])
elif kegg[j][0] == 'K12309':
    K12309.append(kegg[j][1])
elif kegg[j][0] == 'K01188':
    K01188.append(kegg[j][1])
elif kegg[j][0] == 'K05349':
    K05349.append(kegg[j][1])
elif kegg[j][0] == 'K05350':
    K05350.append(kegg[j][1])
elif kegg[j][0] == 'K01198':
    K01198.append(kegg[j][1])
elif kegg[j][0] == 'K15920':
    K15920.append(kegg[j][1])
elif kegg[j][0] == 'K22268':
    K22268.append(kegg[j][1])
elif kegg[j][0] == 'K01179':
    K01179.append(kegg[j][1])
elif kegg[j][0] == 'K19357':
    K19357.append(kegg[j][1])
elif kegg[j][0] == 'K20542':
```

```
        K20542.append(kegg[j][1])
    elif kegg[j][0] == 'K01180':
        K01180.append(kegg[j][1])
    elif kegg[j][0] == 'K20846':
        K20846.append(kegg[j][1])
    elif kegg[j][0] == 'K20850':
        K20850.append(kegg[j][1])
    elif kegg[j][0] == 'K01219':
        K01219.append(kegg[j][1])
    elif kegg[j][0] == 'K20851':
        K20851.append(kegg[j][1])
    elif kegg[j][0] == 'K01200':
        K01200.append(kegg[j][1])
    elif kegg[j][0] == 'K21575':
        K21575.append(kegg[j][1])
    elif kegg[j][0] == 'K01177':
        K01177.append(kegg[j][1])
    elif kegg[j][0] == 'K01208':
        K01208.append(kegg[j][1])
    elif kegg[j][0] == 'K05992':
        K05992.append(kegg[j][1])
    elif kegg[j][0] == 'K22253':
        K22253.append(kegg[j][1])
    elif kegg[j][0] == 'K01178':
        K01178.append(kegg[j][1])
    elif kegg[j][0] == 'K12047':
        K12047.append(kegg[j][1])
    elif kegg[j][0] == 'K21574':
        K21574.append(kegg[j][1])
    elif kegg[j][0] == 'K07024':
        K07024.append(kegg[j][1])
    elif kegg[j][0] == 'K01193':
        K01193.append(kegg[j][1])
    elif kegg[j][0] == 'K00064':
        K00064.append(kegg[j][1])
    elif kegg[j][0] == 'K17993':
        K17993.append(kegg[j][1])
```

```
elif kegg[j][0] == 'K02567':
    K02567.append(kegg[j][1])
elif kegg[j][0] == 'K03778':
    K03778.append(kegg[j][1])
elif kegg[j][0] == 'K00955':
    K00955.append(kegg[j][1])
elif kegg[j][0] == 'K05907':
    K05907.append(kegg[j][1])
elif kegg[j][0] == 'K17229':
    K17229.append(kegg[j][1])
elif kegg[j][0] == 'K01225':
    K01225.append(kegg[j][1])
elif kegg[j][0] == 'K19668':
    K19668.append(kegg[j][1])
elif kegg[j][0] == 'K08688':
    K08688.append(kegg[j][1])
elif kegg[j][0] == 'K00301':
    K00301.append(kegg[j][1])
elif kegg[j][0] == 'K00302':
    K00302.append(kegg[j][1])
elif kegg[j][0] == 'K00303':
    K00303.append(kegg[j][1])
elif kegg[j][0] == 'K00304':
    K00304.append(kegg[j][1])
elif kegg[j][0] == 'K00305':
    K00305.append(kegg[j][1])
elif kegg[j][0] == 'K03851':
    K03851.append(kegg[j][1])
elif kegg[j][0] == 'K03852':
    K03852.append(kegg[j][1])
elif kegg[j][0] == 'K01130':
    K01130.append(kegg[j][1])
elif kegg[j][0] == 'K15923':
    K15923.append(kegg[j][1])
elif kegg[j][0] == 'K00879':
    K00879.append(kegg[j][1])
elif kegg[j][0] == 'K01628':
```

```
        K01628.append(kegg[j][1])
    elif kegg[j][0] == 'K00848':
        K00848.append(kegg[j][1])
    elif kegg[j][0] == 'K01629':
        K01629.append(kegg[j][1])
    elif kegg[j][0] == 'K01183':
        K01183.append(kegg[j][1])
    elif kegg[j][0] == 'K13381':
        K13381.append(kegg[j][1])
    elif kegg[j][0] == 'K14083':
        K14083.append(kegg[j][1])
    elif kegg[j][0] == 'K16178':
        K16178.append(kegg[j][1])
    elif kegg[j][0] == 'K16176':
        K16176.append(kegg[j][1])
    elif kegg[j][0] == 'K00702':
        K00702.append(kegg[j][1])
    elif kegg[j][0] == 'K16149':
        K16149.append(kegg[j][1])
    elif kegg[j][0] == 'K00975':
        K00975.append(kegg[j][1])
    elif kegg[j][0] == 'K00703':
        K00703.append(kegg[j][1])
    elif kegg[j][0] == 'K16146':
        K16146.append(kegg[j][1])
    elif kegg[j][0] == 'K16147':
        K16147.append(kegg[j][1])
    elif kegg[j][0] == 'K01176':
        K01176.append(kegg[j][1])
    elif kegg[j][0] == 'K05973':
        K05973.append(kegg[j][1])
    elif kegg[j][0] == 'K03430':
        K03430.append(kegg[j][1])
    elif kegg[j][0] == 'K05306':
        K05306.append(kegg[j][1])
    elif kegg[j][0] == 'K11472':
        K11472.append(kegg[j][1])
```

```
    elif kegg[j][0] == 'K01941':
        K01941.append(kegg[j][1])
    else:
        other.append(kegg[j][1])
print('KEGG numbers grouped under respective KEGG number variables.')

####Splitting KEGGs - assimilatory/dissimilatory sulfate reduction and ammonia/methane
monooxygenase
limit = 14

## Split K00394
#get sequences for the marker
K00394_ORFseq = []
with open('%s', 'r') as orf_file:
    for record in SeqIO.parse(orf_file, "fasta"):
        for ORF in K00394_ORFname:
            if ORF == record.id:
                K00394_ORFseq.append(record)
marker = K00394_ORFseq ###ORFS
#get the database sequences
db = []
with
open('/srv/scratch/jgi/Cavlab_pipeline_resources/v4/K00394_pathway_database_v1.fasta','r') as
aprA_file:
    db = list(SeqIO.parse(aprA_file, "fasta"))
#make a prediction
for ORF in marker:
    scorei = []
    for seq in db:
        align = pairwise2.align.localms(ORF.seq,seq.seq,2,-1,-.5,-.1,score_only=1)
        scorei.append(align)
        seq.description = align
    keep = max(scorei)
    cat = []
    for seq in db:
        if seq.description >= keep:
            cat.append(seq.id.split('$')[0])
```

```
    dis = 0
    ox = 0
    oth = 0
    un = 0
    for obs in cat:
        if obs == 'Reductive':
            dis = dis + 1
        elif obs == 'Oxidative':
            ox = ox + 1
        elif obs == 'Other':
            oth = oth + 1
        else:
            un = un + 1
    if keep > limit:
        if dis > ox and dis > oth:
            assignment = 'Reductive'
            K00394r.append(float(prtcov[ORF.id]))
        elif ox > dis and ox > oth:
            assignment = 'Oxidative'
            K00394o.append(float(prtcov[ORF.id]))
        elif oth > dis and oth > ox:
            assignment = 'Other'
        else:
            assignment = 'Unknown'
print('K00394 - assimilatory/diisimilatory function prediction complete.')

## Split K00395
#get sequences for the marker
K00395_ORFseq = []
with open('%s', ‘r’) as orf_file:
    for record in SeqIO.parse(orf_file, “fasta”):
        for ORF in K00395_ORFname:
            if ORF == record.id:
                K00395_ORFseq.append(record)
marker=K00395_ORFseq ###ORFS
#get the database sequences
db = []
```

```
with open('/srv/scratch/jgi/Cavlab_pipeline_resources/v4/K00395_pathway_database_v1.fasta',
‘r’) as aprB_file:
    db = list(SeqIO.parse(aprB_file, “fasta”))
#make a prediction
for ORF in marker:
    scorei = []
    for seq in db:
        align = pairwise2.align.localms(ORF.seq,seq.seq,2,-1,-.5,-.1,score_only=1)
        scorei.append(align)
        seq.description = align
    keep = max(scorei)
    cat = []
    for seq in db:
        if seq.description >= keep:
            cat.append(seq.id.split('$')[0])
    dis = 0
    ox = 0
    oth = 0
    un = 0
    for obs in cat:
        if obs == 'Reductive':
            dis = dis + 1
        elif obs == 'Oxidative':
            ox = ox + 1
        elif obs == 'Other':
            oth = oth + 1
        else:
            un = un + 1
    if keep > limit:
        if dis > ox and dis > oth:
            assignment = 'Reductive'
            K00395r.append(float(prtcov[ORF.id]))
        elif ox > dis and ox > oth:
            assignment = 'Oxidative'
            K00395o.append(float(prtcov[ORF.id]))
        elif oth > dis and oth > ox:
            assignment = 'Other'
```

```
        else:
            assignment = 'Unknown'
print('K00395 - assimilatory/diisimilatory function prediction complete.')

## Split K11180
#get sequences for the marker
K11180_ORFseq = []
with open('%s', ‘r’) as orf_file:
    for record in SeqIO.parse(orf_file, “fasta”):
        for ORF in K11180_ORFname:
            if ORF == record.id:
                K11180_ORFseq.append(record)
marker = K11180_ORFseq ###ORFS
#get the database sequences
db = []
with open('/srv/scratch/jgi/Cavlab_pipeline_resources/v4/K11180_pathway_database_v1.fasta',
‘r’) as dsrA_file:
    db = list(SeqIO.parse(dsrA_file, “fasta”))
#make a prediction
for ORF in marker:
    scorei = []
    for seq in db:
        align = pairwise2.align.localms(ORF.seq,seq.seq,2,-1,-.5,-.1,score_only=1)
        scorei.append(align)
        seq.description = align
    keep = max(scorei)
    cat = []
    for seq in db:
        if seq.description >= keep:
            cat.append(seq.id.split('$')[0])
    dis = 0
    ox = 0
    oth = 0
    un = 0
    for obs in cat:
        if obs == 'Reductive':
            dis = dis + 1
```

```
        elif obs == 'Oxidative':
            ox = ox + 1
        elif obs == 'Other':
            oth = oth + 1
        else:
            un = un + 1
    if keep > limit:
        if dis > ox and dis > oth:
            assignment = 'Reductive'
            K11180r.append(float(prtcov[ORF.id]))
        elif ox > dis and ox > oth:
            assignment = 'Oxidative'
            K11180o.append(float(prtcov[ORF.id]))
        elif oth > dis and oth > ox:
            assignment = 'Other'
        else:
            assignment = 'Unknown'
print('K11180 - assimilatory/diisimilatory function prediction complete.')

## Split K11181
#get sequences for the marker
K11181_ORFseq = []
with open('%s', 'r') as orf_file:
    for record in SeqIO.parse(orf_file, "fasta"):
        for ORF in K11181_ORFname:
            if ORF == record.id:
                K11181_ORFseq.append(record)
marker = K11181_ORFseq ###ORFS
#get the database sequences
db = []
with open('/srv/scratch/jgi/Cavlab_pipeline_resources/v4/K11181_pathway_database_v1.fasta',
'r') as dsrB_file:
    db = list(SeqIO.parse(dsrB_file, "fasta"))
#make a prediction
for ORF in marker:
    scorei = []
    for seq in db:
```

```
        align = pairwise2.align.localms(ORF.seq,seq.seq,2,-1,-.5,-.1,score_only=1)
        scorei.append(align)
        seq.description = align
    keep = max(scorei)
    cat = []
    for seq in db:
        if seq.description >= keep:
            cat.append(seq.id.split('$')[0])
    dis = 0
    ox = 0
    oth = 0
    un = 0
    for obs in cat:
        if obs == 'Reductive':
            dis = dis + 1
        elif obs == 'Oxidative':
            ox = ox + 1
        elif obs == 'Other':
            oth = oth + 1
        else:
            un = un + 1
    if keep > limit:
        if dis > ox and dis > oth:
            assignment = 'Reductive'
            K11181r.append(float(prtcov[ORF.id]))
        elif ox > dis and ox > oth:
            assignment = 'Oxidative'
            K11181o.append(float(prtcov[ORF.id]))
        elif oth > dis and oth > ox:
            assignment = 'Other'
        else:
            assignment = 'Unknown'
print('K11181 - assimilatory/diisimilatory function prediction complete.')

## Split K00958
#get sequences for the marker
K00958_ORFseq = []
```

```
with open('%s', ‘r’) as orf_file:
    for record in SeqIO.parse(orf_file, “fasta”):
        for ORF in K00958_ORFname:
            if ORF == record.id:
                K00958_ORFseq.append(record)
marker = K00958_ORFseq ###ORFS
#get the database sequences
db = []
with open('/srv/scratch/jgi/Cavlab_pipeline_resources/v4/K00958_pathway_database_v1.fasta',
‘r’) as sat_file:
    db = list(SeqIO.parse(sat_file, “fasta”))
#make a prediction
for ORF in marker:
    scorei = []
    for seq in db:
        align = pairwise2.align.localms(ORF.seq,seq.seq,2,-1,-.5,-.1,score_only=1)
        scorei.append(align)
        seq.description = align
    keep = max(scorei)
    cat = []
    for seq in db:
        if seq.description >= keep:
            cat.append(seq.id.split('$')[0])
    dis = 0
    ox = 0
    oth = 0
    un = 0
    for obs in cat:
        if obs == 'Reductive':
            dis = dis + 1
        elif obs == 'Oxidative':
            ox = ox + 1
        elif obs == 'Other':
            oth = oth + 1
        else:
            un = un + 1
    if keep > limit:
```

```
        if dis > ox and dis > oth:
            assignment = 'Reductive'
            K00958r.append(float(prtcov[ORF.id]))
        elif ox > dis and ox > oth:
            assignment = 'Oxidative'
            K00958o.append(float(prtcov[ORF.id]))
        elif oth > dis and oth > ox:
            assignment = 'Other'
        else:
            assignment = 'Unknown'
print('K00958 - assimilatory/diisimilatory function prediction complete.')

## Split K10944
#get sequences for the marker
K10944_ORFseq = []
with open('%s', ‘r’) as orf_file:
    for record in SeqIO.parse(orf_file, “fasta”):
        for ORF in K10944_ORFname:
            if ORF == record.id:
                K10944_ORFseq.append(record)
marker = K10944_ORFseq ###ORFS
#get the database sequences
db = []
with open('/srv/scratch/jgi/Cavlab_pipeline_resources/v4/K10944_pathway_database_v1.fasta',
‘r’) as maMOA_file:
    db = list(SeqIO.parse(maMOA_file, “fasta”))
#make a prediction
for ORF in marker:
    scorei = []
    for seq in db:
        align = pairwise2.align.localms(ORF.seq,seq.seq,2,-1,-.5,-.1,score_only=1)
        scorei.append(align)
        seq.description = align
    keep = max(scorei)
    cat = []
    for seq in db:
        if seq.description >= keep:
```

```
            cat.append(seq.id.split('$')[0])
    meth = 0
    ammo = 0
    oth = 0
    un = 0
    for obs in cat:
        if obs == 'MMO':
            meth = meth + 1
        elif obs == 'AMO':
            ammo = ammo + 1
        elif obs == 'Other':
            oth = oth + 1
        else:
            un = un + 1
    if keep > limit:
        if meth > ammo and meth > oth:
            assignment = 'MMO'
            K10944m.append(float(prtcov[ORF.id]))
        elif ammo > dis and ammo > oth:
            assignment = 'AMO'
            K10944a.append(float(prtcov[ORF.id]))
        elif oth > meth and oth > ammo:
            assignment = 'Other'
        else:
            assignment = 'Unknown'
print('K10944 - ammonia/methane monooxygenase function prediction complete.')

## split K10945
#get sequences for the marker
K10945_ORFseq = []
with open('%s', ‘r’) as orf_file:
    for record in SeqIO.parse(orf_file, “fasta”):
        for ORF in K10945_ORFname:
            if ORF == record.id:
                K10945_ORFseq.append(record)
marker = K10945_ORFseq ###ORFS
#get the database sequences
```

```
db = []
with open('/srv/scratch/jgi/Cavlab_pipeline_resources/v4/K10945_pathway_database_v1.fasta',
‘r’) as maMOB_file:
   db = list(SeqIO.parse(maMOB_file, “fasta”))
#make a prediction
for ORF in marker:
   scorei = []
   for seq in db:
      align = pairwise2.align.localms(ORF.seq,seq.seq,2,-1,-.5,-.1,score_only=1)
      scorei.append(align)
      seq.description = align
   keep = max(scorei)
   cat = []
   for seq in db:
      if seq.description >= keep:
         cat.append(seq.id.split('$')[0])
   meth = 0
   ammo = 0
   oth = 0
   un = 0
   for obs in cat:
      if obs == 'MMO':
         meth = meth + 1
      elif obs == 'AMO':
         ammo = ammo + 1
      elif obs == 'Other':
         oth = oth + 1
      else:
         un = un + 1
   if keep > limit:
      if meth > ammo and meth > oth:
         assignment = 'MMO'
         K10945m.append(float(prtcov[ORF.id]))
      elif ammo > dis and ammo > oth:
         assignment = 'AMO'
         K10945a.append(float(prtcov[ORF.id]))
      elif oth > meth and oth > ammo:
```

```
            assignment = 'Other'
        else:
            assignment = 'Unknown'
print('K10945 - ammonia/methane monooxygenase function prediction complete.')

## Split K10946
#get sequences for the marker
K10946_ORFseq = []
with open('%s', ‘r’) as orf_file:
    for record in SeqIO.parse(orf_file, “fasta”):
        for ORF in K10946_ORFname:
            if ORF == record.id:
                K10946_ORFseq.append(record)
marker = K10946_ORFseq ###ORFS
#get the database sequences
db = []
with open('/srv/scratch/jgi/Cavlab_pipeline_resources/v4/K10946_pathway_database_v1.fasta',
‘r’) as maMOC_file:
    db = list(SeqIO.parse(maMOC_file, “fasta”))
#make a prediction
for ORF in marker:
    scorei = []
    for seq in db:
        align = pairwise2.align.localms(ORF.seq,seq.seq,2,-1,-.5,-.1,score_only=1)
        scorei.append(align)
        seq.description = align
    keep = max(scorei)
    cat = []
    for seq in db:
        if seq.description >= keep:
            cat.append(seq.id.split('$')[0])
    meth = 0
    ammo = 0
    oth = 0
    un = 0
    for obs in cat:
        if obs == 'MMO':
```

```
            meth = meth + 1
        elif obs == 'AMO':
            ammo = ammo + 1
        elif obs == 'Other':
            oth = oth + 1
        else:
            un = un + 1
    if keep > limit:
        if meth > ammo and meth > oth:
            assignment = 'MMO'
            K10946m.append(float(prtcov[ORF.id]))
        elif ammo > dis and ammo > oth:
            assignment = 'AMO'
            K10946a.append(float(prtcov[ORF.id]))
        elif oth > meth and oth > ammo:
            assignment = 'Other'
        else:
            assignment = 'Unknown'
print('K10946 - ammonia/methane monooxygenase function prediction complete.')

#### Normalizing by coverage
K00437c = sum(K00437)
K00436c = sum(K00436)
K18332c = sum(K18332)
K17997c = sum(K17997)
K00532c = sum(K00532)
K00533c = sum(K00533)
K05922c = sum(K05922)
K18016c = sum(K18016)
K14068c = sum(K14068)
K00440c = sum(K00440)
K13942c = sum(K13942)
K14126c = sum(K14126)
K01915c = sum(K01915)
K00264c = sum(K00264)
K00265c = sum(K00265)
K00266c = sum(K00266)
```

K00284c = sum(K00284)

K00864c = sum(K00864)

K00005c = sum(K00005)

K19128c = sum(K19128)

K19117c = sum(K19117)

K00169c = sum(K00169)

K00170c = sum(K00170)

K00016c = sum(K00016)

K00174c = sum(K00174)

K00175c = sum(K00175)

K00244c = sum(K00244)

K00194c = sum(K00194)

K00197c = sum(K00197)

K00360c = sum(K00360)

K00367c = sum(K00367)

K10944ac = sum(K10944a)

K10945ac = sum(K10945a)

K10946ac = sum(K10946a)

K10944mc = sum(K10944m)

K10945mc = sum(K10945m)

K10946mc = sum(K10946m)

K20932c = sum(K20932)

K20933c = sum(K20933)

K20934c = sum(K20934)

K20935c = sum(K20935)

K00456c = sum(K00456)

K01011c = sum(K01011)

K00860c = sum(K00860)

K00956c = sum(K00956)

K00957c = sum(K00957)

K19087c = sum(K19087)

K19046c = sum(K19046)

K19127c = sum(K19127)

K19129c = sum(K19129)

K03385c = sum(K03385)

K17877c = sum(K17877)

K00366c = sum(K00366)

K02305c = sum(K02305)

K04561c = sum(K04561)

K00376c = sum(K00376)

K02586c = sum(K02586)

K02591c = sum(K02591)

K10535c = sum(K10535)

K01602c = sum(K01602)

K00855c = sum(K00855)

K15230c = sum(K15230)

K15231c = sum(K15231)

K15234c = sum(K15234)

K15233c = sum(K15233)

K15232c = sum(K15232)

K03518c = sum(K03518)

K03519c = sum(K03519)

K03520c = sum(K03520)

K02256c = sum(K02256)

K02262c = sum(K02262)

K02274c = sum(K02274)

K02276c = sum(K02276)

K00401c = sum(K00401)

K00400c = sum(K00400)

K16157c = sum(K16157)

K16158c = sum(K16158)

K16159c = sum(K16159)

K16161c = sum(K16161)

K00390c = sum(K00390)

K00392c = sum(K00392)

K00380c = sum(K00380)

K00381c = sum(K00381)

K00394rc = sum(K00394r)

K00395rc = sum(K00395r)

K11180rc = sum(K11180r)

K11181rc = sum(K11181r)

K00394oc = sum(K00394o)

K00395oc = sum(K00395o)

K11180oc = sum(K11180o)

K11181oc = sum(K11181o)

K17224c = sum(K17224)

K17227c = sum(K17227)

K17226c = sum(K17226)

K17222c = sum(K17222)

K17223c = sum(K17223)

K17225c = sum(K17225)

K03821c = sum(K03821)

K15342c = sum(K15342)

K09951c = sum(K09951)

K07012c = sum(K07012)

K07475c = sum(K07475)

K19088c = sum(K19088)

K19123c = sum(K19123)

K19127c = sum(K19127)

K07016c = sum(K07016)

K19138c = sum(K19138)

K19141c = sum(K19141)

K09952c = sum(K09952)

K19137c = sum(K19137)

K07464c = sum(K07464)

K02703c = sum(K02703)

K02706c = sum(K02706)

K02705c = sum(K02705)

K02704c = sum(K02704)

K02707c = sum(K02707)

K02708c = sum(K02708)

K02689c = sum(K02689)

K02690c = sum(K02690)

K02691c = sum(K02691)

K02692c = sum(K02692)

K02693c = sum(K02693)

K02694c = sum(K02694)

K08928c = sum(K08928)

K08929c = sum(K08929)

K08940c = sum(K08940)

K08941c = sum(K08941)

K08942c = sum(K08942)
K08943c = sum(K08943)
K04643c = sum(K04643)
K04642c = sum(K04642)
K04641c = sum(K04641)
K04250c = sum(K04250)
K00909c = sum(K00909)
K01428c = sum(K01428)
K01429c = sum(K01429)
K01430c = sum(K01430)
K00111c = sum(K00111)
K00112c = sum(K00112)
K00113c = sum(K00113)
K00096c = sum(K00096)
K00518c = sum(K00518)
K04564c = sum(K04564)
K04565c = sum(K04565)
K16627c = sum(K16627)
K06164c = sum(K06164)
K05780c = sum(K05780)
K06165c = sum(K06165)
K06166c = sum(K06166)
K06163c = sum(K06163)
K08977c = sum(K08977)
K09836c = sum(K09836)
K15746c = sum(K15746)
K16953c = sum(K16953)
K17486c = sum(K17486)
K07306c = sum(K07306)
K17218c = sum(K17218)
K03553c = sum(K03553)
K00370c = sum(K00370)
K00368c = sum(K00368)
K11959c = sum(K11959)
K11960c = sum(K11960)
K11961c = sum(K11961)
K11962c = sum(K11962)

K11963c = sum(K11963)

K02048c = sum(K02048)

K02046c = sum(K02046)

K02047c = sum(K02047)

K02045c = sum(K02045)

K15576c = sum(K15576)

K15577c = sum(K15577)

K15578c = sum(K15578)

K15579c = sum(K15579)

K11950c = sum(K11950)

K11951c = sum(K11951)

K11952c = sum(K11952)

K11953c = sum(K11953)

K15551c = sum(K15551)

K15552c = sum(K15552)

K10831c = sum(K10831)

K15553c = sum(K15553)

K15554c = sum(K15554)

K15555c = sum(K15555)

K11069c = sum(K11069)

K11070c = sum(K11070)

K11071c = sum(K11071)

K11072c = sum(K11072)

K11073c = sum(K11073)

K11074c = sum(K11074)

K11075c = sum(K11075)

K11076c = sum(K11076)

K02040c = sum(K02040)

K02037c = sum(K02037)

K02038c = sum(K02038)

K02036c = sum(K02036)

K02044c = sum(K02044)

K02042c = sum(K02042)

K02041c = sum(K02041)

K11081c = sum(K11081)

K11082c = sum(K11082)

K11083c = sum(K11083)

K11084c = sum(K11084)

K02002c = sum(K02002)

K02001c = sum(K02001)

K02000c = sum(K02000)

K05845c = sum(K05845)

K05846c = sum(K05846)

K05847c = sum(K05847)

K10108c = sum(K10108)

K10109c = sum(K10109)

K10110c = sum(K10110)

K15770c = sum(K15770)

K15771c = sum(K15771)

K15772c = sum(K15772)

K10117c = sum(K10117)

K10118c = sum(K10118)

K10119c = sum(K10119)

K10232c = sum(K10232)

K10233c = sum(K10233)

K10234c = sum(K10234)

K10235c = sum(K10235)

K10196c = sum(K10196)

K10197c = sum(K10197)

K10198c = sum(K10198)

K10199c = sum(K10199)

K17315c = sum(K17315)

K17316c = sum(K17316)

K17317c = sum(K17317)

K10236c = sum(K10236)

K10237c = sum(K10237)

K10238c = sum(K10238)

K17311c = sum(K17311)

K17312c = sum(K17312)

K17313c = sum(K17313)

K17314c = sum(K17314)

K10200c = sum(K10200)

K10201c = sum(K10201)

K10202c = sum(K10202)

K10240c = sum(K10240)

K10241c = sum(K10241)

K10242c = sum(K10242)

K17329c = sum(K17329)

K17330c = sum(K17330)

K17331c = sum(K17331)

K17244c = sum(K17244)

K17245c = sum(K17245)

K17246c = sum(K17246)

K10537c = sum(K10537)

K10538c = sum(K10538)

K10539c = sum(K10539)

K10188c = sum(K10188)

K10189c = sum(K10189)

K10190c = sum(K10190)

K10191c = sum(K10191)

K10543c = sum(K10543)

K10544c = sum(K10544)

K10545c = sum(K10545)

K17326c = sum(K17326)

K17327c = sum(K17327)

K17328c = sum(K17328)

K10546c = sum(K10546)

K10547c = sum(K10547)

K10548c = sum(K10548)

K10552c = sum(K10552)

K10553c = sum(K10553)

K10554c = sum(K10554)

K10559c = sum(K10559)

K10560c = sum(K10560)

K10561c = sum(K10561)

K10562c = sum(K10562)

K10439c = sum(K10439)

K10440c = sum(K10440)

K10441c = sum(K10441)

K17202c = sum(K17202)

K17203c = sum(K17203)

K17204c = sum(K17204)

K10120c = sum(K10120)

K10121c = sum(K10121)

K10122c = sum(K10122)

K17321c = sum(K17321)

K17322c = sum(K17322)

K17323c = sum(K17323)

K17324c = sum(K17324)

K17325c = sum(K17325)

K02027c = sum(K02027)

K02025c = sum(K02025)

K02026c = sum(K02026)

K02058c = sum(K02058)

K02057c = sum(K02057)

K02056c = sum(K02056)

K10013c = sum(K10013)

K10015c = sum(K10015)

K10016c = sum(K10016)

K10017c = sum(K10017)

K10014c = sum(K10014)

K10036c = sum(K10036)

K10037c = sum(K10037)

K10038c = sum(K10038)

K09996c = sum(K09996)

K09997c = sum(K09997)

K09998c = sum(K09998)

K09999c = sum(K09999)

K10000c = sum(K10000)

K10001c = sum(K10001)

K10002c = sum(K10002)

K10003c = sum(K10003)

K10004c = sum(K10004)

K10039c = sum(K10039)

K10040c = sum(K10040)

K10041c = sum(K10041)

K10018c = sum(K10018)

K10019c = sum(K10019)

K10020c = sum(K10020)

K10021c = sum(K10021)

K09969c = sum(K09969)

K09970c = sum(K09970)

K09971c = sum(K09971)

K09972c = sum(K09972)

K10005c = sum(K10005)

K10006c = sum(K10006)

K10007c = sum(K10007)

K10008c = sum(K10008)

K02424c = sum(K02424)

K10009c = sum(K10009)

K10010c = sum(K10010)

K16956c = sum(K16956)

K16957c = sum(K16957)

K16958c = sum(K16958)

K16959c = sum(K16959)

K16960c = sum(K16960)

K10022c = sum(K10022)

K10023c = sum(K10023)

K10024c = sum(K10024)

K10025c = sum(K10025)

K23059c = sum(K23059)

K17077c = sum(K17077)

K23060c = sum(K23060)

K01999c = sum(K01999)

K01997c = sum(K01997)

K01998c = sum(K01998)

K01995c = sum(K01995)

K01996c = sum(K01996)

K11954c = sum(K11954)

K11955c = sum(K11955)

K11956c = sum(K11956)

K11957c = sum(K11957)

K11958c = sum(K11958)

K02073c = sum(K02073)

K02072c = sum(K02072)

K02071c = sum(K02071)

K15580c = sum(K15580)

K15581c = sum(K15581)

K15582c = sum(K15582)

K15583c = sum(K15583)

K10823c = sum(K10823)

K12368c = sum(K12368)

K12369c = sum(K12369)

K12370c = sum(K12370)

K12371c = sum(K12371)

K12372c = sum(K12372)

K16199c = sum(K16199)

K16200c = sum(K16200)

K16201c = sum(K16201)

K16202c = sum(K16202)

K01216c = sum(K01216)

K01199c = sum(K01199)

K19891c = sum(K19891)

K19892c = sum(K19892)

K19893c = sum(K19893)

K01190c = sum(K01190)

K12111c = sum(K12111)

K12308c = sum(K12308)

K12309c = sum(K12309)

K01188c = sum(K01188)

K05349c = sum(K05349)

K05350c = sum(K05350)

K01198c = sum(K01198)

K15920c = sum(K15920)

K22268c = sum(K22268)

K01179c = sum(K01179)

K19357c = sum(K19357)

K20542c = sum(K20542)

K01180c = sum(K01180)

K20846c = sum(K20846)

K20850c = sum(K20850)

K01219c = sum(K01219)

K20851c = sum(K20851)

K01200c = sum(K01200)

K21575c = sum(K21575)

K01177c = sum(K01177)

K01208c = sum(K01208)

K05992c = sum(K05992)

K22253c = sum(K22253)

K01178c = sum(K01178)

K12047c = sum(K12047)

K21574c = sum(K21574)

K07024c = sum(K07024)

K01193c = sum(K01193)

K00064c = sum(K00064)

K17993c = sum(K17993)

K02567c = sum(K02567)

K03778c = sum(K03778)

K00955c = sum(K00955)

K05907c = sum(K05907)

K17229c = sum(K17229)

K00958rc = sum(K00958r)

K00958oc = sum(K00958o)

K01225c = sum(K01225)

K19668c = sum(K19668)

K08688c = sum(K08688)

K00301c = sum(K00301)

K00302c = sum(K00302)

K00303c = sum(K00303)

K00304c = sum(K00304)

K00305c = sum(K00305)

K03851c = sum(K03851)

K03852c = sum(K03852)

K01130c = sum(K01130)

K15923c = sum(K15923)

K00879c = sum(K00879)

K01628c = sum(K01628)

K00848c = sum(K00848)

K01629c = sum(K01629)

```
K01183c = sum(K01183)
K13381c = sum(K13381)
K14083c = sum(K14083)
K16178c = sum(K16178)
K16176c = sum(K16176)
K00702c = sum(K00702)
K16149c = sum(K16149)
K00975c = sum(K00975)
K00703c = sum(K00703)
K16146c = sum(K16146)
K16147c = sum(K16147)
K01176c = sum(K01176)
K05973c = sum(K05973)
K03430c = sum(K03430)
K05306c = sum(K05306)
K11472c = sum(K11472)
K01941c = sum(K01941)
otherc = sum(other)
print('KEGG number coverages calculated.')

#### Calculating pathway coverages
#C cycle
fermentation_c = K00016c + K03778c + np.average([K00169c, K00170c])
respiration_c = np.average([K02256c, K02262c, K02274c, K02276c])
methanogenesis_c = np.average([K00400c, K00401c])
methane_oxidation_c = np.average([K16157c, K16158c, K16159c, K16161c, K10944mc,
K10945mc, K10946mc])
MoCu_CODH_c = np.average([K03518c, K03519c, K03520c])
rTCA_I_c = np.average([K15230c, K15231c]) ##for GSB mostly
rTCA_II_c = np.average([K15234c, K15233c, K15232c, K00174c, K00175c, K00244c]) ##for
other organisms
WL_c = np.average([K00194c, K00197c])
calvin_cycle_c = np.average([K01602c, K00855c])
carbon_fixation_c = rTCA_I_c + rTCA_II_c + WL_c + calvin_cycle_c
#N cycle
nitrogen_fixation_c = np.average([K02586c, K02591c])
ammonia_assimilation_c = np.average([K01915c, K00264c, K00265c, K00266c, K00284c])
```

```
assimilatory_nitrate_reduction_c = np.average([K17877c, K00366c, K00360c, K00367c])
nitrification_c = np.average([K10535c, K10944ac, K10945ac, K10946ac])
anammox_c = np.average([K20932c, K20933c, K20934c, K20935c])
periplasmic_nitrate_reduction_c = K02567c
dissimilatory_nitrate_reduction_c = K00370c
dissimilatory_nitrite_reduction_ammonia_forming_c = K03385c
dissimilatory_nitrite_reduction_NO_forming_c = K00368c
nitric_oxide_reduction_c = np.average([K02305c, K04561c, K00376c])
denitrification_c = dissimilatory_nitrate_reduction_c + periplasmic_nitrate_reduction_c +
dissimilatory_nitrite_reduction_NO_forming_c + nitric_oxide_reduction_c
#S cycle
sox_c = np.average([K17224c, K17227c, K17226c, K17222c, K17223c, K17225c])
cysteine_dioxygenase_c = K00456c
thiosulfate_mercaptopyruvate_sulfurtransferase_c = K01011c
sulfate_reduction_I_c = np.average([K00958rc, K00956c, K00957c, K00955c])
sulfate_reduction_II_c = K00958rc
APS_reduction_I_c = np.average([K00860c, K00955c])
APS_reduction_II_c = np.average([K05907c, K00390c])
APS_reduction_III_c = np.average([K00394rc, K00395rc])
PAPS_reduction_c = K00390c
sulfite_reduction_I_c = np.average([K00380c, K00381c, K00392c])
sulfite_reduction_II_c = np.average([K11180rc, K11181rc])
sulfide_oxidation_c = np.average([K17218c, K17229c])
sulfur_polysulfide_oxidation_c = np.average([K11180oc, K11181oc])
sulfite_oxidation_c = np.average([K00394oc, K00395oc])
APS_oxidation_c = K00958oc
assimilatory_sulfate_reduction_I_c = sulfate_reduction_I_c + APS_reduction_I_c +
PAPS_reduction_c + sulfite_reduction_I_c
assimilatory_sulfate_reduction_II_c = sulfate_reduction_I_c + APS_reduction_II_c +
sulfite_reduction_I_c
dissimilatory_sulfate_reduction_c = sulfate_reduction_II_c + APS_reduction_III_c +
sulfite_reduction_II_c
sulfide_oxidation_sulfate_c = sulfide_oxidation_c + sulfur_polysulfide_oxidation_c +
sulfite_oxidation_c + APS_oxidation_c
#Photosystems
photosystem_II_c = np.average([K02703c, K02706c, K02705c, K02704c, K02707c, K02708c])
photosystem_I_c = np.average([K02689c, K02690c, K02691c, K02692c, K02693c, K02694c])
```

```
anoxygenic_photosystem_II_c = np.average([K08928c, K08929c])
anoxygenic_photosystem_I_c = np.average([K08940c, K08941c, K08942c, K08943c])
rhodopsins_c = K04643c + K04642c + K04641c + K04250c + K00909c
astaxanthin_c = np.average([K09836c, K15746c])
bacterioruberin_c = K08977c
#CRISPR
CRISPR_Cas_spacer_acquisition_c = np.average([K15342c, K09951c])
CRISPR_1I_c = np.average([K07012c, K07475c])
CRISPR_1IA_c = np.average([K19088c, K19087c])
CRISPR_1IC_c = K19117c
CRISPR_1IE_c = np.average([K19123c, K19046c])
CRISPR_1IF_c = np.average([K19127c, K19128c, K19129c])
CRISPR_2II_c = K09952c
CRISPR_2IIA_c = K19137c
CRISPR_2IIB_c = K07464c
CRISPR_1III_c = K07016c
CRISPR_1IIIA_c = K19138c
CRISPR_1IIIB_c = K19141c
#hydrogenases
NiFe_hydrogenase_c = np.average([K00437c, K05922c])
NADreducing_hydrogenase_c = K00436c
NADPreducing_hydrogenase_c = K18332c
Fe_hydrogenase_c = K17997c
ferredoxin_hydrogenase_mono_c = K00532c
ferredoxin_hydrogenase_tri_c = K00533c
membrane_bound_hydrogenase_c = K18016c
methanophenazine_hydrogenase_c = K14068c
coenzyme_F420hydrogenase_c = K00440c
methenyltetrahydromethanopterin_hydrogenase_c = K13942c
F420_nonreducing_hydrogenase_c = K14126c
sulfhydrogenase_c = K17993c
#ABC transporters
Urea_transporter_c = np.average([K11959c, K11960c, K11961c, K11962c, K11963c])
Sulfate_transporter_c = np.average([K02048c, K02046c, K02047c, K02045c])
Nitrate_nitrite_transporter_c = np.average([K15576c, K15577c, K15578c, K15579c])
Bicarbonate_transporter_c = np.average([K11950c, K11951c, K11952c, K11953c])
Taurine_transporter_c = np.average([K15551c, K15552c, K10831c])
```

```
Sulfonate_transporter_c = np.average([K15553c, K15554c, K15555c])
Spermidine_putrescine_transporter_c = np.average([K11069c, K11070c, K11071c, K11072c])
Putrescine_transporter_c = np.average([K11073c, K11074c, K11075c, K11076c])
Phosphate_transporter_c = np.average([K02040c, K02037c, K02038c, K02036c])
Phosphonate_transporter_c = np.average([K02044c, K02042c, K02041c])
Aminoethylphosphonate_transporter_c = np.average([K11081c, K11082c, K11083c, K11084c])
Glycine_betaine_proline_transporter_c = np.average([K02002c, K02001c, K02000c])
Osmoprotectant_transporter_c = np.average([K05845c, K05846c, K05847c])
Maltose_maltodextrin_transporter_c = np.average([K10108c, K10109c, K10110c])
Arabinogalactan_oligomer_maltooligosaccharide_transporter_c = np.average([K15770c,
K15771c, K15772c])
Raffinose_stachyose_melibiose_transporter_c = np.average([K10117c, K10118c, K10119c])
alphaGlucoside_transporter_c = np.average([K10232c, K10233c, K10234c, K10235c])
Glucose_arabinose_transporter_c = np.average([K10196c, K10197c, K10198c, K10199c])
Glucose_mannose_transporter_c = np.average([K17315c, K17316c, K17317c])
Trehalose_maltose_transporter_c = np.average([K10236c, K10237c, K10238c])
Trehalose_transporter_c = np.average([K17311c, K17312c, K17313c, K17314c])
NAcetylglucosamine_transporter_c = np.average([K10200c, K10201c, K10202c])
Cellobiose_transporter_c = np.average([K10240c, K10241c, K10242c])
NNDiacetylchitobiose_transporter_c = np.average([K17329c, K17330c, K17331c])
Putative_chitobiose_transporter_c = np.average([K17244c, K17245c, K17246c])
LArabinose_transporter_c = np.average([K10537c, K10538c, K10539c])
Lactose_Larabinose_transporter_c = np.average([K10188c, K10189c, K10190c, K10191c])
DXylose_transporter_c = np.average([K10543c, K10544c, K10545c])
Xylobiose_transporter_c = np.average([K17326c, K17327c, K17328c])
Multiple_sugar_transporter_c = np.average([K10546c, K10547c, K10548c])
Fructose_transporter_c = np.average([K10552c, K10553c, K10554c])
Rhamnose_transporter_c = np.average([K10559c, K10560c, K10561c, K10562c])
Ribose_transporter_c = np.average([K10439c, K10440c, K10441c])
Erythritol_transporter_c = np.average([K17202c, K17203c, K17204c])
Putative_fructooligosaccharide_transporter_c = np.average([K10120c, K10121c, K10122c])
Glycerol_transporter_c = np.average([K17321c, K17322c, K17323c, K17324c, K17325c])
Putative_multiple_sugar_transporter_c = np.average([K02027c, K02025c, K02026c])
Putative_simple_sugar_transporter_c = np.average([K02058c, K02057c, K02056c])
Lysine_arginine_ornithine_transporter_c = np.average([K10013c, K10015c, K10016c,
K10017c])
Histidine_transporter_c = np.average([K10014c, K10015c, K10016c, K10017c])
```

```
Glutamine_transporter_c = np.average([K10036c, K10037c, K10038c])
Arginine_transporter_c = np.average([K09996c, K09997c, K09998c, K09999c, K10000c])
Glutamate_aspartate_transporter_c = np.average([K10001c, K10002c, K10003c, K10004c])
Aspartate_glutamate_glutamine_transporter_c = np.average([K10039c, K10040c, K10041c])
Octopine_nopaline_transporter_c = np.average([K10018c, K10019c, K10020c, K10021c])
General_Laminoacid_transporter_c = np.average([K09969c, K09970c, K09971c, K09972c])
Glutamate_transporter_c = np.average([K10005c, K10006c, K10007c, K10008c])
Cystine_transporter_c = np.average([K02424c, K10009c, K10010c])
LCystine_transporter_c = np.average([K16956c, K16957c, K16958c, K16959c, K16960c])
Arginine_ornithine_transporter_c = np.average([K10022c, K10023c, K10024c, K10025c])
Arginine_lysine_histidine_transporter_c = np.average([K23059c, K17077c, K23060c])
Branched_chain_aminoacid_transporter_c = np.average([K01999c, K01997c, K01998c,
K01995c, K01996c])
Neutral_aminoacid_transporter_c = np.average([K11954c, K11955c, K11956c, K11957c,
K11958c])
DMethionine_transporter_c = np.average([K02073c, K02072c, K02071c])
Oligopeptide_transporter_c = np.average([K15580c, K15581c, K15582c, K15583c, K10823c])
Dipeptide_transporter_c = np.average([K12368c, K12369c, K12370c, K12371c, K12372c,
K16199c, K16200c, K16201c, K16202c])
#Carbohydrates
licheninase_c = K01216c
glucan_endobeta_glucosidase_c = np.average([K01199c, K19891c, K19892c, K19893c])
beta_galactosidase_c = np.average([K01190c, K12111c, K12308c, K12309c, K01188c,
K05349c, K05350c])
xylan_beta_xylosidase_c = np.average([K01198c, K15920c, K22268c])
cellulase_endoglucanase_c = np.average([K01179c, K19357c, K20542c])
laminarinase_c = K01180c
carrageenase_c = np.average([K20846c, K20850c])
agarase_c = np.average([K01219c, K20851c])
pullulanase_c = np.average([K01200c, K21575c])
beta_amylase_c = K01177c
maltogenic_alpha_amylase_c = np.average([K01208c, K05992c])
exo_amylase_c = K22253c
glucoamylase_glucan_alpha_glucosidase_c = np.average([K01178c, K12047c, K21574c])
sucrose_phosphatase_c = K07024c
beta_fructofuranosidase_c = K01193c
fucose_utilization_II_c = K00064c
```

```
cellobiosidase_c = np.average([K01225c, K19668c])
glycolate_utilization_c = K11472c
creatine_utilization_c = K08688c
sarcosine_utilization_I_c = K00301c
sarcosine_utilization_II_c = np.average([K00302c, K00303c, K00304c, K00305c])
taurine_utilization_c = np.average([K03851c, K03852c])
sulfate_ester_hydrolysis_c = K01130c
fucoidan_degradation_c = K15923c
fucose_utilization_c = np.average([K00879c, K01628c])
rhamnose_utilization_c = np.average([K00848c, K01629c])
chitin_degradation_I_c = K01183c
chitin_degradation_II_c = K13381c
trimethylamine_glycine_betaine_methyltransferase_c = K14083c
dimethylamine_utilization_c = K16178c
monomethylamine_utilization_c = K16176c
cellobiose_utilization_c = K00702c
glycogen_synthesis_overall_c = K16149c
glycogen_synthesis_I_c = np.average([K00975c, K00703c])
glycogen_synthesis_II_c = np.average([K16146c, K16147c])
starch_degradation_c = K01176c
#Others
PHA_storage_c = np.average([K03821c, K05973c])
urea_c = np.average([K01428c, K01429c, K01430c]) + K01941c
glycerol_c = np.average([K00111c, K00112c, K00113c, K00864c])
archaeal_glycerol_c = K00096c
superoxidedismutase_c = np.average([K00518c, K04564c, K04565c, K16627c])
methylphosphonate_catabolism_c = np.average([K06164c, K05780c, K06165c, K06166c,
K06163c])
aminoethylphosphonate_catabolism_c = np.average([K03430c, K05306c])
DMSO_reduction_c = K07306c
DMSP_catabolism_c = K16953c + K17486c
recA_c = K03553c
print('Pathway coverages calculated.')

#### Normalizing by counts
K00437n = len(K00437)-1
K00436n = len(K00436)-1
```

K18332n = len(K18332)-1
K17997n = len(K17997)-1
K00532n = len(K00532)-1
K00533n = len(K00533)-1
K05922n = len(K05922)-1
K18016n = len(K18016)-1
K14068n = len(K14068)-1
K00440n = len(K00440)-1
K13942n = len(K13942)-1
K14126n = len(K14126)-1
K01915n = len(K01915)-1
K00264n = len(K00264)-1
K00265n = len(K00265)-1
K00266n = len(K00266)-1
K00284n = len(K00284)-1
K00864n = len(K00864)-1
K00005n = len(K00005)-1
K19128n = len(K19128)-1
K19117n = len(K19117)-1
K00169n = len(K00169)-1
K00170n = len(K00170)-1
K00016n = len(K00016)-1
K00174n = len(K00174)-1
K00175n = len(K00175)-1
K00244n = len(K00244)-1
K00194n = len(K00194)-1
K00197n = len(K00197)-1
K00360n = len(K00360)-1
K00367n = len(K00367)-1
K10944an = len(K10944a)
K10945an = len(K10945a)
K10946an = len(K10946a)
K10944mn = len(K10944m)
K10945mn = len(K10945m)
K10946mn = len(K10946m)
K20932n = len(K20932)-1
K20933n = len(K20933)-1

K20934n = len(K20934)-1
K20935n = len(K20935)-1
K00456n = len(K00456)-1
K01011n = len(K01011)-1
K00860n = len(K00860)-1
K00956n = len(K00956)-1
K00957n = len(K00957)-1
K19087n = len(K19087)-1
K19046n = len(K19046)-1
K19127n = len(K19127)-1
K19129n = len(K19129)-1
K03385n = len(K03385)-1
K17877n = len(K17877)-1
K00366n = len(K00366)-1
K02305n = len(K02305)-1
K04561n = len(K04561)-1
K00376n = len(K00376)-1
K02586n = len(K02586)-1
K02591n = len(K02591)-1
K10535n = len(K10535)-1
K01602n = len(K01602)-1
K00855n = len(K00855)-1
K15230n = len(K15230)-1
K15231n = len(K15231)-1
K15234n = len(K15234)-1
K15233n = len(K15233)-1
K15232n = len(K15232)-1
K03518n = len(K03518)-1
K03519n = len(K03519)-1
K03520n = len(K03520)-1
K02256n = len(K02256)-1
K02262n = len(K02262)-1
K02274n = len(K02274)-1
K02276n = len(K02276)-1
K00401n = len(K00401)-1
K00400n = len(K00400)-1
K16157n = len(K16157)-1

```
K16158n = len(K16158)-1
K16159n = len(K16159)-1
K16161n = len(K16161)-1
K00390n = len(K00390)-1
K00392n = len(K00392)-1
K00380n = len(K00380)-1
K00381n = len(K00381)-1
K00394rn = len(K00394r)
K00395rn = len(K00395r)
K11180rn = len(K11180r)
K11181rn = len(K11181r)
K00394on = len(K00394o)
K00395on = len(K00395o)
K11180on = len(K11180o)
K11181on = len(K11181o)
K17224n = len(K17224)-1
K17227n = len(K17227)-1
K17226n = len(K17226)-1
K17222n = len(K17222)-1
K17223n = len(K17223)-1
K17225n = len(K17225)-1
K03821n = len(K03821)-1
K15342n = len(K15342)-1
K09951n = len(K09951)-1
K07012n = len(K07012)-1
K07475n = len(K07475)-1
K19088n = len(K19088)-1
K19123n = len(K19123)-1
K19127n = len(K19127)-1
K07016n = len(K07016)-1
K19138n = len(K19138)-1
K19141n = len(K19141)-1
K09952n = len(K09952)-1
K19137n = len(K19137)-1
K07464n = len(K07464)-1
K02703n = len(K02703)-1
K02706n = len(K02706)-1
```

```
K02705n = len(K02705)-1
K02704n = len(K02704)-1
K02707n = len(K02707)-1
K02708n = len(K02708)-1
K02689n = len(K02689)-1
K02690n = len(K02690)-1
K02691n = len(K02691)-1
K02692n = len(K02692)-1
K02693n = len(K02693)-1
K02694n = len(K02694)-1
K08928n = len(K08928)-1
K08929n = len(K08929)-1
K08940n = len(K08940)-1
K08941n = len(K08941)-1
K08942n = len(K08942)-1
K08943n = len(K08943)-1
K04643n = len(K04643)-1
K04642n = len(K04642)-1
K04641n = len(K04641)-1
K04250n = len(K04250)-1
K00909n = len(K00909)-1
K01428n = len(K01428)-1
K01429n = len(K01429)-1
K01430n = len(K01430)-1
K00111n = len(K00111)-1
K00112n = len(K00112)-1
K00113n = len(K00113)-1
K00096n = len(K00096)-1
K00518n = len(K00518)-1
K04564n = len(K04564)-1
K04565n = len(K04565)-1
K16627n = len(K16627)-1
K06164n = len(K06164)-1
K05780n = len(K05780)-1
K06165n = len(K06165)-1
K06166n = len(K06166)-1
K06163n = len(K06163)-1
```

K08977n = len(K08977)-1
K09836n = len(K09836)-1
K15746n = len(K15746)-1
K16953n = len(K16953)-1
K17486n = len(K17486)-1
K07306n = len(K07306)-1
K17218n = len(K17218)-1
K03553n = len(K03553)-1
K00370n = len(K00370)-1
K00368n = len(K00368)-1
K11959n = len(K11959)-1
K11960n = len(K11960)-1
K11961n = len(K11961)-1
K11962n = len(K11962)-1
K11963n = len(K11963)-1
K02048n = len(K02048)-1
K02046n = len(K02046)-1
K02047n = len(K02047)-1
K02045n = len(K02045)-1
K15576n = len(K15576)-1
K15577n = len(K15577)-1
K15578n = len(K15578)-1
K15579n = len(K15579)-1
K11950n = len(K11950)-1
K11951n = len(K11951)-1
K11952n = len(K11952)-1
K11953n = len(K11953)-1
K15551n = len(K15551)-1
K15552n = len(K15552)-1
K10831n = len(K10831)-1
K15553n = len(K15553)-1
K15554n = len(K15554)-1
K15555n = len(K15555)-1
K11069n = len(K11069)-1
K11070n = len(K11070)-1
K11071n = len(K11071)-1
K11072n = len(K11072)-1

K11073n = len(K11073)-1
K11074n = len(K11074)-1
K11075n = len(K11075)-1
K11076n = len(K11076)-1
K02040n = len(K02040)-1
K02037n = len(K02037)-1
K02038n = len(K02038)-1
K02036n = len(K02036)-1
K02044n = len(K02044)-1
K02042n = len(K02042)-1
K02041n = len(K02041)-1
K11081n = len(K11081)-1
K11082n = len(K11082)-1
K11083n = len(K11083)-1
K11084n = len(K11084)-1
K02002n = len(K02002)-1
K02001n = len(K02001)-1
K02000n = len(K02000)-1
K05845n = len(K05845)-1
K05846n = len(K05846)-1
K05847n = len(K05847)-1
K10108n = len(K10108)-1
K10109n = len(K10109)-1
K10110n = len(K10110)-1
K15770n = len(K15770)-1
K15771n = len(K15771)-1
K15772n = len(K15772)-1
K10117n = len(K10117)-1
K10118n = len(K10118)-1
K10119n = len(K10119)-1
K10232n = len(K10232)-1
K10233n = len(K10233)-1
K10234n = len(K10234)-1
K10235n = len(K10235)-1
K10196n = len(K10196)-1
K10197n = len(K10197)-1
K10198n = len(K10198)-1

```
K10199n = len(K10199)-1
K17315n = len(K17315)-1
K17316n = len(K17316)-1
K17317n = len(K17317)-1
K10236n = len(K10236)-1
K10237n = len(K10237)-1
K10238n = len(K10238)-1
K17311n = len(K17311)-1
K17312n = len(K17312)-1
K17313n = len(K17313)-1
K17314n = len(K17314)-1
K10200n = len(K10200)-1
K10201n = len(K10201)-1
K10202n = len(K10202)-1
K10240n = len(K10240)-1
K10241n = len(K10241)-1
K10242n = len(K10242)-1
K17329n = len(K17329)-1
K17330n = len(K17330)-1
K17331n = len(K17331)-1
K17244n = len(K17244)-1
K17245n = len(K17245)-1
K17246n = len(K17246)-1
K10537n = len(K10537)-1
K10538n = len(K10538)-1
K10539n = len(K10539)-1
K10188n = len(K10188)-1
K10189n = len(K10189)-1
K10190n = len(K10190)-1
K10191n = len(K10191)-1
K10543n = len(K10543)-1
K10544n = len(K10544)-1
K10545n = len(K10545)-1
K17326n = len(K17326)-1
K17327n = len(K17327)-1
K17328n = len(K17328)-1
K10546n = len(K10546)-1
```

K10547n = len(K10547)-1
K10548n = len(K10548)-1
K10552n = len(K10552)-1
K10553n = len(K10553)-1
K10554n = len(K10554)-1
K10559n = len(K10559)-1
K10560n = len(K10560)-1
K10561n = len(K10561)-1
K10562n = len(K10562)-1
K10439n = len(K10439)-1
K10440n = len(K10440)-1
K10441n = len(K10441)-1
K17202n = len(K17202)-1
K17203n = len(K17203)-1
K17204n = len(K17204)-1
K10120n = len(K10120)-1
K10121n = len(K10121)-1
K10122n = len(K10122)-1
K17321n = len(K17321)-1
K17322n = len(K17322)-1
K17323n = len(K17323)-1
K17324n = len(K17324)-1
K17325n = len(K17325)-1
K02027n = len(K02027)-1
K02025n = len(K02025)-1
K02026n = len(K02026)-1
K02058n = len(K02058)-1
K02057n = len(K02057)-1
K02056n = len(K02056)-1
K10013n = len(K10013)-1
K10015n = len(K10015)-1
K10016n = len(K10016)-1
K10017n = len(K10017)-1
K10014n = len(K10014)-1
K10036n = len(K10036)-1
K10037n = len(K10037)-1
K10038n = len(K10038)-1

```
K09996n = len(K09996)-1
K09997n = len(K09997)-1
K09998n = len(K09998)-1
K09999n = len(K09999)-1
K10000n = len(K10000)-1
K10001n = len(K10001)-1
K10002n = len(K10002)-1
K10003n = len(K10003)-1
K10004n = len(K10004)-1
K10039n = len(K10039)-1
K10040n = len(K10040)-1
K10041n = len(K10041)-1
K10018n = len(K10018)-1
K10019n = len(K10019)-1
K10020n = len(K10020)-1
K10021n = len(K10021)-1
K09969n = len(K09969)-1
K09970n = len(K09970)-1
K09971n = len(K09971)-1
K09972n = len(K09972)-1
K10005n = len(K10005)-1
K10006n = len(K10006)-1
K10007n = len(K10007)-1
K10008n = len(K10008)-1
K02424n = len(K02424)-1
K10009n = len(K10009)-1
K10010n = len(K10010)-1
K16956n = len(K16956)-1
K16957n = len(K16957)-1
K16958n = len(K16958)-1
K16959n = len(K16959)-1
K16960n = len(K16960)-1
K10022n = len(K10022)-1
K10023n = len(K10023)-1
K10024n = len(K10024)-1
K10025n = len(K10025)-1
K23059n = len(K23059)-1
```

K17077n = len(K17077)-1
K23060n = len(K23060)-1
K01999n = len(K01999)-1
K01997n = len(K01997)-1
K01998n = len(K01998)-1
K01995n = len(K01995)-1
K01996n = len(K01996)-1
K11954n = len(K11954)-1
K11955n = len(K11955)-1
K11956n = len(K11956)-1
K11957n = len(K11957)-1
K11958n = len(K11958)-1
K02073n = len(K02073)-1
K02072n = len(K02072)-1
K02071n = len(K02071)-1
K15580n = len(K15580)-1
K15581n = len(K15581)-1
K15582n = len(K15582)-1
K15583n = len(K15583)-1
K10823n = len(K10823)-1
K12368n = len(K12368)-1
K12369n = len(K12369)-1
K12370n = len(K12370)-1
K12371n = len(K12371)-1
K12372n = len(K12372)-1
K16199n = len(K16199)-1
K16200n = len(K16200)-1
K16201n = len(K16201)-1
K16202n = len(K16202)-1
K01216n = len(K01216)-1
K01199n = len(K01199)-1
K19891n = len(K19891)-1
K19892n = len(K19892)-1
K19893n = len(K19893)-1
K01190n = len(K01190)-1
K12111n = len(K12111)-1
K12308n = len(K12308)-1

K12309n = len(K12309)-1
K01188n = len(K01188)-1
K05349n = len(K05349)-1
K05350n = len(K05350)-1
K01198n = len(K01198)-1
K15920n = len(K15920)-1
K22268n = len(K22268)-1
K01179n = len(K01179)-1
K19357n = len(K19357)-1
K20542n = len(K20542)-1
K01180n = len(K01180)-1
K20846n = len(K20846)-1
K20850n = len(K20850)-1
K01219n = len(K01219)-1
K20851n = len(K20851)-1
K01200n = len(K01200)-1
K21575n = len(K21575)-1
K01177n = len(K01177)-1
K01208n = len(K01208)-1
K05992n = len(K05992)-1
K22253n = len(K22253)-1
K01178n = len(K01178)-1
K12047n = len(K12047)-1
K21574n = len(K21574)-1
K07024n = len(K07024)-1
K01193n = len(K01193)-1
K00064n = len(K00064)-1
K17993n = len(K17993)-1
K02567n = len(K02567)-1
K03778n = len(K03778)-1
K00955n = len(K00955)-1
K05907n = len(K05907)-1
K17229n = len(K17229)-1
K00958rn = len(K00958r)
K00958on = len(K00958o)
K01225n = len(K01225)-1
K19668n = len(K19668)-1

```
K08688n = len(K08688)-1
K00301n = len(K00301)-1
K00302n = len(K00302)-1
K00303n = len(K00303)-1
K00304n = len(K00304)-1
K00305n = len(K00305)-1
K03851n = len(K03851)-1
K03852n = len(K03852)-1
K01130n = len(K01130)-1
K15923n = len(K15923)-1
K00879n = len(K00879)-1
K01628n = len(K01628)-1
K00848n = len(K00848)-1
K01629n = len(K01629)-1
K01183n = len(K01183)-1
K13381n = len(K13381)-1
K14083n = len(K14083)-1
K16178n = len(K16178)-1
K16176n = len(K16176)-1
K00702n = len(K00702)-1
K16149n = len(K16149)-1
K00975n = len(K00975)-1
K00703n = len(K00703)-1
K16146n = len(K16146)-1
K16147n = len(K16147)-1
K01176n = len(K01176)-1
K05973n = len(K05973)-1
K03430n = len(K03430)-1
K05306n = len(K05306)-1
K11472n = len(K11472)-1
K01941n = len(K01941)-1
othern = len(other)-1
print('KEGG number counts calculated.')

####  Calculating pathway counts
#C cycle
fermentation_n = K00016n + K03778n + np.average([K00169n, K00170n])
```

```
respiration_n = np.average([K02256n, K02262n, K02274n, K02276n])
methanogenesis_n = np.average([K00400n, K00401n])
methane_oxidation_n = np.average([K16157n, K16158n, K16159n, K16161n, K10944mn,
K10945mn, K10946mn])
MoCu_CODH_n = np.average([K03518n, K03519n, K03520n])
rTCA_I_n = np.average([K15230n, K15231n]) ##for GSB mostly
rTCA_II_n = np.average([K15234n, K15233n, K15232n, K00174n, K00175n, K00244n]) ##for
other organisms
WL_n = np.average([K00194n, K00197n])
calvin_cycle_n = np.average([K01602n, K00855n])
carbon_fixation_n = rTCA_I_n + rTCA_II_n + WL_n + calvin_cycle_n
#N cycle
nitrogen_fixation_n = np.average([K02586n, K02591n])
ammonia_assimilation_n = np.average([K01915n, K00264n, K00265n, K00266n, K00284n])
assimilatory_nitrate_reduction_n = np.average([K17877n, K00366n, K00360n, K00367n])
nitrification_n = np.average([K10535n, K10944an, K10945an, K10946an])
anammox_n = np.average([K20932n, K20933n, K20934n, K20935n])
periplasmic_nitrate_reduction_n = K02567n
dissimilatory_nitrate_reduction_n = K00370n
dissimilatory_nitrite_reduction_ammonia_forming_n = K03385n
dissimilatory_nitrite_reduction_NO_forming_n = K00368n
nitric_oxide_reduction_n = np.average([K02305n, K04561n, K00376n])
denitrification_n = dissimilatory_nitrate_reduction_n + periplasmic_nitrate_reduction_n +
dissimilatory_nitrite_reduction_NO_forming_n + nitric_oxide_reduction_n
#S cycle
sox_n = np.average([K17224n, K17227n, K17226n, K17222n, K17223n, K17225n])
cysteine_dioxygenase_n = K00456n
thiosulfate_mercaptopyruvate_sulfurtransferase_n = K01011n
sulfate_reduction_I_n = np.average([K00958rn, K00956n, K00957n, K00955n])
sulfate_reduction_II_n = K00958rn
APS_reduction_I_n = np.average([K00860n, K00955n])
APS_reduction_II_n = np.average([K05907n, K00390n])
APS_reduction_III_n = np.average([K00394rn, K00395rn])
PAPS_reduction_n = K00390n
sulfite_reduction_I_n = np.average([K00380n, K00381n, K00392n])
sulfite_reduction_II_n = np.average([K11180rn, K11181rn])
sulfide_oxidation_n = np.average([K17218n, K17229n])
```

```
sulfur_polysulfide_oxidation_n = np.average([K11180on, K11181on])
sulfite_oxidation_n = np.average([K00394on, K00395on])
APS_oxidation_n = K00958on
assimilatory_sulfate_reduction_I_n = sulfate_reduction_I_n + APS_reduction_I_n +
PAPS_reduction_n + sulfite_reduction_I_n
assimilatory_sulfate_reduction_II_n = sulfate_reduction_I_n + APS_reduction_II_n +
sulfite_reduction_I_n
dissimilatory_sulfate_reduction_n = sulfate_reduction_II_n + APS_reduction_III_n +
sulfite_reduction_II_n
sulfide_oxidation_sulfate_n = sulfide_oxidation_n + sulfur_polysulfide_oxidation_n +
sulfite_oxidation_n + APS_oxidation_n
#Photosystems
photosystem_II_n = np.average([K02703n, K02706n, K02705n, K02704n, K02707n,
K02708n])
photosystem_I_n = np.average([K02689n, K02690n, K02691n, K02692n, K02693n, K02694n])
anoxygenic_photosystem_II_n = np.average([K08928n, K08929n])
anoxygenic_photosystem_I_n = np.average([K08940n, K08941n, K08942n, K08943n])
rhodopsins_n = K04643n + K04642n + K04641n + K04250n + K00909n
astaxanthin_n = np.average([K09836n, K15746n])
bacterioruberin_n = K08977n
#CRISPR
CRISPR_Cas_spacer_acquisition_n = np.average([K15342n, K09951n])
CRISPR_1I_n = np.average([K07012n, K07475n])
CRISPR_1IA_n = np.average([K19088n, K19087n])
CRISPR_1IC_n = K19117n
CRISPR_1IE_n = np.average([K19123n, K19046n])
CRISPR_1IF_n = np.average([K19127n, K19128n, K19129n])
CRISPR_2II_n = K09952n
CRISPR_2IIA_n = K19137n
CRISPR_2IIB_n = K07464n
CRISPR_1III_n = K07016n
CRISPR_1IIIA_n = K19138n
CRISPR_1IIIB_n = K19141n
#hydrogenases
NiFe_hydrogenase_n = np.average([K00437n, K05922n])
NADreducing_hydrogenase_n = K00436n
NADPreducing_hydrogenase_n = K18332n
```

```
Fe_hydrogenase_n = K17997n
ferredoxin_hydrogenase_mono_n = K00532n
ferredoxin_hydrogenase_tri_n = K00533n
membrane_bound_hydrogenase_n = K18016n
methanophenazine_hydrogenase_n = K14068n
coenzyme_F420hydrogenase_n = K00440n
methenyltetrahydromethanopterin_hydrogenase_n = K13942n
F420_nonreducing_hydrogenase_n = K14126n
sulfhydrogenase_n = K17993n
#ABC transporters
Dissimilatory_nitrite_reduction_NO_forming_n = K00368n
Dissimilatory_nitrate_reduction_n = K00370n
Urea_transporter_n = np.average([K11959n, K11960n, K11961n, K11962n, K11963n])
Sulfate_transporter_n = np.average([K02048n, K02046n, K02047n, K02045n])
Nitrate_nitrite_transporter_n = np.average([K15576n, K15577n, K15578n, K15579n])
Bicarbonate_transporter_n = np.average([K11950n, K11951n, K11952n, K11953n])
Taurine_transporter_n = np.average([K15551n, K15552n, K10831n])
Sulfonate_transporter_n = np.average([K15553n, K15554n, K15555n])
Spermidine_putrescine_transporter_n = np.average([K11069n, K11070n, K11071n, K11072n])
Putrescine_transporter_n = np.average([K11073n, K11074n, K11075n, K11076n])
Phosphate_transporter_n = np.average([K02040n, K02037n, K02038n, K02036n])
Phosphonate_transporter_n = np.average([K02044n, K02042n, K02041n])
Aminoethylphosphonate_transporter_n = np.average([K11081n, K11082n, K11083n,
K11084n])
Glycine_betaine_proline_transporter_n = np.average([K02002n, K02001n, K02000n])
Osmoprotectant_transporter_n = np.average([K05845n, K05846n, K05847n])
Maltose_maltodextrin_transporter_n = np.average([K10108n, K10109n, K10110n])
Arabinogalactan_oligomer_maltooligosaccharide_transporter_n = np.average([K15770n,
K15771n, K15772n])
Raffinose_stachyose_melibiose_transporter_n = np.average([K10117n, K10118n, K10119n])
alphaGlucoside_transporter_n = np.average([K10232n, K10233n, K10234n, K10235n])
Glucose_arabinose_transporter_n = np.average([K10196n, K10197n, K10198n, K10199n])
Glucose_mannose_transporter_n = np.average([K17315n, K17316n, K17317n])
Trehalose_maltose_transporter_n = np.average([K10236n, K10237n, K10238n])
Trehalose_transporter_n = np.average([K17311n, K17312n, K17313n, K17314n])
NAcetylglucosamine_transporter_n = np.average([K10200n, K10201n, K10202n])
Cellobiose_transporter_n = np.average([K10240n, K10241n, K10242n])
```

```
NNDiacetylchitobiose_transporter_n = np.average([K17329n, K17330n, K17331n])
Putative_chitobiose_transporter_n = np.average([K17244n, K17245n, K17246n])
LArabinose_transporter_n = np.average([K10537n, K10538n, K10539n])
Lactose_Larabinose_transporter_n = np.average([K10188n, K10189n, K10190n, K10191n])
DXylose_transporter_n = np.average([K10543n, K10544n, K10545n])
Xylobiose_transporter_n = np.average([K17326n, K17327n, K17328n])
Multiple_sugar_transporter_n = np.average([K10546n, K10547n, K10548n])
Fructose_transporter_n = np.average([K10552n, K10553n, K10554n])
Rhamnose_transporter_n = np.average([K10559n, K10560n, K10561n, K10562n])
Ribose_transporter_n = np.average([K10439n, K10440n, K10441n])
Erythritol_transporter_n = np.average([K17202n, K17203n, K17204n])
Putative_fructooligosaccharide_transporter_n = np.average([K10120n, K10121n, K10122n])
Glycerol_transporter_n = np.average([K17321n, K17322n, K17323n, K17324n, K17325n])
Putative_multiple_sugar_transporter_n = np.average([K02027n, K02025n, K02026n])
Putative_simple_sugar_transporter_n = np.average([K02058n, K02057n, K02056n])
Lysine_arginine_ornithine_transporter_n = np.average([K10013n, K10015n, K10016n,
K10017n])
Histidine_transporter_n = np.average([K10014n, K10015n, K10016n, K10017n])
Glutamine_transporter_n = np.average([K10036n, K10037n, K10038n])
Arginine_transporter_n = np.average([K09996n, K09997n, K09998n, K09999n, K10000n])
Glutamate_aspartate_transporter_n = np.average([K10001n, K10002n, K10003n, K10004n])
Aspartate_glutamate_glutamine_transporter_n = np.average([K10039n, K10040n, K10041n])
Octopine_nopaline_transporter_n = np.average([K10018n, K10019n, K10020n, K10021n])
General_Laminoacid_transporter_n = np.average([K09969n, K09970n, K09971n, K09972n])
Glutamate_transporter_n = np.average([K10005n, K10006n, K10007n, K10008n])
Cystine_transporter_n = np.average([K02424n, K10009n, K10010n])
LCystine_transporter_n = np.average([K16956n, K16957n, K16958n, K16959n, K16960n])
Arginine_ornithine_transporter_n = np.average([K10022n, K10023n, K10024n, K10025n])
Arginine_lysine_histidine_transporter_n = np.average([K23059n, K17077n, K23060n])
Branched_chain_aminoacid_transporter_n = np.average([K01999n, K01997n, K01998n,
K01995n, K01996n])
Neutral_aminoacid_transporter_n = np.average([K11954n, K11955n, K11956n, K11957n,
K11958n])
DMethionine_transporter_n = np.average([K02073n, K02072n, K02071n])
Oligopeptide_transporter_n = np.average([K15580n, K15581n, K15582n, K15583n, K10823n])
Dipeptide_transporter_n = np.average([K12368n, K12369n, K12370n, K12371n, K12372n,
K16199n, K16200n, K16201n, K16202n])
```

```
#Carbohydrates
licheninase_n = K01216n
glucan_endobeta_glucosidase_n = np.average([K01199n, K19891n, K19892n, K19893n])
beta_galactosidase_n = np.average([K01190n, K12111n, K12308n, K12309n, K01188n,
K05349n, K05350n])
xylan_beta_xylosidase_n = np.average([K01198n, K15920n, K22268n])
cellulase_endoglucanase_n = np.average([K01179n, K19357n, K20542n])
laminarinase_n = K01180n
carrageenase_n = np.average([K20846n, K20850n])
agarase_n = np.average([K01219n, K20851n])
pullulanase_n = np.average([K01200n, K21575n])
beta_amylase_n = K01177n
maltogenic_alpha_amylase_n = np.average([K01208n, K05992n])
exo_amylase_n = K22253n
glucoamylase_glucan_alpha_glucosidase_n = np.average([K01178n, K12047n, K21574n])
sucrose_phosphatase_n = K07024n
beta_fructofuranosidase_n = K01193n
fucose_utilization_II_n = K00064n
cellobiosidase_n = np.average([K01225n, K19668n])
glycolate_utilization_n = K11472n
creatine_utilization_n = K08688n
sarcosine_utilization_I_n = K00301n
sarcosine_utilization_II_n = np.average([K00302n, K00303n, K00304n, K00305n])
taurine_utilization_n = np.average([K03851n, K03852n])
sulfate_ester_hydrolysis_n = K01130n
fucoidan_degradation_n = K15923n
fucose_utilization_n = np.average([K00879n, K01628n])
rhamnose_utilization_n = np.average([K00848n, K01629n])
chitin_degradation_I_n = K01183n
chitin_degradation_II_n = K13381n
trimethylamine_glycine_betaine_methyltransferase_n = K14083n
dimethylamine_utilization_n = K16178n
monomethylamine_utilization_n = K16176n
cellobiose_utilization_n = K00702n
glycogen_synthesis_overall_n = K16149n
glycogen_synthesis_I_n = np.average([K00975n, K00703n])
glycogen_synthesis_II_n = np.average([K16146n, K16147n])
```

```
starch_degradation_n = K01176n
#Others
PHA_storage_n = np.average([K03821n, K05973n])
urea_n = np.average([K01428n, K01429n, K01430n]) + K01941n
glycerol_n = np.average([K00111n, K00112n, K00113n, K00864n, K00005n])
archaeal_glycerol_n = K00096n
superoxidedismutase_n = np.average([K00518n, K04564n, K04565n, K16627n])
methylphosphonate_catabolism_n = np.average([K06164n, K05780n, K06165n, K06166n,
K06163n])
aminoethylphosphonate_catabolism_n = np.average([K03430n, K05306n])
DMSO_reduction_n = K07306n
DMSP_catabolism_n = K16953n + K17486n
recA_n = K03553n
print('Pathway counts calculated.')

#### Write data to output file
header = ['Fermentation', 'Respiration', 'Methanogenesis', 'Methane oxidation', 'Mo/Cu carbon
monoxide dehydrogenase', 'rTCA I', 'rTCA II', 'WL',
        'Calvin cycle', 'Carbon fixation', 'Nitrogen fixation', 'Ammonia assimilation', 'Assimilatory
nitrate reduction', 'Nitrification', 'Anammox',
        'Periplasmic nitrate reduction', 'Dissimilatory nitrate reduction', 'Dissimilatory nitrite
reduction (ammonia forming)',
        'Dissimilatory nitrite reduction (NO forming)', 'Nitric oxide reduction', 'Denitrification',
        'SOX system', 'Cysteine dioxygenase', 'Thiosulfate/3-mercaptopyruvate sulfurtransferase',
'Sulfate reduction I', 'Sulfate reduction II',
        'APS reduction I', 'APS reduction II', 'APS reduction III', 'PAPS reduction', 'Sulfite
reduction I', 'Sulfite reduction II', 'Sulfide oxidation',
        'Sulfur/polysulfide oxidation', 'Sulfite oxidation', 'APS oxidation', 'Assimilatory sulfate
reduction I', 'Assimilatory sulfate reduction II',
        'Dissimilatory sulfate reduction', 'Sulfide oxidation to sulfate', 'Photosystem II',
'Photosystem I', 'Anoxygenic photosystem II',
        'Anoxygenic photosystem I', 'CRISPR-Cas spacer acquisition', 'CRISPR 1I', 'CRISPR
1IA', 'CRISPR 1IC', 'CRISPR 1IE', 'CRISPR 1IF', 'CRISPR 2II', 'CRISPR 2IIA',
        'CRISPR 2IIB', 'CRISPR 1III', 'CRISPR 1IIIA', 'CRISPR 1IIIB', '[NiFe] hydrogenase',
'NAD-reducing hydrogenase/diaphorase', 'NADP-reducing hydrogenase',
        'Iron-hydrogenase', 'Ferredoxin hydrogenase (monomeric)', 'Ferredoxin hydrogenase
(trimeric)',
```

'Membrane-bound hydrogenase', 'Methanophenazine hydrogenase', 'Coenzyme F420 hydrogenase', '5, 10-Methenyltetrahydromethanopterin hydrogenase',

'F420-non-reducing hydrogenase', 'Sulfhydrogenase', 'Rhodopsins', 'Astaxanthin', 'Bacterioruberin', 'PHA storage', 'Urea catabolism',

'Glycerol catabolism', 'Archaeal glycerol synthesis', 'Superoxidedismutase', 'Methylphosphonate catabolism', 'Aminoethylphosphonate catabolism', 'DMSO reduction', 'DMSP catabolism',

'RecA', 'Urea transporter', 'Sulfate transporter', 'Nitrate/nitrite transporter', 'Bicarbonate transporter', 'Taurine transporter',

'Sulfonate transporter', 'Spermidine/putrescine transporter', 'Putrescine transporter', 'Phosphate transporter', 'Phosphonate transporter',

'2-Aminoethylphosphonate transporter', 'Glycine betaine/proline transporter', 'Osmoprotectant transporter', 'Maltose/maltodextrin transporter',

'Arabinogalactan oligomer/maltooligosaccharide transporter', 'Raffinose/stachyose/melibiose transporter', 'alpha-Glucoside transporter',

'Glucose/arabinose transporter', 'Glucose/mannose transporter', 'Trehalose/maltose transporter', 'Trehalose transporter',

'N-Acetylglucosamine transporter', 'Cellobiose transporter', "N, N'-Diacetylchitobiose transporter", 'Putative chitobiose transporter',

'L-Arabinose transporter', 'Lactose/L-arabinose transporter', 'D-Xylose transporter', 'Xylobiose transporter', 'Multiple sugar transporter',

'Fructose transporter', 'Rhamnose transporter', 'Ribose transporter', 'Erythritol transporter', 'Putative fructooligosaccharide transporter',

'Glycerol transporter', 'Putative multiple sugar transporter', 'Putative simple sugar transporter', 'Lysine/arginine/ornithine transporter',

'Histidine transporter', 'Glutamine transporter', 'Arginine transporter', 'Glutamate/aspartate transporter', 'Aspartate/glutamate/glutamine transporter',

'Octopine/nopaline transporter', 'General L-aminoacid transporter', 'Glutamate transporter', 'Cystine transporter', 'L-Cystine transporter',

'Arginine/ornithine transporter', 'Arginine/lysine/histidine transporter', 'Branched-chain aminoacid transporter', 'Neutral aminoacid transporter',

'D-Methionine transporter', 'Oligopeptide transporter', 'Dipeptide transporter', 'Licheninase', 'Glucan endo-1, 3-beta-glucosidase', 'Beta-galactosidase',

'Xylan 1, 4-beta-xylosidase', 'Cellulase/endoglucanase', 'Laminarinase', 'Carrageenase', 'Agarase', 'Pullulanase', 'Beta-amylase', 'Maltogenic alpha-amylase',

'Exo-amylase', 'Glucoamylase/glucan 1, 4-alpha-glucosidase', 'Sucrose-6-phosphatase', 'Beta-fructofuranosidase', 'Fucose utilization II', 'Cellobiosidase',

'Glycolate utilization', 'Creatine utilization', 'Sarcosine utilization I', 'Sarcosine utilization II', 'Taurine utilization', 'Sulfate ester hydrolysis',
'Fucoidan degradation', 'Fucose utilization', 'Rhamnose utilization', 'Chitin degradation I', 'Chitin degradation II', 'Trimethylamine/glycine betaine methyltransferase',
'Dimethylamine utilization', 'Monomethylamine utilization', 'Cellobiose utilization', 'Glycogen synthesis (overall)', 'Glycogen synthesis I',
'Glycogen synthesis II', 'Starch degradation', 'Other processes', 'Issues', 'Total protein avg fold']

results_c = [fermentation_c, respiration_c, methanogenesis_c, methane_oxidation_c, MoCu_CODH_c, rTCA_I_c, rTCA_II_c, WL_c, calvin_cycle_c,
carbon_fixation_c, nitrogen_fixation_c, ammonia_assimilation_c, assimilatory_nitrate_reduction_c, nitrification_c, anammox_c,
periplasmic_nitrate_reduction_c, dissimilatory_nitrate_reduction_c, dissimilatory_nitrite_reduction_ammonia_forming_c,
dissimilatory_nitrite_reduction_NO_forming_c, nitric_oxide_reduction_c, denitrification_c, sox_c, cysteine_dioxygenase_c,
thiosulfate_mercaptopyruvate_sulfurtransferase_c, sulfate_reduction_I_c, sulfate_reduction_II_c, APS_reduction_I_c, APS_reduction_II_c,
APS_reduction_III_c, PAPS_reduction_c, sulfite_reduction_I_c, sulfite_reduction_II_c, sulfide_oxidation_c, sulfur_polysulfide_oxidation_c,
sulfite_oxidation_c, APS_oxidation_c, assimilatory_sulfate_reduction_I_c, assimilatory_sulfate_reduction_II_c, dissimilatory_sulfate_reduction_c,
sulfide_oxidation_sulfate_c, photosystem_II_c, photosystem_I_c, anoxygenic_photosystem_II_c, anoxygenic_photosystem_I_c, CRISPR_Cas_spacer_acquisition_c,
CRISPR_1I_c, CRISPR_1IA_c, CRISPR_1IC_c, CRISPR_1IE_c, CRISPR_1IF_c, CRISPR_2II_c, CRISPR_2IIA_c, CRISPR_2IIB_c, CRISPR_1III_c, CRISPR_1IIIA_c,
CRISPR_1IIIB_c, NiFe_hydrogenase_c, NADreducing_hydrogenase_c, NADPreducing_hydrogenase_c, Fe_hydrogenase_c, ferredoxin_hydrogenase_mono_c,
ferredoxin_hydrogenase_tri_c, membrane_bound_hydrogenase_c, methanophenazine_hydrogenase_c,
coenzyme_F420hydrogenase_c, methenyltetrahydromethanopterin_hydrogenase_c, F420_nonreducing_hydrogenase_c, sulfhydrogenase_c,
rhodopsins_c, astaxanthin_c, bacterioruberin_c, PHA_storage_c, urea_c, glycerol_c, archaeal_glycerol_c, superoxidedismutase_c,

methylphosphonate_catabolism_c, aminoethylphosphonate_catabolism_c, DMSO_reduction_c, DMSP_catabolism_c, recA_c, Urea_transporter_c, Sulfate_transporter_c, Nitrate_nitrite_transporter_c, Bicarbonate_transporter_c, Taurine_transporter_c, Sulfonate_transporter_c, Spermidine_putrescine_transporter_c, Putrescine_transporter_c, Phosphate_transporter_c, Phosphonate_transporter_c, Aminoethylphosphonate_transporter_c, Glycine_betaine_proline_transporter_c, Osmoprotectant_transporter_c, Maltose_maltodextrin_transporter_c, Arabinogalactan_oligomer_maltooligosaccharide_transporter_c, Raffinose_stachyose_melibiose_transporter_c, alphaGlucoside_transporter_c, Glucose_arabinose_transporter_c, Glucose_mannose_transporter_c, Trehalose_maltose_transporter_c, Trehalose_transporter_c, NAcetylglucosamine_transporter_c, Cellobiose_transporter_c, NNDiacetylchitobiose_transporter_c, Putative_chitobiose_transporter_c, LArabinose_transporter_c, Lactose_Larabinose_transporter_c, DXylose_transporter_c, Xylobiose_transporter_c, Multiple_sugar_transporter_c, Fructose_transporter_c, Rhamnose_transporter_c, Ribose_transporter_c, Erythritol_transporter_c, Putative_fructooligosaccharide_transporter_c, Glycerol_transporter_c, Putative_multiple_sugar_transporter_c, Putative_simple_sugar_transporter_c, Lysine_arginine_ornithine_transporter_c, Histidine_transporter_c, Glutamine_transporter_c, Arginine_transporter_c, Glutamate_aspartate_transporter_c, Aspartate_glutamate_glutamine_transporter_c, Octopine_nopaline_transporter_c, General_Laminoacid_transporter_c, Glutamate_transporter_c, Cystine_transporter_c, LCystine_transporter_c, Arginine_ornithine_transporter_c, Arginine_lysine_histidine_transporter_c, Branched_chain_aminoacid_transporter_c, Neutral_aminoacid_transporter_c, DMethionine_transporter_c, Oligopeptide_transporter_c, Dipeptide_transporter_c, licheninase_c, glucan_endobeta_glucosidase_c, beta_galactosidase_c, xylan_beta_xylosidase_c, cellulase_endoglucanase_c, laminarinase_c, carrageenase_c, agarase_c, pullulanase_c, beta_amylase_c, maltogenic_alpha_amylase_c,

exo_amylase_c, glucoamylase_glucan_alpha_glucosidase_c, sucrose_phosphatase_c, beta_fructofuranosidase_c, fucose_utilization_II_c,
cellobiosidase_c, glycolate_utilization_c, creatine_utilization_c, sarcosine_utilization_I_c, sarcosine_utilization_II_c, taurine_utilization_c,
sulfate_ester_hydrolysis_c, fucoidan_degradation_c, fucose_utilization_c, rhamnose_utilization_c, chitin_degradation_I_c, chitin_degradation_II_c,
trimethylamine_glycine_betaine_methyltransferase_c, dimethylamine_utilization_c, monomethylamine_utilization_c, cellobiose_utilization_c, glycogen_synthesis_overall_c,
glycogen_synthesis_I_c, glycogen_synthesis_II_c, starch_degradation_c, otherc]

results_n = [fermentation_n, respiration_n, methanogenesis_n, methane_oxidation_n, MoCu_CODH_n, rTCA_I_n, rTCA_II_n, WL_n, calvin_cycle_n,
carbon_fixation_n, nitrogen_fixation_n, ammonia_assimilation_n, assimilatory_nitrate_reduction_n, nitrification_n, anammox_n,
periplasmic_nitrate_reduction_n, dissimilatory_nitrate_reduction_n, dissimilatory_nitrite_reduction_ammonia_forming_n,
dissimilatory_nitrite_reduction_NO_forming_n, nitric_oxide_reduction_n, denitrification_n, sox_n, cysteine_dioxygenase_n,
thiosulfate_mercaptopyruvate_sulfurtransferase_n, sulfate_reduction_I_n, sulfate_reduction_II_n, APS_reduction_I_n, APS_reduction_II_n,
APS_reduction_III_n, PAPS_reduction_n, sulfite_reduction_I_n, sulfite_reduction_II_n, sulfide_oxidation_n, sulfur_polysulfide_oxidation_n,
sulfite_oxidation_n, APS_oxidation_n, assimilatory_sulfate_reduction_I_n, assimilatory_sulfate_reduction_II_n, dissimilatory_sulfate_reduction_n,
sulfide_oxidation_sulfate_n, photosystem_II_n, photosystem_I_n, anoxygenic_photosystem_II_n, anoxygenic_photosystem_I_n, CRISPR_Cas_spacer_acquisition_n,
CRISPR_1I_n, CRISPR_1IA_n, CRISPR_1IC_n, CRISPR_1IE_n, CRISPR_1IF_n, CRISPR_2II_n, CRISPR_2IIA_n, CRISPR_2IIB_n, CRISPR_1III_n, CRISPR_1IIIA_n,
CRISPR_1IIIB_n, NiFe_hydrogenase_n, NADreducing_hydrogenase_n, NADPreducing_hydrogenase_n, Fe_hydrogenase_n, ferredoxin_hydrogenase_mono_n,
ferredoxin_hydrogenase_tri_n, membrane_bound_hydrogenase_n, methanophenazine_hydrogenase_n,
coenzyme_F420hydrogenase_n, methenyltetrahydromethanopterin_hydrogenase_n, F420_nonreducing_hydrogenase_n, sulfhydrogenase_n,
rhodopsins_n, astaxanthin_n, bacterioruberin_n, PHA_storage_n, urea_n, glycerol_n, archaeal_glycerol_n, superoxidedismutase_n,

methylphosphonate_catabolism_n, aminoethylphosphonate_catabolism_n, DMSO_reduction_n, DMSP_catabolism_n, recA_n, Urea_transporter_n, Sulfate_transporter_n, Nitrate_nitrite_transporter_n, Bicarbonate_transporter_n, Taurine_transporter_n, Sulfonate_transporter_n, Spermidine_putrescine_transporter_n, Putrescine_transporter_n, Phosphate_transporter_n, Phosphonate_transporter_n, Aminoethylphosphonate_transporter_n, Glycine_betaine_proline_transporter_n, Osmoprotectant_transporter_n, Maltose_maltodextrin_transporter_n, Arabinogalactan_oligomer_maltooligosaccharide_transporter_n, Raffinose_stachyose_melibiose_transporter_n, alphaGlucoside_transporter_n, Glucose_arabinose_transporter_n, Glucose_mannose_transporter_n, Trehalose_maltose_transporter_n, Trehalose_transporter_n, NAcetylglucosamine_transporter_n, Cellobiose_transporter_n, NNDiacetylchitobiose_transporter_n, Putative_chitobiose_transporter_n, LArabinose_transporter_n, Lactose_Larabinose_transporter_n, DXylose_transporter_n, Xylobiose_transporter_n, Multiple_sugar_transporter_n, Fructose_transporter_n, Rhamnose_transporter_n, Ribose_transporter_n, Erythritol_transporter_n, Putative_fructooligosaccharide_transporter_n, Glycerol_transporter_n, Putative_multiple_sugar_transporter_n, Putative_simple_sugar_transporter_n, Lysine_arginine_ornithine_transporter_n, Histidine_transporter_n, Glutamine_transporter_n, Arginine_transporter_n, Glutamate_aspartate_transporter_n, Aspartate_glutamate_glutamine_transporter_n, Octopine_nopaline_transporter_n, General_Laminoacid_transporter_n, Glutamate_transporter_n, Cystine_transporter_n, LCystine_transporter_n, Arginine_ornithine_transporter_n, Arginine_lysine_histidine_transporter_n, Branched_chain_aminoacid_transporter_n, Neutral_aminoacid_transporter_n, DMethionine_transporter_n, Oligopeptide_transporter_n, Dipeptide_transporter_n, licheninase_n, glucan_endobeta_glucosidase_n, beta_galactosidase_n, xylan_beta_xylosidase_n, cellulase_endoglucanase_n, laminarinase_n, carrageenase_n, agarase_n, pullulanase_n, beta_amylase_n, maltogenic_alpha_amylase_n,

```
        exo_amylase_n, glucoamylase_glucan_alpha_glucosidase_n, sucrose_phosphatase_n,
beta_fructofuranosidase_n, fucose_utilization_II_n,
        cellobiosidase_n, glycolate_utilization_n, creatine_utilization_n,
sarcosine_utilization_I_n, sarcosine_utilization_II_n, taurine_utilization_n,
        sulfate_ester_hydrolysis_n, fucoidan_degradation_n, fucose_utilization_n,
rhamnose_utilization_n, chitin_degradation_I_n, chitin_degradation_II_n,
        trimethylamine_glycine_betaine_methyltransferase_n, dimethylamine_utilization_n,
monomethylamine_utilization_n, cellobiose_utilization_n, glycogen_synthesis_overall_n,
        glycogen_synthesis_I_n, glycogen_synthesis_II_n, starch_degradation_n, othern]

all_header = ['K00437', 'K00436', 'K18332', 'K17997', 'K00532', 'K00533', 'K05922', 'K18016',
'K14068', 'K00440', 'K13942', 'K14126', 'K01915', 'K00264', 'K00265', 'K00370', 'K00368',
        'K00266', 'K00284', 'K00864', 'K00005', 'K00169', 'K00170', 'K00016', 'K00174',
'K00175', 'K00244',
        'K00194', 'K00197', 'K00360', 'K00367', 'K20932', 'K20933', 'K20934', 'K20935',
'K00456', 'K01011', 'K00860', 'K00956', 'K00957', 'K19087', 'K19046',
        'K19127', 'K19129', 'K03385', 'K17877', 'K00366', 'K02305', 'K04561', 'K00376',
'K02586', 'K02591', 'K10535',
        'K01602', 'K00855', 'K15230', 'K15231', 'K15234', 'K15233', 'K15232', 'K02256',
'K02262', 'K02274', 'K02276', 'K03520',
        'K03519', 'K03518', 'K00401', 'K00400', 'K16157', 'K16158', 'K16159', 'K16161',
'K00390', 'K00392', 'K00380', 'K00381', 'K17224', 'K17227', 'K17226', 'K17222',
        'K17223', 'K17225', 'K03821', 'K15342', 'K09951', 'K07012', 'K07475', 'K19088',
'K19123', 'K19127', 'K07016', 'K19138', 'K19141', 'K09952', 'K19137',
        'K07464', 'K02703', 'K02706', 'K02705', 'K02704', 'K02707', 'K02708', 'K02689',
'K02690', 'K02691', 'K02692', 'K02693', 'K02694', 'K08928', 'K08929', 'K08940',
        'K08941', 'K08942', 'K08943', 'K04643', 'K04642', 'K04641', 'K04250', 'K00909',
'K01428', 'K01429', 'K01430', 'K00111', 'K00112', 'K00113', 'K00096', 'K00518',
        'K04564', 'K04565', 'K16627', 'K06164', 'K05780', 'K06165', 'K06166', 'K06163',
'K08977', 'K09836', 'K15746', 'K16953', 'K17486', 'K07306',
        'K17218', 'K03553', 'K00394r', 'K00395r', 'K11180r', 'K11181r', 'K00394o', 'K00395o',
'K11180o', 'K11181o', 'K10944a', 'K10945a', 'K10946a', 'K10944m', 'K10945m',
        'K10946m', 'K19117', 'K19128', 'K11959', 'K11960', 'K11961', 'K11962', 'K11963',
'K02048', 'K02046', 'K02047', 'K02045', 'K15576', 'K15577', 'K15578', 'K15579',
        'K11950', 'K11951', 'K11952', 'K11953', 'K15551', 'K15552', 'K10831', 'K15553',
'K15554', 'K15555', 'K11069', 'K11070', 'K11071', 'K11072', 'K11073', 'K11074',
```

'K11075', 'K11076', 'K02040', 'K02037', 'K02038', 'K02036', 'K02044', 'K02042', 'K02041', 'K11081', 'K11082', 'K11083', 'K11084', 'K02002', 'K02001', 'K02000', 'K05845', 'K05846', 'K05847', 'K10108', 'K10109', 'K10110', 'K15770', 'K15771', 'K15772', 'K10117', 'K10118', 'K10119', 'K10232', 'K10233', 'K10234', 'K10235', 'K10196', 'K10197', 'K10198', 'K10199', 'K17315', 'K17316', 'K17317', 'K10236', 'K10237', 'K10238', 'K17311', 'K17312', 'K17313', 'K17314', 'K10200', 'K10201', 'K10202', 'K10240', 'K10241', 'K10242', 'K17329', 'K17330', 'K17331', 'K17244', 'K17245', 'K17246', 'K10537', 'K10538', 'K10539', 'K10188', 'K10189', 'K10190', 'K10191', 'K10543', 'K10544', 'K10545', 'K17326', 'K17327', 'K17328', 'K10546', 'K10547', 'K10548', 'K10552', 'K10553', 'K10554', 'K10559', 'K10560', 'K10561', 'K10562', 'K10439', 'K10440', 'K10441', 'K17202', 'K17203', 'K17204', 'K10120', 'K10121', 'K10122', 'K17321', 'K17322', 'K17323', 'K17324', 'K17325', 'K02027', 'K02025', 'K02026', 'K02058', 'K02057', 'K02056', 'K10013', 'K10015', 'K10016', 'K10017', 'K10014', 'K10036', 'K10037', 'K10038', 'K09996', 'K09997', 'K09998', 'K09999', 'K10000', 'K10001', 'K10002', 'K10003', 'K10004', 'K10039', 'K10040', 'K10041', 'K10018', 'K10019', 'K10020', 'K10021', 'K09969', 'K09970', 'K09971', 'K09972', 'K10005', 'K10006', 'K10007', 'K10008', 'K02424', 'K10009', 'K10010', 'K16956', 'K16957', 'K16958', 'K16959', 'K16960', 'K10022', 'K10023', 'K10024', 'K10025', 'K23059', 'K17077', 'K23060', 'K01999', 'K01997', 'K01998', 'K01995', 'K01996', 'K11954', 'K11955', 'K11956', 'K11957', 'K11958', 'K02073', 'K02072', 'K02071', 'K15580', 'K15581', 'K15582', 'K15583', 'K10823', 'K12368', 'K12369', 'K12370', 'K12371', 'K12372', 'K16199', 'K16200', 'K16201', 'K16202', 'K01216', 'K01199', 'K19891', 'K19892', 'K19893', 'K01190', 'K12111', 'K12308', 'K12309', 'K01188', 'K05349', 'K05350', 'K01198', 'K15920', 'K22268', 'K01179', 'K19357', 'K20542', 'K01180', 'K20846', 'K20850', 'K01219', 'K20851', 'K01200', 'K21575', 'K01177', 'K01208', 'K05992', 'K22253', 'K01178', 'K12047', 'K21574', 'K07024', 'K01193', 'K00064', 'K17993', 'K02567', 'K03778', 'K00955', 'K17229', 'K00958r', 'K00958o', 'K01225', 'K19668', 'K08688', 'K00301', 'K00302', 'K00303', 'K00304', 'K00305', 'K03851', 'K03852', 'K01130', 'K15923', 'K00879', 'K01628', 'K00848', 'K01629', 'K01183', 'K13381', 'K14083', 'K16178', 'K16176', 'K00702', 'K16149', 'K00975', 'K00703', 'K16146', 'K16147', 'K01176', 'K05973', 'K03430', 'K05306', 'K11472', 'K01941']

all_c = [K00437c, K00436c, K18332c, K17997c, K00532c, K00533c, K05922c, K18016c, K14068c, K00440c, K13942c, K14126c, K01915c, K00264c, K00265c, K00370c, K00368c, K00266c, K00284c, K00864c, K00005c, K00169c, K00170c, K00016c, K00174c, K00175c, K00244c,

K00194c, K00197c, K00360c, K00367c, K20932c, K20933c, K20934c, K20935c, K00456c, K01011c, K00860c, K00956c, K00957c, K19087c, K19046c,

K19127c, K19129c, K03385c, K17877c, K00366c, K02305c, K04561c, K00376c, K02586c, K02591c, K10535c,

K01602c, K00855c, K15230c, K15231c, K15234c, K15233c, K15232c, K02256c, K02262c, K02274c, K02276c, K03520c,

K03519c, K03518c, K00401c, K00400c, K16157c, K16158c, K16159c, K16161c, K00390c, K00392c, K00380c, K00381c, K17224c, K17227c, K17226c, K17222c,

K17223c, K17225c, K03821c, K15342c, K09951c, K07012c, K07475c, K19088c, K19123c, K19127c, K07016c, K19138c, K19141c, K09952c, K19137c,

K07464c, K02703c, K02706c, K02705c, K02704c, K02707c, K02708c, K02689c, K02690c, K02691c, K02692c, K02693c, K02694c, K08928c, K08929c, K08940c,

K08941c, K08942c, K08943c, K04643c, K04642c, K04641c, K04250c, K00909c, K01428c, K01429c, K01430c, K00111c, K00112c, K00113c, K00096c, K00518c,

K04564c, K04565c, K16627c, K06164c, K05780c, K06165c, K06166c, K06163c, K08977c, K09836c, K15746c, K16953c, K17486c, K07306c,

K17218c, K03553c, K00394rc, K00395rc, K11180rc, K11181rc, K00394oc, K00395oc, K11180oc, K11181oc, K10944ac, K10945ac, K10946ac, K10944mc, K10945mc,

K10946mc, K19117c, K19128c, K11959c, K11960c, K11961c, K11962c, K11963c, K02048c, K02046c, K02047c, K02045c, K15576c, K15577c, K15578c, K15579c,

K11950c, K11951c, K11952c, K11953c, K15551c, K15552c, K10831c, K15553c, K15554c, K15555c, K11069c, K11070c, K11071c, K11072c, K11073c, K11074c,

K11075c, K11076c, K02040c, K02037c, K02038c, K02036c, K02044c, K02042c, K02041c, K11081c, K11082c, K11083c, K11084c, K02002c, K02001c, K02000c,

K05845c, K05846c, K05847c, K10108c, K10109c, K10110c, K15770c, K15771c, K15772c, K10117c, K10118c, K10119c, K10232c, K10233c, K10234c, K10235c,

K10196c, K10197c, K10198c, K10199c, K17315c, K17316c, K17317c, K10236c, K10237c, K10238c, K17311c, K17312c, K17313c, K17314c, K10200c, K10201c,

K10202c, K10240c, K10241c, K10242c, K17329c, K17330c, K17331c, K17244c, K17245c, K17246c, K10537c, K10538c, K10539c, K10188c, K10189c, K10190c,

K10191c, K10543c, K10544c, K10545c, K17326c, K17327c, K17328c, K10546c, K10547c, K10548c, K10552c, K10553c, K10554c, K10559c, K10560c, K10561c,

K10562c, K10439c, K10440c, K10441c, K17202c, K17203c, K17204c, K10120c, K10121c, K10122c, K17321c, K17322c, K17323c, K17324c, K17325c, K02027c,

K02025c, K02026c, K02058c, K02057c, K02056c, K10013c, K10015c, K10016c, K10017c, K10014c, K10036c, K10037c, K10038c, K09996c, K09997c, K09998c,

K09999c, K10000c, K10001c, K10002c, K10003c, K10004c, K10039c, K10040c, K10041c, K10018c, K10019c, K10020c, K10021c, K09969c, K09970c, K09971c, K09972c, K10005c, K10006c, K10007c, K10008c, K02424c, K10009c, K10010c, K16956c, K16957c, K16958c, K16959c, K16960c, K10022c, K10023c, K10024c, K10025c, K23059c, K17077c, K23060c, K01999c, K01997c, K01998c, K01995c, K01996c, K11954c, K11955c, K11956c, K11957c, K11958c, K02073c, K02072c, K02071c, K15580c, K15581c, K15582c, K15583c, K10823c, K12368c, K12369c, K12370c, K12371c, K12372c, K16199c, K16200c, K16201c, K16202c, K01216c, K01199c, K19891c, K19892c, K19893c, K01190c, K12111c, K12308c, K12309c, K01188c, K05349c, K05350c, K01198c, K15920c, K22268c, K01179c, K19357c, K20542c, K01180c, K20846c, K20850c, K01219c, K20851c, K01200c, K21575c, K01177c, K01208c, K05992c, K22253c, K01178c, K12047c, K21574c, K07024c, K01193c, K00064c, K17993c, K02567c, K03778c, K00955c, K17229c, K00958rc, K00958oc, K01225c, K19668c, K08688c, K00301c, K00302c, K00303c, K00304c, K00305c, K03851c, K03852c, K01130c, K15923c, K00879c, K01628c, K00848c, K01629c, K01183c, K13381c, K14083c, K16178c, K16176c, K00702c, K16149c, K00975c, K00703c, K16146c, K16147c, K01176c, K05973c, K03430c, K05306c, K11472c, K01941c]

all_n = [K00437n, K00436n, K18332n, K17997n, K00532n, K00533n, K05922n, K18016n, K14068n, K00440n, K13942n, K14126n, K01915n, K00264n, K00265n, K00370n, K00368n, K00266n, K00284n, K00864n, K00005n, K00169n, K00170n, K00016n, K00174n, K00175n, K00244n, K00194n, K00197n, K00360n, K00367n, K20932n, K20933n, K20934n, K20935n, K00456n, K01011n, K00860n, K00956n, K00957n, K19087n, K19046n, K19127n, K19129n, K03385n, K17877n, K00366n, K02305n, K04561n, K00376n, K02586n, K02591n, K10535n, K01602n, K00855n, K15230n, K15231n, K15234n, K15233n, K15232n, K02256n, K02262n, K02274n, K02276n, K03520n, K03519n, K03518n, K00401n, K00400n, K16157n, K16158n, K16159n, K16161n, K00390n, K00392n, K00380n, K00381n, K17224n, K17227n, K17226n, K17222n, K17223n, K17225n, K03821n, K15342n, K09951n, K07012n, K07475n, K19088n, K19123n, K19127n, K07016n, K19138n, K19141n, K09952n, K19137n, K07464n, K02703n, K02706n, K02705n, K02704n, K02707n, K02708n, K02689n, K02690n, K02691n, K02692n, K02693n, K02694n, K08928n, K08929n, K08940n, K08941n, K08942n, K08943n, K04643n, K04642n, K04641n, K04250n, K00909n, K01428n, K01429n, K01430n, K00111n, K00112n, K00113n, K00096n, K00518n,

K04564n, K04565n, K16627n, K06164n, K05780n, K06165n, K06166n, K06163n, K08977n, K09836n, K15746n, K16953n, K17486n, K07306n,
K17218n, K03553n, K00394rn, K00395rn, K11180rn, K11181rn, K00394on, K00395on, K11180on, K11181on, K10944an, K10945an, K10946an, K10944mn, K10945mn,
K10946mn, K19117n, K19128n, K11959n, K11960n, K11961n, K11962n, K11963n, K02048n, K02046n, K02047n, K02045n, K15576n, K15577n, K15578n, K15579n,
K11950n, K11951n, K11952n, K11953n, K15551n, K15552n, K10831n, K15553n, K15554n, K15555n, K11069n, K11070n, K11071n, K11072n, K11073n, K11074n,
K11075n, K11076n, K02040n, K02037n, K02038n, K02036n, K02044n, K02042n, K02041n, K11081n, K11082n, K11083n, K11084n, K02002n, K02001n, K02000n,
K05845n, K05846n, K05847n, K10108n, K10109n, K10110n, K15770n, K15771n, K15772n, K10117n, K10118n, K10119n, K10232n, K10233n, K10234n, K10235n,
K10196n, K10197n, K10198n, K10199n, K17315n, K17316n, K17317n, K10236n, K10237n, K10238n, K17311n, K17312n, K17313n, K17314n, K10200n, K10201n,
K10202n, K10240n, K10241n, K10242n, K17329n, K17330n, K17331n, K17244n, K17245n, K17246n, K10537n, K10538n, K10539n, K10188n, K10189n, K10190n,
K10191n, K10543n, K10544n, K10545n, K17326n, K17327n, K17328n, K10546n, K10547n, K10548n, K10552n, K10553n, K10554n, K10559n, K10560n, K10561n,
K10562n, K10439n, K10440n, K10441n, K17202n, K17203n, K17204n, K10120n, K10121n, K10122n, K17321n, K17322n, K17323n, K17324n, K17325n, K02027n,
K02025n, K02026n, K02058n, K02057n, K02056n, K10013n, K10015n, K10016n, K10017n, K10014n, K10036n, K10037n, K10038n, K09996n, K09997n, K09998n,
K09999n, K10000n, K10001n, K10002n, K10003n, K10004n, K10039n, K10040n, K10041n, K10018n, K10019n, K10020n, K10021n, K09969n, K09970n, K09971n,
K09972n, K10005n, K10006n, K10007n, K10008n, K02424n, K10009n, K10010n, K16956n, K16957n, K16958n, K16959n, K16960n, K10022n, K10023n, K10024n,
K10025n, K23059n, K17077n, K23060n, K01999n, K01997n, K01998n, K01995n, K01996n, K11954n, K11955n, K11956n, K11957n, K11958n, K02073n, K02072n,
K02071n, K15580n, K15581n, K15582n, K15583n, K10823n, K12368n, K12369n, K12370n, K12371n, K12372n, K16199n, K16200n, K16201n, K16202n, K01216n,
K01199n, K19891n, K19892n, K19893n, K01190n, K12111n, K12308n, K12309n, K01188n, K05349n, K05350n, K01198n, K15920n, K22268n, K01179n, K19357n,
K20542n, K01180n, K20846n, K20850n, K01219n, K20851n, K01200n, K21575n, K01177n, K01208n, K05992n, K22253n, K01178n, K12047n, K21574n, K07024n,
K01193n, K00064n, K17993n, K02567n, K03778n, K00955n, K17229n, K00958rn, K00958on, K01225n, K19668n, K08688n, K00301n, K00302n, K00303n,

```
        K00304n, K00305n, K03851n, K03852n, K01130n, K15923n, K00879n, K01628n,
K00848n, K01629n, K01183n, K13381n, K14083n, K16178n, K16176n, K00702n,
        K16149n, K00975n, K00703n, K16146n, K16147n, K01176n, K05973n, K03430n,
K05306n, K11472n, K01941n]

with open('%s' + '_KEGG.txt', 'w', newline = '') as outfile:
    outfilec = csv.writer(outfile, delimiter = '\t')
    outfilec.writerow(['Pathway', 'By coverage', 'By count'])
    for i in range(len(header)-2):
        outfilec.writerow([header[i], results_c[i], results_n[i]])
    outfilec.writerow([header[-2], len(no_cov), len(no_cov)])
    outfilec.writerow([header[-1], foldCount_total, directCount_total])
    outfilec.writerow([])
    outfilec.writerow(['KEGG numbers', 'By coverage', 'By count'])
    for i in range(len(all_header)):
        outfilec.writerow([all_header[i],all_c[i],all_n[i]])
print('Finished writing KEGG output to file')
"""%(COV_file, MAP_file, PROTEIN_file, KEGG_file, PROTEIN_file, PROTEIN_file,
PROTEIN_file, PROTEIN_file, PROTEIN_file, PROTEIN_file, PROTEIN_file,
PROTEIN_file, cogkegg_dir + assembly_num)

with open(res_dir + 'KEGG_pathways.py', 'w') as KEGG_script:
    KEGG_script.write(KEGG_py)

# Write bash script
KEGG_bash = """#!/bin/bash
#PBS -N Cavlab-KEGG
#PBS -l select=1:ncpus=1:mem=64gb
#PBS -l walltime=12:00:00
#PBS -j oe
#PBS -o %sKEGG_report
#PBS -M rcavlab@gmail.com
#PBS -m ae

cd %s
module load python/3.8.2
python3 KEGG_pathways.py
```

```
"""%(res_dir, res_dir)

with open(res_dir + 'KEGG.pbs', 'w') as KEGG_pbs:
    KEGG_pbs.write(KEGG_bash)

#### Write the job submission bash script
preprocess_bash = """#!/bin/bash
#PBS -N Cavlab-Launch
#PBS -l select=1:ncpus=1:mem=8gb
#PBS -l walltime=12:00:00
#PBS -j oe
#PBS -o %sJobSubmission_report
#PBS -M rcavlab@gmail.com
#PBS -m ae

cd %s
module load python/3.8.2
python3 append_name2proteins.py
python3 jobsubmission.py
"""%(res_dir, res_dir)

with open(res_dir + 'jobsubmission.pbs', 'w') as preprocess_pbs:
    preprocess_pbs.write(preprocess_bash)

#### Submit first job
command = """cd %s
qsub jobsubmission.pbs"""%(res_dir)
screen = subprocess.check_output(command, shell = True)
screen = screen.decode()[0:6]
with open(res_dir + 'job_log.txt', 'a') as job_log:
    job_csv = csv.writer(job_log, delimiter = '\t')
    job_csv.writerow(['Cavlab pipeline launch', screen])
print('First job submitted.')
```

Appendix D

arCOG pipeline v1.2 — a functional potential analysis pipeline for metagenomes from archaea-rich environments

Code D1. Python code for arCOG pipeline v1.2. The arCOG pipeline was developed for the functional potential analysis of archaea-rich environments. PSI-BLAST was used to match the archaeal protein sequences in a metagenome to a database of arCOG protein sequences available from ftp://ftp.ncbi.nih.gov/pub/wolf/COGs/arCOG. COGsoft was used to assign arCOG numbers to the metagenome proteins (Chapter 2 section 2.2.3.3).

```
'''Original version of arCOG pipeline created on July 9 2017.

@author: Pratibha Panwar

This pipeline should be run from the JGI IMG metagenome folder containing 'IMG_data' and 'QC_and_Genome_Assembly' folders.
Prior to pipeline run, the user needs to create a folder called 'arCOGs' in the JGI IMG metagenome folder and upload two files to it:
1. FASTA file of Archaea protein sequences extracted using Cavlab pipeline MEGAN6 output. This FASTA file must be named as 'Samplename.archaea.fa', where Samplename will vary depending on the sample being assessed. Ensure that the sample name has ONLY letters, numerals, and/or underscore symbol, e.g., Deep_Dec2013_1.archaea.fa.
2. FASTA file containing proteins and their corresponding contig read depths. This file can be found in Cavlab_YYMMDD folders and is also produced through the Cavlab_pipeline. It goes by the extension '.assembled_cov.faa'.

v1.1
Changed the e-value in COGreadblast to 0.001, from 0.1. @Pratibha (14 July, 2017)

v1.2
The requirements for the arCOG pipeline have been changed. Does not require the two FASTA files any longer. Instead, a text file containing Archaea protein IDs, prepared from the Cavlab pipeline MEGAN6 output (RMA file), is required. This text file must be named as 'Samplename.archaea.txt', where Samplename should include lake name, sample collection date, depth, filter fraction, and sample number, e.g., Deep_Dec14_0m_0.8_290.archaea.txt. Ensure that the sample name has ONLY letters, numerals, and/or underscore symbol. A protein to contig read depth mapping file will be created as part of this pipeline.
The abundances are now calculated as absolute abundances and are no longer represented as a fraction of the total abundance of assigned proteins. @Pratibha (2-3 June, 2020)
'''

from datetime import date
import os
import subprocess
import sys
```

```
import csv
import Bio.SeqIO as SeqIO

current_dir = subprocess.check_output('pwd', shell = True).decode().strip() + '/' # get current
directory path

#### Search for resource files and folders
# Verify Archaea protein IDs file
go = []
archaeaPrt = 0
if os.path.isdir('./arCOGs') == True:
    for file in os.listdir('./arCOGs'):
        if file[-12:] == '.archaea.txt':
            archaeaPrt = 1
            Samplename = file.split('.archaea')[0]
            arcPROTEINid_file = file
            break
else:
    print('Error: arCOGs folder not found.')

# Verify other input files
prot, cog, mapf, cov = 0, 0, 0, 0
if os.path.isdir('./IMG_Data') == True:
    for file in os.listdir('./IMG_Data'):
        if file[-13:] == 'assembled.faa': # find protein sequence file
            assembly_num = file.split('.')[0]
            prot = 1
            PROTEIN_file = current_dir + 'IMG_Data/' + file
        if file[-17:] == 'assembled.faa.COG' or file[-13:] == 'assembled.COG': # find COG
annotation file
            cog = 1
            COG_file = current_dir + 'IMG_Data/' + file
        if file[-19:] == 'assembled.names_map': # find scaffold to conig mapping file
            mapf = 1
            MAP_file = current_dir + 'IMG_Data/' + file
        if file[-13:] == 'scaffolds.cov' or file[0:17] == 'seq_coverage_file' or file[-14:] ==
'sorted.bam.cov': # find contig coverage file
```

```
            cov = 1
            COV_file = current_dir + 'IMG_Data/' + file
else:
    print('Error: IMG_data folder not found.')

if archaeaPrt == 0:
    print('Error: Archaea protein ID file not found. Please prepare and upload the file.')
else:
    print('Archaea protein ID file found.')
    go.append(1)
if prot == 0:
    print('Error: Protein sequence file not found.')
else:
    print('Protein sequence file found.')
    go.append(1)
if cog == 0:
    print('Error: COG file not found.')
else:
    print('COG file found.')
    go.append(1)
if mapf == 0:
    print('Error: Scaffold to contig mapping file not found.')
else:
    print('Scaffold to contig mapping file found.')
    go.append(1)
if cov == 0:
    print('Error: Contig coverage file not found.')
else:
    print('Contig coverage file found.')
    go.append(1)

# Verify database files
if os.path.isfile('/srv/scratch/jgi/Cavlab_pipeline_resources/v4/arCOG_conversion_v1.csv') ==
1:
    print('arCOG_conversion_v1.csv file found.')
    go.append(1)
else:
```

```
    print('Error: arCOG_conversion_v1.csv file not found.')

if os.path.isfile('/srv/scratch/jgi/Cavlab_pipeline_resources/v4/ar14_prtseq.fa')==1:
    print('ar14_prtseq.fa file found.')
    go.append(1)
else:
    print('Error: ar14_prtseq.fa file not found.')

if
os.path.isfile('/srv/scratch/jgi/Cavlab_pipeline_resources/v4/ar14.arCOG_domainids.csv')==1:
    print('ar14.arCOG_domainids.csv file found.')
    go.append(1)
else:
    print('Error: ar14.arCOG_domainids.csv file not found.')

if os.path.isfile('/srv/scratch/jgi/Cavlab_pipeline_resources/v4/ar14.arCOG.csv')==1:
    print('ar14.arCOG.csv file found.')
    go.append(1)
else:
    print('Error: ar14.arCOG.csv file not found.')

if os.path.isfile('/srv/scratch/jgi/Cavlab_pipeline_resources/v4/COG_conversion_v2.csv')==1:
    print('COG_conversion_v2.csv file found.')
    go.append(1)
else:
    print('Error: COG_conversion_v2.csv file not found.')

# Check if all input files were found
if sum(go) == 10:
    print('All required resources are present. You may proceed.')
else:
    print('Error: Some files are missing. Script run aborted.')
    sys.exit()

#### Make sub folders
now = date.today()
if now.month < 10:
```

```
    month = '0' + str(now.month)
else:
    month = now.month

if now.day < 10:
    day = '0' + str(now.day)
else:
    day = now.day

head_folder = 'arCOG_v1.2_' + str(now.year)[-2:] + str(month) + str(day)
arcog_dir = current_dir + 'arCOGs/'
head_dir = current_dir + head_folder + '/'

os.rename(arcog_dir, head_dir) # change head folder name from arCOGs to
arCOG_v1.2_YYMMDD
subprocess.call('mkdir ' + head_dir + 'scripts', shell = True)
subprocess.call('mkdir ' + head_dir + 'reports', shell=True)

script_dir = head_dir + 'scripts/'
report_dir = head_dir + 'reports/'

#### Write arCOG readme file
readme_code = '''This is the head folder for the arCOG analysis pipeline v1.2, created on
%s.%s.%s (DDMMYYYY format).
@author:Pratibha Panwar

This script compares the probable archaeal protein sequences against the arCOG protein
sequences, to assign them an arCOG number.
The arCOG number file is analysed using COGsoft, which produces a CSV file containing COG
categories.
The CSV file also has a comparative analysis between COG and arCOG categorization of the
archaeal proteins (classified as Archaea as per MEGAN classification).
It uses the following softwares:
    blast+/2.9.0
    python/3.8.2
    cogsoft/04.19.2012
```

```
It also needs the following input files:
    Samplename.archaea.txt
    ar14.arCOG.csv
    ar14.arCOG_domainids.csv
    ar14_prtseq.fa
    arCOG_conversion_v1.csv
    COG_conversion_v2.csv

For any issues with the pipeline, please contact Pratibha Panwar
(p.panwar@student.unsw.edu.au).
'''%(str(now.day), str(now.month), str(now.year))

with open(head_dir + 'Readme.txt', 'w') as text_file:
    text_file.write(readme_code)

#### Prepare protein to coverage mapping file in head_dir
coverage = {}
with open(COV_file, 'r') as covf:
    covfc = csv.reader(covf, delimiter = '\t')
    next(covfc)
    for row in covfc:
        coverage[row[0]] = row[1]
print('Coverage file read.')

maps = {}
with open(MAP_file, 'r') as mapf:
    mapfc = csv.reader(mapf, delimiter = '\t')
    for row in mapfc:
        maps[row[0]] = row[1]
mapk = list(maps.keys())
print('Contig to scaffold mapping file read.')

covmap = {}
for i in range(len(mapk)):
    covmap[maps[mapk[i]]] = coverage[mapk[i]]
contname_len = len(list(covmap.keys())[0])
print('Contig to coverage mapping complete.')
```

```
prtcov = {}
with open(PROTEIN_file, 'r') as prtf:
        with open(head_dir + assembly_num + '_prt2cov.txt', 'w', newline = '') as outf:
        outfc = csv.writer(outf, delimiter = '\t')
        outfc.writerow(['Protein ID', 'Average fold'])
        for record in SeqIO.parse(prtf, "fasta"):
            outfc.writerow ([record.id, float(covmap[record.id[0:contname_len]])])
print('Protein to coverage mapping complete.')

#### Prepare archaeal protein sequence file
prt_dict = {}
with open(head_dir + arcPROTEINid_file, 'r') as prtID:
    prtIDc = csv.reader(prtID, delimiter = '\t')
    for row in prtIDc:
        proteinID = row[0].split('|')[0]
        prt_dict[proteinID] = ''

prtcount = 0
with open(PROTEIN_file, 'r') as prtf:
    with open(head_dir + Samplename + '.archaea.faa', 'w') as prtseq:
        for rec in SeqIO.parse(prtf, 'fasta'):
            if rec.id in prt_dict.keys():
                SeqIO.write(rec, prtseq, 'fasta')
                prtcount += 1
            else:
                continue
if len(prt_dict) != prtcount:
    print('ERROR: did not find protein sequence for all archaeal protein IDs.')
    sys.exit()
else:
    arcPROTEIN_file = Samplename + '.archaea.faa'
    print('Archaeal protein sequence file prepared.')

#### Prepare query file and extract proteinIDs
fileprep_code = """import csv
import Bio.SeqIO as SeqIO
```

```
#### Prepare archaea protein file without product names in headers
with open('%s', 'r') as read_file:
    with open('%s_query.fa', 'w') as newfile:
        for record in SeqIO.parse(read_file, "fasta"):
            record.id = record.id.split('|')[0]
            record.description = record.id # for cases where record id and description are different
            SeqIO.write(record, newfile, 'fasta')

#### Create protein ID list
proteinID = []
with open('%s_query.fa', 'r') as read_file:
    for record in SeqIO.parse(read_file, "fasta"):
        proteinID.append(record.id)

with open('%s.p2o.csv', 'w') as out_file:
    out_csv = csv.writer(out_file)
    for i in range(len(proteinID)):
        out_csv.writerow([proteinID[i], '%s'])

#### Submit job to begin COGsoft run
import subprocess

command = 'qsub ' + '%spsiblast1.pbs'
subprocess.check_output(command, shell = True)
""" %(head_dir + arcPROTEIN_file, head_dir + assembly_num, head_dir + assembly_num,
head_dir + assembly_num, Samplename, script_dir)

with open(script_dir + 'Fileprep.py', 'w') as fileprep:
    fileprep.write(fileprep_code)
Query_file = head_dir + assembly_num + '_query.fa'

#### Write Psi-blast 1 bash script
psiblast1_script = '''#!/bin/bash
#PBS -N arCOG-Psiblast1
#PBS -l select=1:ncpus=16:mem=96gb
#PBS -l walltime=48:00:00
```

```
#PBS -j oe
#PBS -o %spsiblast1_Report
#PBS -M rcavlab@gmail.com
#PBS -m ae

cd %s
module load blast+/2.9.0
makeblastdb -in %s -dbtype prot -out Querydb
makeblastdb -in /srv/scratch/jgi/Cavlab_pipeline_resources/v4/ar14_prtseq.fa -dbtype prot -out
arCOGdb

mkdir BLASTff
psiblast -query %s -db arCOGdb -show_gis -outfmt 7 -max_target_seqs 1000 -dbsize
100000000 -comp_based_stats T -seg yes -out ./BLASTff/QueryarCOGs.tab -num_threads 16

qsub %spsiblast2.pbs
qsub %spsiblast3.pbs
'''%(report_dir, head_dir, Query_file, Query_file, script_dir, script_dir)

with open(script_dir + 'psiblast1.pbs', 'w') as pblast1:
    pblast1.write(psiblast1_script)

#### Write Psi-blast 2 bash script
psiblast2_script = '''#!/bin/bash
#PBS -N arCOG-Psiblast2
#PBS -l select=1:ncpus=16:mem=96gb
#PBS -l walltime=48:00:00
#PBS -j oe
#PBS -o %spsiblast2_Report
#PBS -M rcavlab@gmail.com
#PBS -m ae

cd %s
module load blast+/2.9.0
mkdir BLASTss
psiblast -query %s -db Querydb -show_gis -outfmt 7 -max_target_seqs 10 -dbsize 100000000 -
comp_based_stats F -seg no -out ./BLASTss/QuerySelf.tab -num_threads 16
```

```
'''%(report_dir, head_dir, Query_file)

with open(script_dir + 'psiblast2.pbs', 'w') as pblast2:
    pblast2.write(psiblast2_script)

#### Write Psi-blast 3 bash script
psiblast3_script = '''#!/bin/bash
#PBS -N arCOG-Psiblast3
#PBS -l select=1:ncpus=16:mem=96gb
#PBS -l walltime=48:00:00
#PBS -j oe
#PBS -o %spsiblast3_Report
#PBS -M rcavlab@gmail.com
#PBS -m ae

cd %s
module load blast+/2.9.0
mkdir BLASTno
psiblast -query %s -db arCOGdb -show_gis -outfmt 7 -max_target_seqs 1000 -dbsize
100000000 -comp_based_stats F -seg no -out ./BLASTno/QueryarCOGs.tab -num_threads 16
qsub %scogsoft.pbs
'''%(report_dir, head_dir, Query_file, script_dir)

with open(script_dir + 'psiblast3.pbs', 'w') as pblast3:
    pblast3.write(psiblast3_script)

#### Write Cogsoft bash script
cogsoft_script = '''#!/bin/bash
#PBS -N arCOG-Cogsoft
#PBS -l select=1:ncpus=1:mem=8gb
#PBS -l walltime=12:00:00
#PBS -j oe
#PBS -o %scogsoft_Report
#PBS -M rcavlab@gmail.com
#PBS -m ae

cd %s
```

```
module load cogsoft/04.19.2012
cat %s.p2o.csv /srv/scratch/jgi/Cavlab_pipeline_resources/v4/ar14.arCOG_domainids.csv >
tmp.p2o.csv

mkdir BLASTcogn
COGmakehash -i=tmp.p2o.csv -o=./BLASTcogn -s="," -n=1
COGreadblast -d=./BLASTcogn -u=./BLASTno -f=./BLASTff -s=./BLASTss -e=0.001 -q=2 -
t=2
COGcognitor -i=./BLASTcogn -
t=/srv/scratch/jgi/Cavlab_pipeline_resources/v4/ar14.arCOG.csv -q=%s.p2o.csv -
o=%s.arCOG.txt

module load python/3.8.2
python3 %spostcogsoft.py
'''%(report_dir, head_dir, head_dir + assembly_num, head_dir + assembly_num, head_dir +
Samplename, script_dir)

with open(script_dir + 'cogsoft.pbs','w') as cogsoft:
    cogsoft.write(cogsoft_script)

#### Extracting arCOGs from cognitor output
postcogsoft_code = '''import csv
import Bio.SeqIO as SeqIO

print('Post-COGsoft steps running.')
#### Read protein coverages
coverage = {}
with open('%s' + '_prt2cov.txt', 'r') as covf:
    covfc = csv.reader(covf, delimiter = '\t')
    next(covfc)
    for row in covfc:
        coverage[row[0]] = row[1]
print('Protein coverages read.')

#### Creating arCOG conversion dictionary
```

```
reader =
csv.reader(open('/srv/scratch/jgi/Cavlab_pipeline_resources/v4/arCOG_conversion_v1.csv',
'r'))
d = {}
for row in reader:
  k, v = row
  d[k] = v

#### Creating COG conversion dictionary
reader =
csv.reader(open('/srv/scratch/jgi/Cavlab_pipeline_resources/v4/COG_conversion_v2.csv', 'r'))
e = {}
for row in reader:
  k, v = row
  e[k] = v
print('arCOG and COG conversion dictionaries prepared.')

#### Initialising arCOG and COG number based COG categories
arA, A = [0], [0]
arB, B = [0], [0]
arC, C = [0], [0]
arD, D = [0], [0]
arE, E = [0], [0]
arF, F = [0], [0]
arG, G = [0], [0]
arH, H = [0], [0]
arI, I = [0], [0]
arJ, J = [0], [0]
arK, K = [0], [0]
arL, L = [0], [0]
arM, M = [0], [0]
arN, N = [0], [0]
arO, O = [0], [0]
arP, P = [0], [0]
arQ, Q = [0], [0]
arR, R = [0], [0]
arS, S = [0], [0]
```

```
arT, T = [0], [0]
arU, U = [0], [0]
arV, V = [0], [0]
arW, W = [0], [0]
arX, X = [0], [0]
arY, Y = [0], [0]
arZ, Z = [0], [0]
arOther, Other = [], []

#### Reading data from cogsoft output arCOG file
with open('%s.arCOG.txt', 'r') as cogsoftf:
    cogsoftfc = csv.reader(cogsoftf, delimiter=',')
    arcogs = []
    for row in cogsoftfc:
        if row[5][0:5] == 'arCOG':
            if row[0] in coverage.keys():
                arcogs.append([row[5], float(coverage[row[0]]), row[0]])
            else:
                arOther.append(row[5])
        else:
            continue
print('COGsoft output file read.')

#### Categorizing arCOGs
for j in range(len(arcogs)):
    arCOGCat = d[arcogs[j][0]]
    if arCOGCat == 'A':
        arA.append(arcogs[j][1])
    elif arCOGCat == 'B':
        arB.append(arcogs[j][1])
    elif arCOGCat == 'C':
        arC.append(arcogs[j][1])
    elif arCOGCat == 'D':
        arD.append(arcogs[j][1])
    elif arCOGCat == 'E':
        arE.append(arcogs[j][1])
    elif arCOGCat == 'F':
```

```
        arF.append(arcogs[j][1])
    elif arCOGCat == 'G':
        arG.append(arcogs[j][1])
    elif arCOGCat == 'H':
        arH.append(arcogs[j][1])
    elif arCOGCat == 'I':
        arI.append(arcogs[j][1])
    elif arCOGCat == 'J':
        arJ.append(arcogs[j][1])
    elif arCOGCat == 'K':
        arK.append(arcogs[j][1])
    elif arCOGCat == 'L':
        arL.append(arcogs[j][1])
    elif arCOGCat == 'M':
        arM.append(arcogs[j][1])
    elif arCOGCat == 'N':
        arN.append(arcogs[j][1])
    elif arCOGCat == 'O':
        arO.append(arcogs[j][1])
    elif arCOGCat == 'P':
        arP.append(arcogs[j][1])
    elif arCOGCat == 'Q':
        arQ.append(arcogs[j][1])
    elif arCOGCat == ‘r’:
        arR.append(arcogs[j][1])
    elif arCOGCat == 'S':
        arS.append(arcogs[j][1])
    elif arCOGCat == 'T':
        arT.append(arcogs[j][1])
    elif arCOGCat == 'U':
        arU.append(arcogs[j][1])
    elif arCOGCat == 'V':
        arV.append(arcogs[j][1])
    elif arCOGCat == ‘w’:
        arW.append(arcogs[j][1])
    elif arCOGCat == 'X':
        arX.append(arcogs[j][1])
```

```
    elif arCOGCat == 'Y':
        arY.append(arcogs[j][1])
    elif arCOGCat == 'Z':
        arZ.append(arcogs[j][1])
    else:
        arOther.append(arCOGCat)
print('arCOG numbers categorised.')

#### arCOG info by coverage
arAc = sum(arA)
arBc = sum(arB)
arCc = sum(arC)
arDc = sum(arD)
arEc = sum(arE)
arFc = sum(arF)
arGc = sum(arG)
arHc = sum(arH)
arIc = sum(arI)
arJc = sum(arJ)
arKc = sum(arK)
arLc = sum(arL)
arMc = sum(arM)
arNc = sum(arN)
arOc = sum(arO)
arPc = sum(arP)
arQc = sum(arQ)
arRc = sum(arR)
arSc = sum(arS)
arTc = sum(arT)
arUc = sum(arU)
arVc = sum(arV)
arWc = sum(arW)
arXc = sum(arX)
arYc = sum(arY)
arZc = sum(arZ)
print('arCOG category coverage-based abundances calculated.')
```

```
#### arCOG info by count
arAn = (len(arA)-1)
arBn = (len(arB)-1)
arCn = (len(arC)-1)
arDn = (len(arD)-1)
arEn = (len(arE)-1)
arFn = (len(arF)-1)
arGn = (len(arG)-1)
arHn = (len(arH)-1)
arIn = (len(arI)-1)
arJn = (len(arJ)-1)
arKn = (len(arK)-1)
arLn = (len(arL)-1)
arMn = (len(arM)-1)
arNn = (len(arN)-1)
arOn = (len(arO)-1)
arPn = (len(arP)-1)
arQn = (len(arQ)-1)
arRn = (len(arR)-1)
arSn = (len(arS)-1)
arTn = (len(arT)-1)
arUn = (len(arU)-1)
arVn = (len(arV)-1)
arWn = (len(arW)-1)
arXn = (len(arX)-1)
arYn = (len(arY)-1)
arZn = (len(arZ)-1)
print('arCOG category count-based abundances calculated.')

#### Reading data from COG file
cogsdict = {}
with open('%s', 'r') as read_file:
    COGs_csv = csv.reader(read_file, delimiter = '\t')
    for row in COGs_csv:
        cogsdict.setdefault(row[0], []).append(row[1])
print('COG file read.')
```

```
cogs = []
for i in range(len(arcogs)):
    if arcogs[i][2] in list(cogsdict.keys()):
        for value in cogsdict[arcogs[i][2]]:
            cogs.append([value, float(coverage[arcogs[i][2]])])
    else:
        continue
print('COG file data compared to COGsoft output.')

#### Categorising cogs
for j in range(len(cogs)):
    COGCat = e[cogs[j][0]]
    if COGCat == 'A':
        A.append(cogs[j][1])
    elif COGCat == 'B':
        B.append(cogs[j][1])
    elif COGCat == 'C':
        C.append(cogs[j][1])
    elif COGCat == 'D':
        D.append(cogs[j][1])
    elif COGCat == 'E':
        E.append(cogs[j][1])
    elif COGCat == 'F':
        F.append(cogs[j][1])
    elif COGCat == 'G':
        G.append(cogs[j][1])
    elif COGCat == 'H':
        H.append(cogs[j][1])
    elif COGCat == 'I':
        I.append(cogs[j][1])
    elif COGCat == 'J':
        J.append(cogs[j][1])
    elif COGCat == 'K':
        K.append(cogs[j][1])
    elif COGCat == 'L':
        L.append(cogs[j][1])
    elif COGCat == 'M':
```

```
        M.append(cogs[j][1])
    elif COGCat == 'N':
        N.append(cogs[j][1])
    elif COGCat == 'O':
        O.append(cogs[j][1])
    elif COGCat == 'P':
        P.append(cogs[j][1])
    elif COGCat == 'Q':
        Q.append(cogs[j][1])
    elif COGCat == ‘r’:
        R.append(cogs[j][1])
    elif COGCat == 'S':
        S.append(cogs[j][1])
    elif COGCat == 'T':
        T.append(cogs[j][1])
    elif COGCat == 'U':
        U.append(cogs[j][1])
    elif COGCat == 'V':
        V.append(cogs[j][1])
    elif COGCat == ‘w’:
        W.append(cogs[j][1])
    elif COGCat == 'X':
        X.append(cogs[j][1])
    elif COGCat == 'Y':
        Y.append(cogs[j][1])
    elif COGCat == 'Z':
        Z.append(cogs[j][1])
    else:
        Other.append(COGCat)
print('COG numbers categorised.')

#### COG info by coverage
Ac = sum(A)
Bc = sum(B)
Cc = sum(C)
Dc = sum(D)
Ec = sum(E)
```

```
Fc = sum(F)
Gc = sum(G)
Hc = sum(H)
Ic = sum(I)
Jc = sum(J)
Kc = sum(K)
Lc = sum(L)
Mc = sum(M)
Nc = sum(N)
Oc = sum(O)
Pc = sum(P)
Qc = sum(Q)
Rc = sum(R)
Sc = sum(S)
Tc = sum(T)
Uc = sum(U)
Vc = sum(V)
Wc = sum(W)
Xc = sum(X)
Yc = sum(Y)
Zc = sum(Z)
print('COG category coverage-based abundances calculated.')

#### COG info by count
An = (len(A)-1)
Bn = (len(B)-1)
Cn = (len(C)-1)
Dn = (len(D)-1)
En = (len(E)-1)
Fn = (len(F)-1)
Gn = (len(G)-1)
Hn = (len(H)-1)
In = (len(I)-1)
Jn = (len(J)-1)
Kn = (len(K)-1)
Ln = (len(L)-1)
Mn = (len(M)-1)
```

```
Nn = (len(N)-1)
On = (len(O)-1)
Pn = (len(P)-1)
Qn = (len(Q)-1)
Rn = (len(R)-1)
Sn = (len(S)-1)
Tn = (len(T)-1)
Un = (len(U)-1)
Vn = (len(V)-1)
Wn = (len(W)-1)
Xn = (len(X)-1)
Yn = (len(Y)-1)
Zn = (len(Z)-1)
print('COG category count-based abundances calculated.')

#### Writing data to csv file
arCOGresults_c = [arAc, arBc, arCc, arDc, arEc, arFc, arGc, arHc, arIc, arJc, arKc, arLc, arMc,
arNc, arOc, arPc, arQc, arRc, arSc, arTc, arUc, arVc, arWc, arXc, arYc, arZc]
arCOGresults_n = [arAn, arBn, arCn, arDn, arEn, arFn, arGn, arHn, arIn, arJn, arKn, arLn,
arMn, arNn, arOn, arPn, arQn, arRn, arSn, arTn, arUn, arVn, arWn, arXn, arYn, arZn]
COGresults_c = [Ac, Bc, Cc, Dc, Ec, Fc, Gc, Hc, Ic, Jc, Kc, Lc, Mc, Nc, Oc, Pc, Qc, Rc, Sc, Tc,
Uc, Vc, Wc, Xc, Yc, Zc]
COGresults_n = [An, Bn, Cn, Dn, En, Fn, Gn, Hn, In, Jn, Kn, Ln, Mn, Nn, On, Pn, Qn, Rn, Sn,
Tn, Un, Vn, Wn, Xn, Yn, Zn]
header = ['(A) RNA processing and modification', '(B) Chromatin structure and dynamics', '(C)
Energy production and conversion',
'(D) Cell cycle control,  cell division,  chromosome partitioning', '(E) Amino acid transport and
metabolism', '(F) Nucleotide transport and metabolism',
'(G) Carbohydrate transport and metabolism', '(H) Coenzyme transport and metabolism', '(I)
Lipid transport and metabolism',
'(J) Translation,  ribosomal structure and biogenesis', '(K) Transcription', '(L) Replication,
recombination and repair',
'(M) Cell wall/membrane/envelope biogenesis', '(N) Cell motility', '(O) Post-translational
modification,  protein turnover,  and chaperones',
'(P) Inorganic ion transport and metabolism', '(Q) Secondary metabolites biosynthesis,
transport,  and catabolism', '(R) General function prediction only',
```

```
'(S) Function unknown', '(T) Signal transduction mechanisms', '(U) Intracellular trafficking, secretion,  and vesicular transport', '(V) Defense mechanisms',
'(W) Extracellular structures', '(X) Mobilome: prophages,  transposons', '(Y) Nuclear structure', '(Z) Cytoskeleton', 'Issues']

with open('%s.arCOG_summary.csv', 'w') as out_file:
    out_csv=csv.writer(out_file)
    out_csv.writerow(['Category', 'arCOG by count', 'COG by count', 'arCOG by coverage', 'COG by coverage'])
    for i in range(len(header)-1):
        out_csv.writerow([header[i], arCOGresults_n[i], COGresults_n[i], arCOGresults_c[i], COGresults_c[i]])
    out_csv.writerow([header[-1], len(arOther), len(Other), len(arOther), len(Other)])
print('arCOG and COG data written to output file.')
'''%(head_dir + assembly_num, head_dir + Samplename, COG_file, head_dir + Samplename)

with open(script_dir + 'postcogsoft.py', 'w') as post_cogsoft:
    post_cogsoft.write(postcogsoft_code)

#### Write preprocess bash
fileprep_script = """#!/bin/bash
#PBS -N arCOG-Launch
#PBS -l select=1:ncpus=1:mem=8gb
#PBS -l walltime=12:00:00
#PBS -j oe
#PBS -o %sfileprep_Report
#PBS -M rcavlab@gmail.com
#PBS -m ae

cd %s
module load python/3.8.2
python3 Fileprep.py
"""%(report_dir, script_dir)

with open(script_dir + 'fileprep.pbs', 'w') as fileprep_pbs:
    fileprep_pbs.write(fileprep_script)
```

```
#### Submit first job
command = 'qsub ' + script_dir + 'fileprep.pbs'
subprocess.check_output(command, shell = True)
```

Appendix E

Clade-specific markers added to MetaPhlAn2 database

Table E1. List of clade-specific markers added to MetaPhlAn2 database. The table includes the gene markers for specific microbes that were added to the MetaPhlAn2 database. The specific microbes were selected based on the taxonomic output of DIAMOND/MEGAN6 runs (Chapter 2 section 2.2.2.1) on Ace Lake and Deep Lake Megahit-assembled metagenomes (Appendix A: Table A1). The methodology for the addition of these markers to the MetaPhlAn2 database is discussed in Chapter 2 section 2.2.2.3. *k, kingdom; p, phylum; c, class; o, order; f, family; g, genus; s, species; t, strain.

Clade (genome length)	Taxonomy*	Marker IDs (sequence length)
Halohasta litchfieldiae tADL (3332020 bp)	k__Archaea p__Euryarchaeota c__Halobacteria o__Haloferacales f__Halorubraceae g__Halohasta s__Halohasta_litchfieldiae_tADL t__GCF_900109065	gi|1279136099|ref|CP024845.1|:42441-40969 (1473 bp) gi|645321082|ref|NR_118135.1|:1-1473 (1473 bp)
Aureococcus anophagefferens (56660600 bp)	k__Eukaryota p__Eukaryota_noname c__Pelagophyceae o__Pelagomonadales f__Pelagomonadaceae g__Aureococcus| s__Aureococcus_anophagefferens t__GCF_000186865	gi|984294609|gb|KT390070.1|:1-1455 (1455 bp) gi|984294576|gb|KT390037.1|:1-1451 (1451 bp) gi|676392214|ref|XM_009041137.1|:1-1179 (1179 bp) gi|676393690|ref|XM_009041875.1|:1-447 (447 bp) gi|676393158|ref|XM_009041609.1|:1-642 (642 bp) gi|676379860|ref|XM_009034961.1|:1-767 (767 bp)
Candidatus Methanoperedens nitroreducens (3203390 bp)	k__Archaea p__Euryarchaeota c__Methanomicrobia o__Methanosarcinales f__Candidatus_Methanoperedenaceae g__Candidatus_Methanoperedens	GeneID:2618674875 (1471 bp) GeneID:2515322110 (1295 bp)

	s__Candidatus_Methanoperedens_nitroreducens t__GCF_000685155	
Candidatus Omnitrophus magneticus (3145900 bp)	k__Bacteria p__Candidatus_Omnitrophica c__Candidatus_Omnitrophica_nonam e o__Candidatus_Omnitrophica_nonam e f__Candidatus_Omnitrophica_nonam e g__Candidatus_Omnitrophus|s__Cand idatus_Omnitrophus_magneticus t__GCA_000954095	GeneID:2639715750 (1567 bp)
Chlamydomonas reinhardtii (120190000 bp)	k__Eukaryota p__Chlorophyta c__Chlorophyceae o__Chlamydomonadales f__Chlamydomonadaceae g__Chlamydomonas s__Chlamydomonas_reinhardtii t__GCF_000002595	gi|164665428|gb|EF682842.2|:1-556 (556 bp) gi|449331407|gb|KC166137.1|:1-1480 (1480 bp) gi|167643754|gb|EU410820.1|:1-148 (148 bp)
Chlorella variabilis (46159500 bp)	k__Eukaryota p__Chlorophyta c__Trebouxiophyceae o__Chlorellales f__Chlorellaceae g__Chlorella s__Chlorella_variabilis t__GCF_000147415	gi|577858493|gb|KF887350.1|:1-752 (752 bp) gi|552847757|ref|XM_005851879.1|:1-1161 (1161 bp) gi|552846485|ref|XM_005851574.1|:1-1768 (1768 bp) gi|552833545|ref|XM_005848492.1|:1-1497 (1497 bp) gi|552831571|ref|XM_005848103.1|:1-213 (213 bp) gi|552828230|ref|XM_005847371.1|:1-567 (567 bp)
Coccomyxa subellipsoidea	k__Eukaryota p__Chlorophyta	gi|864421734|gb|HG972975.1|:1-3209 (3209 bp)

(48826600 bp)	c__Trebouxiophyceae o__Trebouxiophyceae_noname f__Coccomyxaceae g__Coccomyxa s__Coccomyxa_subellipsoidea t__GCF_000258705	gi|545375863|ref|XM_005652011.1|:1-949 (949 bp) gi|545363864|ref|XM_005646970.1|:1-1997 (1997 bp)
Dunaliella salina (343700000 bp)	k__Eukaryota p__Chlorophyta c__Chlorophyceae o__Chlamydomonadales f__Dunaliellaceae g__Dunaliella s__Dunaliella_salina t__GCA_002284615	gi|167989|gb|M84320.1|:21-2162 (2142 bp) gi|559767353|gb|KF573420.1|:1-253 (253 bp) gi|71482598|gb|DQ116743.1|:1-352 (352 bp) gi|700653926|gb|KF825552.1|:1-2068 (2068 bp) gi|63029920|gb|DQ009777.1|:1-2088 (2088 bp) gi|699257993|gb|KM211532.1|:1-654 (654 bp) gi|662257977|gb|JN807321.2|:1-1575 (1575 bp) gi|338163258|gb|JF900404.1|:1-2117 (2117 bp)
Ectocarpus siliculosus (195811000 bp)	k__Eukaryota p__Eukaryota_noname c__Phaeophyceae o__Ectocarpales f__Ectocarpaceae g__Ectocarpus s__Ectocarpus_siliculosus t__GCA_000310025	gi|1145575|gb|U38758.1|:1-1141 (1141 bp) gi|291191871|gb|GQ351370.1|:1-798 (798 bp) gi|1145611|gb|U38832.1|:1-507 (507 bp) gi|29120014|gb|AJ550048.1|:1-660 (660 bp) gi|301599175|gb|FR668885.1|:1-1832 (1832 bp) gi|29120031|gb|AJ550056.1|:1-867 (867 bp)
Gonium pectorale (148806000 bp)	k__Eukaryota p__Chlorophyta	gi|1043351839|gb|KX247730.1|:1-676 (676 bp)

	c__Chlorophyceae o__Chlamydomonadales f__Goniaceae g__Gonium s__Gonium_pectorale t__GCF_001584585	gi|1043351838|gb|KX247729.1|:1-671 (671 bp) gi|1043351835|gb|KX247726.1|:1-669 (669 bp)
Guillardia theta (87145300 bp)	k__Eukaryota p__Cryptophyta c__Cryptophyceae o__Pyrenomonadales f__Geminigeraceae g__Guillardia s__Guillardia_theta t__GCF_000315625	GeneID:638270151 (2040 bp)
Halapricum salinum (3451490 bp)	k__Archaea p__Euryarchaeota c__Halobacteria o__Halobacteriales f__Haloarculaceae g__Halapricum s__Halapricum_salinum t__GCF_000755225	gi|699005439|ref|NR_126308.1|:1-1472 (1472 bp) gi|699005438|ref|NR_126307.1|:1-1472 (1472 bp) gi|699005424|ref|NR_126293.1|:1-1472 (1472 bp)
Haloarchaeobius iranensis (3768606 bp)	k__Archaea p__Euryarchaeota c__Halobacteria o__Halobacteriales f__Halobacteriaceae g__Haloarchaeobius s__Haloarchaeobius_iranensis t__GCF_900103505	GeneID:2653873010 (1472 bp) GeneID:2653872430 (916 bp)
Halobacterium jilantaiense (2952790 bp)	k__Archaea p__Euryarchaeota c__Halobacteria o__Halobacteriales f__Halobacteriaceae g__Halobacterium	GeneID:2618015680 (1473 bp) gi|343203496|ref|NR_043676.1|:1-(1396 bp)

	s__Halobacterium_jilantaiense t__GCF_900110535	
Halofilum ochraceum (3644300 bp)	k__Bacteria p__Proteobacteria c__Gammaproteobacteria o__Chromatiales f__Ectothiorhodospiraceae g__Halofilum s__Halofilum_ochraceum t__GCF_001614315	gi|1093992509|ref|NZ_LVEG02000032.1|:1-1759 (1759 bp) gi|755573811|gb|KP052777.1|:1-1463 (1463 bp)
Halolamina sp. (3472520 bp)	k__Archaea p__Euryarchaeota c__Halobacteria o__Haloferacales f__Halorubraceae g__Halolamina s__Halolamina_sp. t__GCF_002025255	gi|582992856|gb|KF314045.2|:1-1473 (1473 bp) gi|582992855|gb|KF314044.2|:1-1473 (1473 bp) gi|402483711|gb|JX192605.1|:1-1472 (1472 bp) gi|672238891|ref|NR_125479.1|:1-1473 (1473 bp) gi|631252256|ref|NR_113454.1|:1-1472 (1472 bp) gi|469657984|gb|JX014295.2|:1-1473 (1473 bp) gi|782804855|gb|KJ573433.3|:1-1470 (1470 bp) gi|1109434189|gb|LT634694.1|:1-1422 (1422 bp)
Halomicrobium zhouii (4250330 bp)	k__Archaea p__Euryarchaeota c__Halobacteria o__Halobacteriales f__Haloarculaceae g__Halomicrobium s__Halomicrobium_zhouii t__GCF_900114435	GeneID:2667960611 (1474 bp) GeneID:2667961851 (1473 bp)
Halomonas subglaciescola	k__Bacteria p__Proteobacteria	gi|408688|gb|M93358.1|:1-1481 (1481 bp)

(3110200 bp)	c__Gammaproteobacteria o__Oceanospirillales f__Halomonadaceae g__Halomonas s__Halomonas_subglaciescola t__GCF_900142895	gi|17976827|gb|AJ306892.1|: 1-1531 (1531 bp)
Hyphomonas sp. L-53-1-40 (3310970 bp)	k__Bacteria p__Proteobacteria c__Alphaproteobacteria o__Rhodobacterales f__Hyphomonadaceae g__Hyphomonas s__Hyphomonas_sp._L-53-1-40 t__GCF_000682775	GeneID:2583254374 (1460 bp) gi|736811441|ref|NZ_AWFI01000010.1|:257-1506 (1250 bp)
Lentimicrobium saccharophilum (4514410 bp)	k__Bacteria p__Bacteroidetes c__ Bacteroidia o__Bacteroidales f__Lentimicrobiaceae g__Lentimicrobium s__Lentimicrobium_saccharophilum t__GCF_001192835	gi|1270116549|gb|MG264261.1|:1-1335 (1335 bp) gi|1270116492|gb|MG264204.1|:1-1181 (1181 bp) gi|1270116473|gb|MG264185.1|:1-1378 (1378 bp)
Methanosaeta sp. (2550000 bp)	k__Archaea p__Euryarchaeota c__Methanomicrobia o__Methanosarcinales f__Methanosaetaceae g__Methanosaeta s__Methanosaeta_sp. t__GCA_001412415	gi|941503567|gb|LKUG01000956.1|:1-819 (819 bp) gi|941506653|gb|LKUG01000340.1|:1-2179 (2179 bp) gi|941507386|gb|LKUG01000041.1|:1106-2395 (1290 bp) gi|941507386|gb|LKUG01000041.1|:2557-3012 (456 bp) gi|941507386|gb|LKUG01000041.1|:3028-3570 (543 bp) gi|941507386|gb|LKUG01000041.1|:3966-4766 (801 bp)
Micromonas sp. (20000000 bp)	k__Eukaryota p__Chlorophyta	gi|1269271348|gb|KY095012.1|:1-663 (663 bp)

	c__Mamiellophyceae o__Mamiellales f__Mamiellaceae g__Micromonas s__Micromonas_sp. t__GCA_001430725	gi|1269271344|gb|KY095008.1|:1-922 (922 bp) gi|1269271342|gb|KY095006.1|:1-926 (926 bp) gi|1027901359|gb|KU244632.1|:1-1703 (1703 bp) gi|65427940|gb|AY955011.1|:1-1727 (1727 bp) gi|959096220|gb|KT860759.1|:1-803 (803 bp) gi|1269271382|gb|KY095046.1|:1-166 (166 bp)
Monoraphidium neglectum (69711800 bp)	k__Eukaryota p__Chlorophyta c__Chlorophyceae o__Sphaeropleales f__Selenastraceae g__Monoraphidium s__Monoraphidium_neglectum t__GCF_000611645	gi|926773737|ref|XM_014039889.1|:1-426 (426 bp)
Monosiga brevicollis (41709900 bp)	k__Eukaryota p__Eukaryota_noname c__Eukaryota_noname o__Choanoflagellida f__Salpingoecidae g__Monosiga s__Monosiga_brevicollis t__GCF_000002865	gi|4093172|gb|AF100940.1|:1-1796 (1796 bp) gi|167521649|ref|XM_001745111.1|:1-1164 (1164 bp) gi|167519279|ref|XM_001743928.1|:1-2139 (2139 bp)
Nanohaloarchaea archaeon SG9 (1118570 bp)	k__Archaea p__Candidatus_Nanohaloarchaeota c__Nanohaloarchaea o__Nanohaloarchaea_noname f__Nanohaloarchaea_noname g__Nanohaloarchaea_noname s__Nanohaloarchaea_archaeon_SG9 t__GCA_001761425	gi|1078647179|gb|CP012986.1|:1-1383 (1383 bp)

Symbiodinium sp. (700000000 bp)	k__Eukaryota p__Eukaryota_noname c__Dinophyceae o__Suessiales f__Symbiodiniaceae g__Symbiodinium s__Symbiodinium_sp. t__GCA_001939145	gi|831180849|gb|LK934668.1|:1-3674 (3674 bp) gi|831180848|gb|LK934667.1|:1-2200 (2200 bp) gi|321373282|gb|HQ407545.1|:1-347 (347 bp) gi|321373279|gb|HQ407542.1|:1-348 (348 bp) gi|171676041|gb|EU567175.1|:1-395 (395 bp) gi|171676035|gb|EU567169.1|:1-395 (395 bp) gi|171676030|gb|EU567164.1|:1-390 (390 bp) gi|171676018|gb|EU567152.1|:1-378 (378 bp)
Thalassiosira oceanica (92185600 bp)	k__Eukaryota p__Bacillariophyta c__Coscinodiscophyceae o__Thalassiosirales f__Thalassiosiraceae g__Thalassiosira s__Thalassiosira_oceanica t__GCA_000296195	gi|126022826|gb|EF362633.1|:1-663 (663 bp) gi|126022825|gb|EF362632.1|:1-663 (663 bp) gi|126022823|gb|EF362630.1|:1-663 (663 bp) gi|119633043|gb|EF134955.1|:1-665 (665 bp) gi|119633042|gb|EF134954.1|:1-666 (666 bp)
Volvox carteri (137684000 bp)	k__Eukaryota p__Chlorophyta c__Chlorophyceae o__Chlamydomonadales f__Volvocaceae g__Volvox s__Volvox_carteri t__GCF_000143455	gi|485820272|gb|AB771954.1|:1-363 (363 bp) gi|485820271|gb|AB771953.1|:1-363 (363 bp) gi|302831268|ref|XM_002947154.1|:1-1993 (1993 bp) gi|302829127|ref|XM_002946085.1|:1-1137 (1137 bp)

Appendix F

KO numbers associated with specific pathways/enzymes used in the KEGG analysis component of Cavlab pipeline

Table F1. List of KO numbers associated with specific pathways. The table mentions all KO numbers explored for the calculation of various pathway/enzyme abundances, including KO numbers associated with enzymes that catalyse redox reactions (o/r – oxidation/reduction) or are homologous enzymes (a/m – ammonia/methane monooxygenase). The yellow-highlighted KO numbers were introduced in the preliminary Cavlab pipeline v1.2 (Appendix B). Of these, the KO numbers in red-highlighted text were removed and/or replaced in Cavlab pipeline v4. All other KO numbers were added to the Cavlab pipeline in v4. The pathway/enzyme abundances were calculated using the method described in Chapter 2 section 2.3.3.4 and the python code provided in Appendix C.

KO number	Pathway/enzyme	Protein name	EC number
K00016	Fermentation	L-lactate dehydrogenase	EC:1.1.1.27
K03778	Fermentation	D-lactate dehydrogenase; LdhA	EC:1.1.1.28
K00169	Fermentation	Pyruvate ferredoxin oxidoreductase alpha subunit	EC:1.2.7.1
K00170	Fermentation	Pyruvate ferredoxin oxidoreductase beta subunit	EC:1.2.7.1
K02256	Respiration	Cytochrome c oxidase subunit 1	EC:1.9.3.1
K02262	Respiration	Cytochrome c oxidase subunit 3	
K02274	Respiration	Cytochrome c oxidase subunit I	EC:1.9.3.1
K02276	Respiration	Cytochrome c oxidase subunit III	EC:1.9.3.1
K00400	Methanogenesis	Methyl coenzyme M reductase system, component A2	
K00401	Methanogenesis	Methyl coenzyme M reductase beta subunit; McrB	EC:2.8.4.1
K16157	Methane oxidation	Methane monooxygenase component A alpha chain	EC:1.14.13.25
K16158	Methane oxidation	Methane monooxygenase component A beta chain	EC:1.14.13.25
K16159	Methane oxidation	Methane monooxygenase component A gamma chain	EC:1.14.13.25
K16161	Methane oxidation	Methane monooxygenase component C	EC:1.14.13.25
K10944a/m	Methane oxidation/Nitrification	Methane/ammonia monooxygenase subunit A	EC:1.14.18.3, EC:1.14.99.39

K10945a/m	Methane oxidation/Nitrification	Methane/ammonia monooxygenase subunit B	
K10946a/m	Methane oxidation/Nitrification	Methane/ammonia monooxygenase subunit C	
K03518	Mo/Cu carbon monoxide dehydrogenase	Aerobic carbon-monoxide dehydrogenase small subunit	EC:1.2.5.3
K03519	Mo/Cu carbon monoxide dehydrogenase	Aerobic carbon-monoxide dehydrogenase medium subunit	EC:1.2.5.3
K03520	Mo/Cu carbon monoxide dehydrogenase	Aerobic carbon-monoxide dehydrogenase large subunit	EC:1.2.5.3
K15230	rTCA cycle	ATP-citrate lyase alpha-subunit; AclA	EC:2.3.3.8
K15231	rTCA cycle	ATP-citrate lyase beta-subunit; AclB	EC:2.3.3.8
K15232	rTCA cycle II	Citryl-CoA synthetase large subunit	EC:6.2.1.18
K15233	rTCA cycle II	Citryl-CoA synthetase small subunit	
K15234	rTCA cycle II	Citryl-CoA lyase	EC:4.1.3.34
K00174	rTCA cycle II	2-Oxoglutarate/2-oxoacid ferredoxin oxidoreductase subunit alpha	EC:1.2.7.3, EC1.2.7.11
K00175	rTCA cycle II	2-Oxoglutarate/2-oxoacid ferredoxin oxidoreductase subunit beta	EC:1.2.7.3 1.2.7.11
K00244	rTCA cycle II	Fumarate reductase flavoprotein subunit	EC:1.3.5.4
K00192	Wood-Ljungdahl pathway	Anaerobic carbon-monoxide dehydrogenase, CODH/ACS complex subunit alpha; CdhA	EC:1.2.7.4
K00194	Wood-Ljungdahl pathway	Acetyl-CoA decarbonylase/synthase; AcsD	EC:2.1.1.245
K00197	Wood-Ljungdahl pathway	Acetyl-CoA decarbonylase/synthase; AcsC	EC:2.1.1.245
K00198	Wood-Ljungdahl pathway	Anaerobic carbon-monoxide dehydrogenase catalytic subunit; CooS, AcsA	EC:1.2.7.4
K14138	Wood-Ljungdahl pathway	Acetyl-CoA synthase; AcsB	EC:2.3.1.169
K01602	Calvin cycle	Ribulose-bisphosphate carboxylase small chain; RbcS	EC:4.1.1.39
K00855	Calvin cycle	Phosphoribulokinase; PrkB	EC:2.7.1.19

K01601	Calvin cycle	Ribulose-bisphosphate carboxylase large chain; RbcL, CbbL	EC:4.1.1.39
K02586	Nitrogen fixation	Nitrogenase molybdenum-iron protein alpha chain; NifD	EC:1.18.6.1
K02591	Nitrogen fixation	Nitrogenase molybdenum-iron protein beta chain; NifK	EC:1.18.6.1
K00531	Nitrogen fixation	Nitrogenase delta subunit; AnfG	EC:1.18.6.1
K02588	Nitrogen fixation	Nitrogenase iron protein; NifH	
K01915	Ammonia assimilation	Glutamine synthetase; GlnA	EC 6.3.1.2
K00264	Ammonia assimilation	Glutamate synthase (NADH); Glt1	EC 1.4.1.14
K00265	Ammonia assimilation	Glutamate synthase (NADPH) large chain; GltB	EC 1.4.1.13
K00266	Ammonia assimilation	Glutamate synthase (NADPH) small chain; GltD	EC 1.4.1.13
K00284	Ammonia assimilation	Glutamate synthase (ferredoxin); GltS	EC 1.4.7.1
K00370	Dissimilatory nitrate reduction	Nitrate reductase/nitrite oxidoreductase, alpha subunit; NarG, NarZ, NxrA	EC:1.7.5.1, EC:1.7.99.-
K00363	Dissimilatory nitrate reduction	Nitrite reductase (NADH) small subunit; NirD	EC:1.7.1.15
K15876	Dissimilatory nitrate reduction	Cytochrome c nitrite reductase small subunit; NrfH	
K00362	Dissimilatory nitrite reduction (ammonia-forming)	Nitrite reductase (NADH) large subunit; NirB	EC:1.7.1.15
K03385	Dissimilatory nitrite reduction (ammonia-forming)	Nitrite reductase (cytochrome c-552); NrfA	EC:1.7.2.2
K00368	Dissimilatory nitrite reduction (NO-forming)	Nitrite reductase (NO-forming); NirK	EC:1.7.2.1
K17877	Assimilatory nitrate reduction	Nitrite reductase (NAD(P)H); Nit-6	EC:1.7.1.4
K00360	Assimilatory nitrate reduction	Assimilatory nitrate reductase electron transfer subunit; NasB	EC:1.7.99.-
K00366	Assimilatory nitrate reduction	Ferredoxin-nitrite reductase; NirA	EC:1.7.7.1

K00367	Assimilatory nitrate reduction	Ferredoxin-nitrate reductase; NarB	EC:1.7.7.2
K10535	Nitrification	Hydroxylamine dehydrogenase	EC:1.7.2.6
K02305	Nitric oxide reduction	Nitric oxide reductase subunit C; NorC	
K04561	Nitric oxide reduction	Nitric oxide reductase subunit B; NorB	EC:1.7.2.5
K00376	Nitric oxide reduction	Nitrous oxide reductase; NosZ	EC:1.7.2.4
K20932	Anammox	Hydrazine synthase subunit	EC:1.7.2.7
K20933	Anammox	Hydrazine synthase subunit	EC:1.7.2.7
K20934	Anammox	Hydrazine synthase subunit	EC:1.7.2.7
K20935	Anammox	Hydrazine dehydrogenase [EC:1.7.2.8]	
K02567	Periplasmic nitrate reduction	Periplasmic nitrate reductase; NapA	EC:1.7.99.-
K17222	SOX system	L-cysteine S-thiosulfotransferase; SoxA	EC:2.8.5.2
K17223	SOX system	L-cysteine S-thiosulfotransferase; SoxX	EC:2.8.5.2
K17224	SOX system	S-Sulfosulfanyl-L-cysteine sulfohydrolase; SoxB	EC:3.1.6.20
K17225	SOX system	Sulfane dehydrogenase subunit; SoxC	
K17226	SOX system	Sulfur-oxidizing protein; SoxY	
K17227	SOX system	Sulfur-oxidizing protein; SoxZ	
K00456	Cysteine dioxygenase	Cysteine dioxygenase; Cdo1	EC:1.13.11.20
K01011	Thiosulfate/3-mercaptopyruvate sulfurtransferase	Thiosulfate/3-mercaptopyruvate sulfurtransferase; Tst, Mpst	EC:2.8.1.1, EC:2.8.1.2
K00955	Sulfate reduction I/APS reduction I	Bifunctional enzyme CysN/CysC; CycNC	EC:2.7.7.4, EC:2.7.1.25
K00956	Sulfate reduction I	Sulfate adenylyltransferase subunit 1; CysN	EC:2.7.7.4
K00957	Sulfate reduction I	Sulfate adenylyltransferase subunit 2; CycD	EC:2.7.7.4
K00958o/r	Sulfate reduction I/Sulfate reduction II/APS oxidation	Sulfate adenylyltransferase; Sat	EC:2.7.7.4
K00860	APS reduction I	Adenylylsulfate kinase; CycC	EC:2.7.1.25

K05907	APS reduction II	Adenylylsulfate reductase (glutathione); Apr	EC:1.8.4.9
K00390	APS reduction II/PAPS reduction	Phosphoadenosine phosphosulfate reductase; CysH	EC:1.8.4.8, EC:1.8.4.10
K00394o/r	APS reduction III/Sulfite oxidation	Adenylylsulfate reductase, subunit A; AprA	EC:1.8.99.2
K00395o/r	APS reduction III/Sulfite oxidation	Adenylylsulfate reductase, subunit B; AprB	EC:1.8.99.2
K00380	Sulfite reduction I	Sulfite reductase (NADPH) flavoprotein alpha-component; CysJ	EC:1.8.1.2
K00381	Sulfite reduction I	Sulfite reductase (NADPH) hemoprotein beta-component; CysI	EC:1.8.1.2
K00392	Sulfite reduction I	Sulfite reductase (ferredoxin); Sir	EC:1.8.7.1
K11180o/r	Sulfite reduction II/Sulfur/polysulfide oxidation	Dissimilatory sulfite reductase alpha subunit; DsrA	EC:1.8.99.5
K11181o/r	Sulfite reduction II/Sulfur/polysulfide oxidation	Dissimilatory sulfite reductase beta subunit; DsrB	EC:1.8.99.5
K17218	Sulfide oxidation	Sulfide:quinone oxidoreductase; Sqr	EC:1.8.5.4
K17229	Sulfide oxidation	Sulfide dehydrogenase [flavocytochrome c] flavoprotein chain; FccB	EC:1.8.2.3
K02689	Photosystem I	Photosystem I P700 chlorophyll a apoprotein A1; PsaA	
K02690	Photosystem I	Photosystem I P700 chlorophyll a apoprotein A2; PsaB	
K02691	Photosystem I	Photosystem I subunit VII; PsaC	
K02692	Photosystem I	Photosystem I subunit II; PsaD	
K02693	Photosystem I	Photosystem I subunit IV; PsaE	
K02694	Photosystem I	Photosystem I subunit III; PsaF	
K02703	Photosystem II	Photosystem II P680 reaction center D1 protein; PsbA	EC:1.10.3.9
K02704	Photosystem II	Photosystem II CP47 chlorophyll apoprotein; PsbB	

K02705	Photosystem II	Photosystem II CP43 chlorophyll apoprotein; PsbC	
K02706	Photosystem II	Photosystem II P680 reaction center D2 protein; PsbD	EC:1.10.3.9
K02707	Photosystem II	Photosystem II cytochrome b559 subunit alpha; PsbE	
K02708	Photosystem II	Photosystem II cytochrome b559 subunit beta; PsbF	
K08940	Type 1 RC core complex (GSB)	Photosystem P840 reaction center large subunit; PscA	
K08941	Type 1 RC core complex (GSB)	Photosystem P840 reaction center iron-sulfur protein; PscB	
K08942	Type 1 RC core complex (GSB)	Photosystem P840 reaction center cytochrome c551; PscC	
K08943	Type 1 RC core complex (GSB)	Photosystem P840 reaction center protein PscD	
K08928	RC complex (purple bacteria)	Photosynthetic reaction center L subunit; PufL	
K08929	RC complex (purple bacteria)	Photosynthetic reaction center M subunit; PufM	
K00909	Rhodopsins	Rhodopsin kinase; GRK1_7	EC:2.7.11.14
K04250	Rhodopsins	Rhodopsin; RHO, OPN2	
K04641	Rhodopsins	Bacteriorhodopsin; Bop	
K04642	Rhodopsins	Halorhodopsin; Hop	
K04643	Rhodopsins	Sensory rhodopsin; Sop	
K09836	Astaxanthin	Beta-carotene ketolase (CrtW type)	
K15746	Astaxanthin	Beta-carotene 3-hydroxylase	EC:1.14.15.24
K15342	CRISPR-Cas spacer acquisition	CRISP-associated protein Cas1	
K09951	CRISPR-Cas spacer acquisition	CRISPR-associated protein Cas2	
K07012	CRISPR 1I	CRISPR-associated endonuclease/helicase Cas3	EC:3.1.-.-, EC:3.6.4.-
K07475	CRISPR 1I	CRISPR-associated endonuclease Cas3-HD	EC:3.1.-.-

K19088	CRISPR 1IA	CRISPR-associated protein Cst1; Cas8a	
K19087	CRISPR 1IA	CRISPR-associated protein Csa5	
K19117	CRISPR 1IC	CRISPR-associated protein Csd1; Cas8c	
K19123	CRISPR 1IE	CRISPR system Cascade subunit CasA; Cse1	
K19046	CRISPR 1IE	CRISPR system Cascade subunit CasB; Cse2	
K19127	CRISPR 1IF	CRISPR-associated protein Csy1	
K19128	CRISPR 1IF	CRISPR-associated protein Csy2	
K19129	CRISPR 1IF	CRISPR-associated protein Csy3	
K09952	CRISPR 2II	CRISPR-associated endonuclease Csn1; Cas9	EC:3.1.-.-
K19137	CRISPR 2IIA	CRISPR-associated protein Csn2	
K07464	CRISPR 2IIB	CRISPR-associated exonuclease Cas4	EC:3.1.12.1
K07016	CRISPR 1III	CRISPR-associated protein Csm1; Cas10	
K19138	CRISPR 1IIIA	CRISPR-associated protein Csm2	
K19141	CRISPR 1IIIB	CRISPR-associated protein Cmr5	
K00437	[NiFe] hydrogenase	[NiFe] Hydrogenase large subunit; HydB	EC:1.12.2.1
K05922	[NiFe] hydrogenase	Quinone-reactive Ni/Fe-hydrogenase large subunit; HydB	EC:1.12.5.1
K00436	NAD-reducing hydrogenase/diaphorase	NAD-reducing hydrogenase large subunit; HoxH	EC:1.12.1.2
K18332	NADP-reducing hydrogenase	NADP-reducing hydrogenase subunit; HndD	EC:1.12.1.3
K17997	Iron-hydrogenase	Iron-hydrogenase subunit alpha; HydA	EC:1.12.1.4
K00532	Ferredoxin hydrogenase (monomeric)	Ferredoxin hydrogenase	EC:1.12.7.2
K00533	Ferredoxin hydrogenase (trimeric)	Ferredoxin hydrogenase large subunit	EC:1.12.7.2
K18016	Membrane-bound hydrogenase	Membrane-bound hydrogenase subunit alpha; MbhL	EC:1.12.7.2

K14068	Methanophenazine hydrogenase	Methanophenazine hydrogenase, large subunit; VhoA, VhtA	EC:1.12.98.3
K00440	Coenzyme F420 hydrogenase	Coenzyme F420 hydrogenase subunit alpha; FrhA	EC:1.12.98.1
K13942	5,10-Methenyltetrahydrometh anopterin hydrogenase	5,10-Methenyltetrahydromethanopterin hydrogenase; Hmd	EC:1.12.98.2
K14126	F420-non-reducing hydrogenase	F420-Non-reducing hydrogenase large subunit; MvhA, VhuA, VhcA	EC:1.12.99.-, EC:1.8.98.5
K17993	Sulfhydrogenase	Sulfhydrogenase alpha subunit; HydA	EC:1.12.1.3, EC:1.12.1.5
K11472	Glycolate utilization	Glycolate oxidase FAD binding subunit; GlcE	
K08688	Creatine utilization	Creatinase	EC:3.5.3.3
K00301	Sarcosine utilization I	Sarcosine oxidase	EC:1.5.3.1
K00302	Sarcosine utilization II	Sarcosine oxidase, subunit alpha	EC:1.5.3.1
K00303	Sarcosine utilization II	Sarcosine oxidase, subunit beta	EC:1.5.3.1
K00304	Sarcosine utilization II	Sarcosine oxidase, subunit delta	EC:1.5.3.1
K00305	Sarcosine utilization II	Sarcosine oxidase, subunit gamma	EC:1.5.3.1
K03851	Taurine utilization	Taurine-pyruvate aminotransferase; Tpa	EC:2.6.1.77
K03852	Taurine utilization	Sulfoacetaldehyde acetyltransferase	EC:2.3.3.15
K01130	Sulfate ester hydrolysis	Arylsulfatase; AslA	EC:3.1.6.1
K15923	Fucoidan degradation	Alpha-L-fucosidase 2; AXY8, FUC95A, AfcA	EC:3.2.1.51
K00879	Fucose utilization	L-fuculokinase; FucK	EC:2.7.1.51
K01628	Fucose utilization	L-fuculose-phosphate aldolase; FucA	EC:4.1.2.17
K00064	Fucose utilization II	D-threo-aldose 1-dehydrogenase	EC:1.1.1.122
K00848	Rhamnose utilization	Rhamnulokinase; RhaB	EC:2.7.1.5
K01629	Rhamnose utilization	Rhamnulose-1-phosphate aldolase; RhaD	EC:4.1.2.19
K01183	Chitin degradation I	Chitinase	EC:3.2.1.14
K13381	Chitin degradation II	Bifunctional chitinase/lysozyme	EC:3.2.1.14, EC:3.2.1.17

K14083	Trimethylamine/glycine betaine methyltransferase	Trimethylamine---corrinoid protein Co-methyltransferase; MttB	EC:2.1.1.250
K16178	Dimethylamine utilization	Dimethylamine---corrinoid protein Co-methyltransferase; MtbB	EC:2.1.1.249
K16176	Monomethylamine utilization	Methylamine---corrinoid protein Co-methyltransferase; MtmB	EC:2.1.1.248
K00702	Cellobiose utilization	Cellobiose phosphorylase	EC:2.4.1.20
K16149	Glycogen synthesis (overall)	1,4-alpha-glucan branching enzyme	EC:2.4.1.18
K00975	Glycogen synthesis I	Glucose-1-phosphate adenylyltransferase; GlgC	EC:2.7.7.27
K00703	Glycogen synthesis I	Starch synthase; GlgA	EC:2.4.1.21
K16146	Glycogen synthesis II	Maltokinase; Pep2	EC:2.7.1.175
K16147	Glycogen synthesis II	Starch synthase (maltosyl-transferring); GlcE	EC:2.4.99.16
K01176	Starch degradation	Alpha-amylase; AMY, AmyA, MalS	EC:3.2.1.1
K11959	Urea transporter	Urea transport system substrate-binding protein; UrtA	
K11960	Urea transporter	Urea transport system permease protein; UrtB	
K11961	Urea transporter	Urea transport system permease protein; UrtC	
K11962	Urea transporter	urea transport system ATP-binding protein; UrtD	
K11963	Urea transporter	urea transport system ATP-binding protein; UrtE	
K02045	Sulfate transporter	Sulfate/thiosulfate transport system ATP-binding protein; CysA	EC:7.3.2.3
K02046	Sulfate transporter	Sulfate/thiosulfate transport system permease protein; CysU	
K02047	Sulfate transporter	Sulfate/thiosulfate transport system permease protein; CysW	
K02048	Sulfate transporter	Sulfate/thiosulfate transport system substrate-binding protein; CysP	

K15576	Nitrate/nitrite transporter	Nitrate/nitrite transport system substrate-binding protein; NrtA, NasF, CynA	
K15577	Nitrate/nitrite transporter	Nitrate/nitrite transport system permease protein; NrtB, NasE, CynB	
K15578	Nitrate/nitrite transporter	Nitrate/nitrite transport system ATP-binding protein; NrtC, NasD	EC:3.6.3.-
K15579	Nitrate/nitrite transporter	Nitrate/nitrite transport system ATP-binding protein; NrtD, CynD	
K11950	Bicarbonate transporter	Bicarbonate transport system substrate-binding protein; CmpA	
K11951	Bicarbonate transporter	Bicarbonate transport system permease protein; CmpB	
K11952	Bicarbonate transporter	Bicarbonate transport system ATP-binding protein; CmpC	EC:3.6.3.-
K11953	Bicarbonate transporter	Bicarbonate transport system ATP-binding protein; CmpD	EC:3.6.3.-
K10831	Taurine transporter	Taurine transport system ATP-binding protein; TauB	EC:7.6.2.7
K15551	Taurine transporter	Taurine transport system substrate-binding protein; TauA	
K15552	Taurine transporter	Taurine transport system permease protein; TauC	
K15553	Sulfonate transporter	Sulfonate transport system substrate-binding protein; SsuA	
K15554	Sulfonate transporter	Sulfonate transport system permease protein; SsuC	
K15555	Sulfonate transporter	Sulfonate transport system ATP-binding protein; SsuB	EC:3.6.3.-
K11069	Spermidine/putrescine transporter	Spermidine/putrescine transport system substrate-binding protein; PotD	
K11070	Spermidine/putrescine transporter	Spermidine/putrescine transport system permease protein; PotC	
K11071	Spermidine/putrescine transporter	Spermidine/putrescine transport system permease protein; PotB	

K11072	Spermidine/putrescine transporter	Spermidine/putrescine transport system ATP-binding protein; PotA	EC:7.6.2.11
K11073	Putrescine transporter	Putrescine transport system substrate-binding protein; PotF	
K11074	Putrescine transporter	Putrescine transport system permease protein; PotI	
K11075	Putrescine transporter	Putrescine transport system permease protein; PotH	
K11076	Putrescine transporter	Putrescine transport system ATP-binding protein; PotG	
K02036	Phosphate transporter	Phosphate transport system ATP-binding protein; PstB	EC:7.3.2.1
K02037	Phosphate transporter	Phosphate transport system permease protein; PstC	
K02038	Phosphate transporter	Phosphate transport system permease protein; PstA	
K02040	Phosphate transporter	Phosphate transport system substrate-binding protein; PstS	
K02041	Phosphonate transporter	Phosphonate transport system ATP-binding protein	EC:7.3.2.2
K02042	Phosphonate transporter	Phosphonate transport system permease protein; PhnE	
K02044	Phosphonate transporter	Phosphonate transport system substrate-binding protein; PhnD	
K11081	2-Aminoethylphosphonate transporter	2-Aminoethylphosphonate transport system substrate-binding protein; PhnS	
K11082	2-Aminoethylphosphonate transporter	2-Aminoethylphosphonate transport system permease protein; PhnV	
K11083	2-Aminoethylphosphonate transporter	2-Aminoethylphosphonate transport system permease protein; PhnU	
K11084	2-Aminoethylphosphonate transporter	2-Aminoethylphosphonate transport system ATP-binding protein; PhnT	

K02000	Glycine betaine/proline transporter	Glycine betaine/proline transport system ATP-binding protein; ProV	EC:7.6.2.9
K02001	Glycine betaine/proline transporter	Glycine betaine/proline transport system permease protein; ProW	
K02002	Glycine betaine/proline transporter	Glycine betaine/proline transport system substrate-binding protein; ProX	
K05845	Osmoprotectant transporter	Osmoprotectant transport system substrate-binding protein; OpuC	
K05846	Osmoprotectant transporter	Osmoprotectant transport system permease protein; OpuBD	
K05847	Osmoprotectant transporter	Osmoprotectant transport system ATP-binding protein; OpuA	EC:7.6.2.9
K10108	Maltose/maltodextrin transporter	Maltose/maltodextrin transport system substrate-binding protein; MalE	
K10109	Maltose/maltodextrin transporter	Maltose/maltodextrin transport system permease protein; MalF	
K10110	Maltose/maltodextrin transporter	Maltose/maltodextrin transport system permease protein; MalG	
K15770	Arabinogalactan oligomer/maltooligosaccharide transporter	Arabinogalactan oligomer/maltooligosaccharide transport system substrate-binding protein; CycB, GanO	
K15771	Arabinogalactan oligomer/maltooligosaccharide transporter	Arabinogalactan oligomer/maltooligosaccharide transport system permease protein; GanP	
K15772	Arabinogalactan oligomer/maltooligosaccharide transporter	Arabinogalactan oligomer/maltooligosaccharide transport system permease protein; GanQ	
K10117	Raffinose/stachyose/melibiose transporter	Raffinose/stachyose/melibiose transport system substrate-binding protein; MsmE	
K10118	Raffinose/stachyose/melibiose transporter	Raffinose/stachyose/melibiose transport system permease protein; MsmF	

K10119	Raffinose/stachyose/melibiose transporter	Raffinose/stachyose/melibiose transport system permease protein; MsmG
K10232	Alpha-Glucoside transporter	Alpha-glucoside transport system substrate-binding protein; AglE, GgtB
K10233	Alpha-Glucoside transporter	Alpha-glucoside transport system permease protein; AglF, GgtC
K10234	Alpha-Glucoside transporter	Alpha-glucoside transport system permease protein; AglG, GgtD
K10235	Alpha-Glucoside transporter	Alpha-glucoside transport system ATP-binding protein; AglK
K10196	Glucose/arabinose transporter	Glucose/arabinose transport system substrate-binding protein
K10197	Glucose/arabinose transporter	Glucose/arabinose transport system permease protein
K10198	Glucose/arabinose transporter	Glucose/arabinose transport system permease protein
K10199	Glucose/arabinose transporter	Glucose/arabinose transport system ATP-binding protein
K17315	Glucose/mannose transporter	Glucose/mannose transport system substrate-binding protein; GtsA, GlcE
K17316	Glucose/mannose transporter	Glucose/mannose transport system permease protein; GtsB, GlcF
K17317	Glucose/mannose transporter	Glucose/mannose transport system permease protein; GtsC, GlcG
K10236	Trehalose/maltose transporter	Trehalose/maltose transport system substrate-binding protein; ThuE
K10237	Trehalose/maltose transporter	Trehalose/maltose transport system permease protein; ThuF, SugA
K10238	Trehalose/maltose transporter	Trehalose/maltose transport system permease protein; ThuG, SugB
K17311	Trehalose transporter	Trehalose transport system substrate-binding protein; TreS
K17312	Trehalose transporter	Trehalose transport system permease protein; TreT

K17313	Trehalose transporter	Trehalose transport system permease protein; TreU	
K17314	Trehalose transporter	Trehalose transport system ATP-binding protein; TreV	
K10200	N-Acetylglucosamine transporter	N-acetylglucosamine transport system substrate-binding protein	
K10201	N-Acetylglucosamine transporter	N-acetylglucosamine transport system permease protein	
K10202	N-Acetylglucosamine transporter	N-acetylglucosamine transport system permease protein	
K10240	Cellobiose transporter	Cellobiose transport system substrate-binding protein; CebE	
K10241	Cellobiose transporter	Cellobiose transport system permease protein; CebF	
K10242	Cellobiose transporter	Cellobiose transport system permease protein; CebG	
K17329	N,N'-Diacetylchitobiose transporter	N,N'-diacetylchitobiose transport system substrate-binding protein; DasA	
K17330	N,N'-Diacetylchitobiose transporter	N,N'-diacetylchitobiose transport system permease protein; DasB	
K17331	N,N'-Diacetylchitobiose transporter	N,N'-diacetylchitobiose transport system permease protein; DasC	
K17244	Putative chitobiose transporter	Putative chitobiose transport system substrate-binding protein; ChiE	
K17245	Putative chitobiose transporter	Putative chitobiose transport system permease protein; ChiF	
K17246	Putative chitobiose transporter	Putative chitobiose transport system permease protein; ChiG	
K10537	L-Arabinose transporter	L-arabinose transport system substrate-binding protein; AraF	
K10538	L-Arabinose transporter	L-arabinose transport system permease protein; AraH	
K10539	L-Arabinose transporter	L-arabinose transport system ATP-binding protein; AraG	EC:7.5.2.12

K10188	Lactose/L-arabinose transporter	Lactose/L-arabinose transport system substrate-binding protein; LacE, AraN	
K10189	Lactose/L-arabinose transporter	Lactose/L-arabinose transport system permease protein; LacF, AraP	
K10190	Lactose/L-arabinose transporter	Lactose/L-arabinose transport system permease protein; LacG, AraQ	
K10191	Lactose/L-arabinose transporter	Lactose/L-arabinose transport system ATP-binding protein; LacK	
K10543	D-Xylose transporter	D-xylose transport system substrate-binding protein; XylF	
K10544	D-Xylose transporter	D-xylose transport system permease protein; XylH	
K10545	D-Xylose transporter	D-xylose transport system ATP-binding protein; XylG	EC:3.6.3.17
K17326	Xylobiose transporter	Xylobiose transport system substrate-binding protein; BxlE	
K17327	Xylobiose transporter	Xylobiose transport system permease protein; BxlF	
K17328	Xylobiose transporter	Xylobiose transport system permease protein; BxlG	
K10546	Multiple sugar transporter	Putative multiple sugar transport system substrate-binding protein; ChvE	
K10547	Multiple sugar transporter	Putative multiple sugar transport system permease protein; GguB	
K10548	Multiple sugar transporter	Putative multiple sugar transport system ATP-binding protein; GguA	EC:3.6.3.17
K10552	Fructose transporter	Fructose transport system substrate-binding protein; FrcB	
K10553	Fructose transporter	Fructose transport system permease protein; FrcC	
K10554	Fructose transporter	Fructose transport system ATP-binding protein; FrcA	
K10559	Rhamnose transporter	Rhamnose transport system substrate-binding protein; RhaS	

K10560	Rhamnose transporter	Rhamnose transport system permease protein; RhaP	
K10561	Rhamnose transporter	Rhamnose transport system permease protein; RhaQ	
K10562	Rhamnose transporter	Rhamnose transport system ATP-binding protein; RhaT	EC:3.6.3.17
K10439	Ribose transporter	Ribose transport system substrate-binding protein; RbsB	
K10440	Ribose transporter	Ribose transport system permease protein; RbsC	
K10441	Ribose transporter	Ribose transport system ATP-binding protein	EC:3.6.3.17
K17202	Erythritol transporter	Erythritol transport system substrate-binding protein; EryG	
K17203	Erythritol transporter	Erythritol transport system permease protein; EryF	
K17204	Erythritol transporter	Erythritol transport system ATP-binding protein; EryE	
K10120	Putative fructooligosaccharide transporter	Fructooligosaccharide transport system substrate-binding protein; MsmE	
K10121	Putative fructooligosaccharide transporter	Fructooligosaccharide transport system permease protein; MsmF	
K10122	Putative fructooligosaccharide transporter	Fructooligosaccharide transport system permease protein; MsmG	
K17321	Glycerol transporter	Glycerol transport system substrate-binding protein; GlpV	
K17322	Glycerol transporter	Glycerol transport system permease protein; GlpP	
K17323	Glycerol transporter	Glycerol transport system permease protein; GlpQ	
K17324	Glycerol transporter	Glycerol transport system ATP-binding protein; GlpS	

K17325	Glycerol transporter	Glycerol transport system ATP-binding protein; GlpT	
K02025	Putative multiple sugar transporter	Multiple sugar transport system permease protein	
K02026	Putative multiple sugar transporter	Multiple sugar transport system permease protein	
K02027	Putative multiple sugar transporter	Multiple sugar transport system substrate-binding protein	
K02056	Putative simple sugar transporter	Simple sugar transport system ATP-binding protein	EC:3.6.3.17
K02057	Putative simple sugar transporter	Simple sugar transport system permease protein	
K02058	Putative simple sugar transporter	Simple sugar transport system substrate-binding protein	
K10013	Lysine/arginine/ornithine transporter	Lysine/arginine/ornithine transport system substrate-binding protein; ArgT	
K10014	Histidine transporter	Histidine transport system substrate-binding protein; HisJ	
K10015	Lysine/arginine/ornithine transporter/Histidine transporter	Histidine transport system permease protein; HisM	
K10016	Lysine/arginine/ornithine transporter/Histidine transporter	Histidine transport system permease protein; HisQ	
K10017	Lysine/arginine/ornithine transporter/Histidine transporter	Histidine transport system ATP-binding protein; HisP	EC:7.4.2.1
K10036	Glutamine transporter	Glutamine transport system substrate-binding protein; GlnH	
K10037	Glutamine transporter	Glutamine transport system permease protein; GlnP	
K10038	Glutamine transporter	Glutamine transport system ATP-binding protein; GlnQ	EC:7.4.2.1
K09996	Arginine transporter	Arginine transport system substrate-binding protein; ArtJ	

K09997	Arginine transporter	Arginine transport system substrate-binding protein; ArtI	
K09998	Arginine transporter	Arginine transport system permease protein; ArtM	
K09999	Arginine transporter	Arginine transport system permease protein; ArtQ	
K10000	Arginine transporter	Arginine transport system ATP-binding protein; ArtP	EC:7.4.2.1
K10001	Glutamate/aspartate transporter	Glutamate/aspartate transport system substrate-binding protein; GltI	
K10002	Glutamate/aspartate transporter	Glutamate/aspartate transport system permease protein; GltK, AatM	
K10003	Glutamate/aspartate transporter	Glutamate/aspartate transport system permease protein; GltJ, AatQ	
K10004	Glutamate/aspartate transporter	Glutamate/aspartate transport system ATP-binding protein; GltL, AatP	EC:7.4.2.1
K10039	Aspartate/glutamate/glutamine transporter	Aspartate/glutamate/glutamine transport system substrate-binding protein; Peb1A, GlnH	
K10040	Aspartate/glutamate/glutamine transporter	Aspartate/glutamate/glutamine transport system permease protein; Peb1B, GlnP, GlnM	
K10041	Aspartate/glutamate/glutamine transporter	Aspartate/glutamate/glutamine transport system ATP-binding protein; Peb1C, GlnQ	EC:7.4.2.1
K10018	Octopine/nopaline transporter	Octopine/nopaline transport system substrate-binding protein; OccT, NocT	
K10019	Octopine/nopaline transporter	Octopine/nopaline transport system permease protein; OccM, NocM	
K10020	Octopine/nopaline transporter	Octopine/nopaline transport system permease protein; OccQ, NocQ	
K10021	Octopine/nopaline transporter	Octopine/nopaline transport system ATP-binding protein; OccP, NocP	EC:7.4.2.1
K09969	General L-amino acid transporter	General L-amino acid transport system substrate-binding protein; AapJ, BztA	

K09970	General L-amino acid transporter	General L-amino acid transport system permease protein; AapQ, BztB	
K09971	General L-amino acid transporter	General L-amino acid transport system permease protein; AapM, BztC	
K09972	General L-amino acid transporter	General L-amino acid transport system ATP-binding protein; AapP, BztD	EC:7.4.2.1
K10005	Glutamate transporter	Glutamate transport system substrate-binding protein; GluB	
K10006	Glutamate transporter	Glutamate transport system permease protein; GluC	
K10007	Glutamate transporter	Glutamate transport system permease protein; GluD	
K10008	Glutamate transporter	glutamate transport system ATP-binding protein; GluA	EC:7.4.2.1
K02424	Cystine transporter	L-cystine transport system substrate-binding protein; FliY, TcyA	
K10009	Cystine transporter	L-cystine transport system permease protein; TcyB, YecS	
K10010	Cystine transporter	L-cystine transport system ATP-binding protein; TcyC, YecC	EC:7.4.2.1
K16956	L-Cystine transporter	L-cystine transport system substrate-binding protein; TcyJ	
K16957	L-Cystine transporter	L-cystine transport system substrate-binding protein; TcyK	
K16958	L-Cystine transporter	L-cystine transport system permease protein; TcyL	
K16959	L-Cystine transporter	L-cystine transport system permease protein; TcyM	
K16960	L-Cystine transporter	L-cystine transport system ATP-binding protein; TcyN	EC:7.4.2.1
K10022	Arginine/ornithine transporter	Arginine/ornithine transport system substrate-binding protein; AotJ	
K10023	Arginine/ornithine transporter	Arginine/ornithine transport system permease protein; AotM	
K10024	Arginine/ornithine transporter	Arginine/ornithine transport system permease protein; AotQ	

K10025	Arginine/ornithine transporter	Arginine/ornithine transport system ATP-binding protein; AotP	EC:7.4.2.1
K23059	Arginine/lysine/histidine transporter	Arginine/lysine/histidine transporter system substrate-binding protein; ArtP, ArtI	
K17077	Arginine/lysine/histidine transporter	Arginine/lysine/histidine transport system permease protein; ArtQ	
K23060	Arginine/lysine/histidine transporter	arginine/lysine/histidine transport system ATP-binding protein; ArtR, ArtM	EC:7.4.2.1
K01995	Branched-chain amino acid transporter	Branched-chain amino acid transport system ATP-binding protein; LivG	
K01996	Branched-chain amino acid transporter	Branched-chain amino acid transport system ATP-binding protein; LivF	
K01997	Branched-chain amino acid transporter	Branched-chain amino acid transport system permease protein; LivH	
K01998	Branched-chain amino acid transporter	Branched-chain amino acid transport system permease protein; LivM	
K01999	Branched-chain amino acid transporter	Branched-chain amino acid transport system substrate-binding protein; LivK	
K11954	Neutral amino acid transporter	Neutral amino acid transport system substrate-binding protein; NatB	
K11955	Neutral amino acid transporter	Neutral amino acid transport system permease protein; NatC	
K11956	Neutral amino acid transporter	Neutral amino acid transport system permease protein; NatD	
K11957	Neutral amino acid transporter	Neutral amino acid transport system ATP-binding protein; NatA	
K11958	Neutral amino acid transporter	Neutral amino acid transport system ATP-binding protein; NatE	
K02073	D-Methionine transporter	D-methionine transport system substrate-binding protein; MetQ	
K02072	D-Methionine transporter	D-methionine transport system permease protein; MetI	
K02071	D-Methionine transporter	D-methionine transport system ATP-binding protein; MetN	

K15580	Oligopeptide transporter	Oligopeptide transport system substrate-binding protein; OppA, MppA	
K15581	Oligopeptide transporter	Oligopeptide transport system permease protein; OppB	
K15582	Oligopeptide transporter	Oligopeptide transport system permease protein; OppC	
K15583	Oligopeptide transporter	Oligopeptide transport system ATP-binding protein; OppD	
K10823	Oligopeptide transporter	Oligopeptide transport system ATP-binding protein; OppF	
K12368	Dipeptide transporter	Dipeptide transport system substrate-binding protein; DppA	
K12369	Dipeptide transporter	Dipeptide transport system permease protein; DppB	
K12370	Dipeptide transporter	Dipeptide transport system permease protein; DppC	
K12371	Dipeptide transporter	Dipeptide transport system ATP-binding protein; DppD	
K12372	Dipeptide transporter	Dipeptide transport system ATP-binding protein; DppF	
K16199	Dipeptide transporter	Dipeptide transport system substrate-binding protein; DppE	
K16200	Dipeptide transporter	Dipeptide transport system permease protein; DppB1	
K16201	Dipeptide transporter	Dipeptide transport system permease protein; DppC	
K16202	Dipeptide transporter	Dipeptide transport system ATP-binding protein; DppD	
K01216	Licheninase	Licheninase	EC:3.2.1.73
K01199	Glucan endo-1,3-beta-glucosidase	Glucan endo-1,3-beta-D-glucosidase	EC:3.2.1.39
K19891	Glucan endo-1,3-beta-glucosidase	Glucan endo-1,3-beta-glucosidase 1/2/3	EC:3.2.1.39
K19892	Glucan endo-1,3-beta-glucosidase	Glucan endo-1,3-beta-glucosidase 4	EC:3.2.1.39

K19893	Glucan endo-1,3-beta-glucosidase	Glucan endo-1,3-beta-glucosidase 5/6	EC:3.2.1.39
K01190	Beta-galactosidase	Beta-galactosidase; LacZ	EC:3.2.1.23
K12111	Beta-galactosidase	Evolved beta-galactosidase subunit alpha; EbgA	EC:3.2.1.23
K12308	Beta-galactosidase	Beta-galactosidase; LacA, BgaB	EC:3.2.1.23
K12309	Beta-galactosidase	Beta-galactosidase; GLB1, ELNR1	EC:3.2.1.23
K01188	Beta-galactosidase	Beta-glucosidase	EC:3.2.1.21
K05349	Beta-galactosidase	Beta-glucosidase; BglX	EC:3.2.1.21
K05350	Beta-galactosidase	Beta-glucosidase; BglB	EC:3.2.1.21
K01198	Xylan 1,4-beta-xylosidase	Xylan 1,4-beta-xylosidase; XynB	EC:3.2.1.37
K15920	Xylan 1,4-beta-xylosidase	Xylan 1,4-beta-xylosidase; XYL4	EC:3.2.1.37
K22268	Xylan 1,4-beta-xylosidase	Xylan 1,4-beta-xylosidase; XylA	EC:3.2.1.37
K01179	Cellulase/endoglucanase	Endoglucanase	EC:3.2.1.4
K19357	Cellulase/endoglucanase	Cellulase; CELB	EC:3.2.1.4
K20542	Cellulase/endoglucanase	Endoglucanase; BcsZ	EC:3.2.1.4
K01180	Laminarinase	Endo-1,3(4)-beta-glucanase	EC:3.2.1.6
K20846	Carrageenase	Kappa-carrageenase; CgkA	EC:3.2.1.83
K20850	Carrageenase	Iota-carrageenase; CgiA	EC:3.2.1.157
K01219	Agarase	Beta-agarase	EC:3.2.1.81
K20851	Agarase	Alpha-agarase; AgaA	EC:3.2.1.158
K01200	Pullulanase	Pullulanase; PulA	EC:3.2.1.41
K21575	Pullulanase	Neopullulanase; SusA	EC:3.2.1.135
K01177	Beta-amylase	Beta-amylase	EC:3.2.1.2
K01208	Maltogenic alpha-amylase	Cyclomaltodextrinase/maltogenic alpha-amylase/neopullulanase; Cd, Ma, NplT	EC:3.2.1.54, EC:3.2.1.133, EC:3.2.1.135
K05992	Maltogenic alpha-amylase	Maltogenic alpha-amylase; AmyM	EC:3.2.1.133
K22253	Exo-amylase	Glucan 1,4-alpha-maltotetraohydrolase; Mta	EC:3.2.1.60
K01178	Glucoamylase/glucan 1,4-alpha-glucosidase	Glucoamylase; SGA1	EC:3.2.1.3

K12047	Glucoamylase/glucan 1,4-alpha-glucosidase	Maltase-glucoamylase; MGAM	EC:3.2.1.20, EC:3.2.1.3
K21574	Glucoamylase/glucan 1,4-alpha-glucosidase	Glucan 1,4-alpha-glucosidase; SusB	EC:3.2.1.3
K07024	Sucrose-6-phosphatase	Sucrose-6-phosphatase	EC:3.1.3.24
K01193	Beta-fructofuranosidase	Beta-fructofuranosidase; INV, SacA	EC:3.2.1.26
K01225	Cellobiosidase	Cellulose 1,4-beta-cellobiosidase	EC:3.2.1.91
K19668	Cellobiosidase	Cellulose 1,4-beta-cellobiosidase; CbhA	EC:3.2.1.91
K08977	Bacterioruberin	Bisanhydrobacterioruberin hydratase	EC:4.2.1.161
K03821	PHA storage	Polyhydroxyalkanoate synthase subunit PhaC	EC:2.3.1.-
K05973	PHA storage	Poly(3-hydroxybutyrate) depolymerase; PhaZ	EC:3.1.1.75
K01428	Urea catabolism	Urease subunit alpha; UreC	EC:3.5.1.5
K01429	Urea catabolism	Urease subunit beta; UreB	EC:3.5.1.5
K01430	Urea catabolism	Urease subunit gamma; UreA	EC:3.5.1.5
K01941	Urea catabolism	Urea carboxylase	EC:6.3.4.6
K00111	Glycerol catabolism	Glycerol-3-phosphate dehydrogenase; GlpA, GlpD	EC:1.1.5.3
K00112	Glycerol catabolism	Glycerol-3-phosphate dehydrogenase subunit B; GlpB	EC:1.1.5.3
K00113	Glycerol catabolism	Glycerol-3-phosphate dehydrogenase subunit C; GlpC	EC:1.1.5.3
K00864	Glycerol catabolism	Glycerol kinase; GlpK, GK	EC:2.7.1.30
K00005	Glycerol catabolism	Glycerol dehydrogenase; GldA	EC:1.1.1.6
K00096	Archaeal glycerol synthesis	Glycerol-1-phosphate dehydrogenase [NAD(P)+]	EC:1.1.1.261
K00518	Superoxide dismutase	Nickel superoxide dismutase; SodN	EC:1.15.1.1
K04564	Superoxide dismutase	Superoxide dismutase, Fe-Mn family; SOD2	EC:1.15.1.1
K04565	Superoxide dismutase	Superoxide dismutase, Cu-Zn family; SOD1	EC:1.15.1.1
K16627	Superoxide dismutase	Superoxide dismutase, Cu-Zn family; SOD3	EC:1.15.1.1

K06162	Methylphosphonate catabolism	Alpha-D-ribose 1-methylphosphonate 5-triphosphate diphosphatase; PhnM	EC:3.6.1.63
K06163	Methylphosphonate catabolism	Alpha-D-ribose 1-methylphosphonate 5-phosphate C-P lyase; PhnJ	EC:4.7.1.1
K06164	Methylphosphonate catabolism	Alpha-D-ribose 1-methylphosphonate 5-triphosphate synthase subunit PhnI	EC:2.7.8.37
K06165	Methylphosphonate catabolism	Alpha-D-ribose 1-methylphosphonate 5-triphosphate synthase subunit PhnH	EC:2.7.8.37
K06166	Methylphosphonate catabolism	Alpha-D-ribose 1-methylphosphonate 5-triphosphate synthase subunit PhnG	EC:2.7.8.37
K05780	Methylphosphonate catabolism	Alpha-D-ribose 1-methylphosphonate 5-triphosphate synthase subunit PhnL	EC:2.7.8.37
K06167	Methylphosphonate catabolism	Phosphoribosyl 1,2-cyclic phosphate phosphodiesterase; PhnP	EC:3.1.4.55
K03430	Aminoethylphosphonate catabolism	2-Aminoethylphosphonate-pyruvate transaminase; PhnW	EC:2.6.1.37
K05306	Aminoethylphosphonate catabolism	Phosphonoacetaldehyde hydrolase; PhnX	EC:3.11.1.1
K07306	DMSO reduction	Anaerobic dimethyl sulfoxide reductase subunit A	EC:1.8.5.3
K20452	DMSO reduction	Dimethylmaleate hydratase large subunit; DmdA	EC:4.2.1.85
K16953	DMSP catabolism	Dimethylpropiothetin dethiomethylase; DddL	EC:4.4.1.3
K17486	DMSP catabolism	Dimethylsulfoniopropionate demethylase; DmdA	EC:2.1.1.269
K03553	RecA	Recombination protein; RecA	

Appendix G

Taxonomy verification of abundant OTUs in Ace Lake

Table G1. List of abundant OTUs identified in Ace Lake. The abundant OTUs were identified in Ace Lake metagenomes using the method described in Chapter 3 section 3.2.1. [A] The taxonomies of OTU bins were verified through their *16S rRNA* gene identities and ANIs to the reference genomes as well as their matches to Ace Lake MetaBAT MAGs (Chapter 3 section 3.2.2). [B] Original taxonomic classification refers to the protein taxonomies provided in the IMG Phylodist files, which were used for classifying contigs and OTUs. Some of these OTUs were merged and/or split based on the taxonomy verification output. The finalised OTU taxonomies are mentioned in the first column. [C] In each OTU bin, the contigs with matches to the red-highlighted MetaBAT MAGs were excluded from functional potential analysis of that OTU, due to insufficient number of genes being associated with these MAGs. The functional potentials of the algal OTU and the five viral OTUs were not analysed, therefore, the MetaBAT MAGs with matches to the *Micromonas* and *Phycodnaviridae* 1-5 OTUs are also highlighted in red. d, domain; p, phylum; c, class; o, order; f, family; g, genus; s, species. NA, not applicable; NM, no match.

OTUs[A]	Original taxonomic classification (reference genome accession ID) [B]	% ANI (% alignment fraction)	16S/18S SSU % identity	MetaBAT MAG matches[C]
Micromonas	*Micromonas commode* (NC_013038.1 - NC_013054.1)	75 (6)	NM	Bin919 Unclassified Bin1249 Unclassified Bin1079 Unclassified Bin282 Unclassified
	Micromonas pusilla (GCF_000151265.2)	75 (7)	NM	
Phycodnaviridae 1	Bathycoccus sp. RCC1105 virus BpV (NC_014765.1)	70 (18)	NA	Bin62 p_*Proteobacteria*
Phycodnaviridae 2	Micromonas sp. RCC1109 virus MpV1 (NC_014767.1)	76 (54)	NA	Bin62 p_*Proteobacteria*
Phycodnaviridae 3	Chrysochromulina ericina virus (GCF_001399245.1)	71 (5)	NA	Bin1350 d_Bacteria Bin1042 d_Bacteria Bin1755 d_Bacteria

				Bin97 d_Bacteria Bin1551 d_Bacteria Bin1998 d_Bacteria Bin784 d_Bacteria Bin2102 d_Bacteria Bin494 Unclassified Bin651 Unclassified Bin932 Unclassified Bin1852 Unclassified
Phycodnaviridae 4	Micromonas pusilla virus 12T (GCF_000906035.1)	75 (21)	NA	Bin62 p_*Proteobacteria*
Phycodnaviridae 5	Micromonas pusilla virus SP1 sensu lato		NA	Bin62 p_*Proteobacteria*
Algoriphagus	*Algoriphagus antarcticus* (GCF_002150685.1)	78 (46)	NM	Bin1943 g_*Algoriphagus*
Leadbetterella	*Cytophagales* bacterium TFI 002 (NZ_LT907983.1)	71 (19)	91	Bin277 g_*Leadbetterella*
Nonlabens	*Nonlabens xylanidelens* (GCF_002934445.1)	75 (47)	NM	Bin1375 g_*Nonlabens* Bin690 s_*Nonlabens dokdonensis*
	Nonlabens dokdonensis (GCF_000332115.1)	75 (43)	NM	
Saprospiraceae sp.	*Phaeodactylibacter xiamenensis* (GCF_000759025.1)	69 (6)	NM	Bin420 f_*Saprospiraceae*
Polaribacter	unclassified *Polaribacter* (NZ_LT629752.1)	84 (63)	NM	Bin1415 g_*Polaribacter* Bin385 g_*Polaribacter* Bin670 g_*Polaribacter*

	Polaribacter sp. KT25b (NZ_LT629752.1)	85 (68)	NM	Bin246 g_*Polaribacter* Bin574 g_*Polaribacter* Bin776 g_*Polaribacter*
Hydrogenophaga	*Hydrogenophaga crassostreae* (GCF_001640105.1)	79 (49)	NM	Bin22 g_*Hydrogenophaga*
	Hydrogenophaga taeniospiralis (GCF_001592305.1)	79 (49)	NM	
Burkholderiaceae MOLA814	*Betaproteobacteria* bacterium MOLA814 (GCF_000496475.1)	98 (94)	100	Bin1173 g_RS62
Loktanella	*Loktanella salsilacus* (GCF_900114485.1)	84 (77)	NM	Bin864 s_*Loktanella salsilacus*
Yoonia vestfoldensis	*Yoonia vestfoldensis* (strain SKA53 – GCF_000152785.1)	SKA53 – 93 (88)	SKA53 – 99.9	Bin1729 s_*Yoonia vestfoldensis*
	(strain DSM 16212 – GCF_000382265.1)	DSM 16212 – 86 (77)	DSM 16212 – 99	
Flavobacteriaceae MAG-120531	*Sediminicola* sp. YIK13 (GCF_001430825.1)	72 (37)	NM	Bin1744 g_MAG-120531 Bin896 g_MAG-120531
Burkholderiaceae SCGC-AAA027-K21	Beta proteobacterium MWH-P2sevCIIIb (GCF_003003055.1)	72 (25)	98	Bin1507 g_SCGC-AAA027-K21
Aquiluna	*Candidatus* Aquiluna sp. IMCC13023 (GCF_000257665.1)	87 (83)	100 99 99	Bin802 s_*Aquiluna sp1* Bin842 s_*Aquiluna sp1* Bin1781 s_*Aquiluna sp1*
Microbacteriaceae BACL25	*Mesorhizobium* sp. F7 (GCF_000798645.1)	72 (29)	NM	Bin1187 s_BACL25 sp1 Bin1172 s_BACL25 sp1 Bin1399 s_BACL25 sp1

	Microcella sp. HL-107 (GCF_002813345.1)	72 (27)	96 96	Bin534 s_BACL25 sp1
	Yonghaparkia sp. Root332 (GCF_001425665.1)	72 (31)	NM	
*Pseudomonas*_E	*Pseudomonas alcaliphila* (PA – GCF_900101755.1)	92 (82)	NM	Bin911 s_*Pseudomonas*_E *alcaliphila*
	Pseudomonas pseudoalcaligenes (PP – GCF_000297075.2)	96 (78)	NM	
	unclassified *Pseudomonas*	PA – 91 (68) PP – 94 (58)	NM	
Halioglobus	*Halioglobus pacificus* (GCF_001953075.1)	72 (37)	NM	Bin1377 g_*Halioglobus*
	Marine gamma proteobacterium HTCC2148 (GCF_000156295.1)	73 (32)	NM	
Methylophilaceae BACL14	*Methylophilales* bacterium HTCC2181 (GCF_000168995.1)	82 (92)	99	Bin470 s_BACL14 sp1
Porticoccaceae HTCC2207	gamma proteobacterium HTCC2207 (GCF_000153445.1)	75 (45)	97	Bin525 g_HTCC2207 Bin686 g_HTCC2207 Bin271 g_HTCC2207

Pseudohongiellaceae 1 *Pseudohongiellaceae* 2	*Pseudohongiella spirulinae* (GCF_001444425.1)	71 (18)	NM	Bin706 - g_OM182 Bin2107 - s_OM182 sp1
Pelagibacter	Candidatus *Pelagibacter ubique* (GCF_000504225.1)	77 (68)	99 99 92	Bin1939 s_*Pelagibacter ubique* Bin2016 s_*Pelagibacter ubique* Bin1535 s_*Pelagibacter ubique* Bin978 s_*Pelagibacter ubique*
	Candidatus *Pelagibacter* sp. IMCC9063 (GCF_000195085.1)	90 (90)	99.9 99.9	Bin1105 s_*Pelagibacter ubique* Bin1323 s_*Pelagibacter ubique* Bin887 s_*Pelagibacter ubique* Bin2004 s_*Pelagibacter ubique* Bin1518 s_*Pelagibacter ubique* Bin363 s_*Pelagibacter ubique* Bin1541 g_*Pelagibacter* Bin1782 g_*Pelagibacter* Bin1123 g_*Pelagibacter* Bin1666 g_*Pelagibacter* Bin1485 g_IMCC9063 Bin1036 g_IMCC9063
Verrucomicrobia Arctic95D-9	*Coraliomargarita akajimensis* (NC_014008.1)	73 (19)	95 94 84	Bin1608 g_*Haloferula* Bin1509 g_Arctic95D-9 Bin831 g_Arctic95D-9 Bin560 g_Arctic95D-9 Bin1278 g_BACL24
Verrucomicrobia BACL24	*Chthoniobacter flavus*	71 (6)	83	

Verrucomicrobia SW10 *Verrucomicrobia* UBA4506 *Haloferula*	(GCF_000173075.1)			Bin82 g_BACL24 Bin341 g_BACL24
	Haloferula sp. BvORR071 (GCF_000739615.1)	72 (18)	85 91	Bin1259 g_SW10 Bin1231 g_UBA4506 Bin1414 f_*Opitutaceae* Bin192 f_*Opitutaceae* Bin869 g_UBA6053
	Prosthecobacter debontii (GCF_900167535.1)	71 (7)	85 88	
	Rubritalea squalenifaciens (GCF_900141815.1)	72 (15)	88	
Gimesia	*Gimesia maris* (GCF_000181475.1)	76 (59)	98 98 98	Bin1542 s_*Gimesia maris* Bin1604 s_*Gimesia maris*
Crocinitomix	*Crocinitomix catalasitica* (GCF_000621625.1)	73 (34)	96	Bin223 g_*Crocinitomix*
Cyclobacterium	*Cyclobacterium qasimii* (GCF_000427295.1)	86 (82)	99	Bin1381 g_*Cyclobacterium*
Fabibacter	*Roseivirga spongicola* (GCF_001592965.1)	73 (47)	94	Bin155 s_*Fabibacter sp1*
Oligoflexus	*Pseudobacteriovorax antillogorgiicola* RKEM611 (GCF_900177345.1)	70(8)	NM	Bin927 s_*Oligoflexus tunisiensis* Bin255 s_*Oligoflexus tunisiensis*
Balneolaceae UBA2664	*Rhodohalobacter halophilus* (GCF_001715195.1)	72 (29)	94 94	Bin306 g_ UBA2664
Synechococcus sp. SynAce01	*Synechococcus* sp. SynAce01	99 (97)	99.9	Bin1724 g_*Cyanobium*

	(NZ_CP018091.1)			
Nisaea	alpha proteobacterium BAL199 (GCF_000171835.1)	80 (57)	98 97	Bin1427 g_BAL199 Bin283 g_BAL199
Chlorobium	*Chlorobium phaeovibrioides* DSM 265 (NC_009337.1)	85 (85)	99	Bin1268 s_*Chlorobium phaeovibrioides*
Izimaplasma	Candidatus *Izimaplasma* sp. HR2 (GCF_000753575.1)	75 (54)	NM	Bin1380 g_*Izimaplasma*
Bacteroidales UBA4459	*Lentimicrobium saccharophilum* (GCF_001192835.1)	70 (11)	90 89 89 89 88 88	Bin1394 g_ UBA4459
Desulfobacterium	*Desulfobacterium vacuolatum* (GCF_900176365.1)	82 (41)	98 98	Bin703 g_*Desulfobacterium* Bin1072 g_*Desulfobacterium*
Desulfocapsa	*Desulfocapsa sulfexigens* (NC_020304.1)	78 (63)	97 97 97	Bin20 s_*Desulfocapsa sulfexigens* Bin2043 s_*Desulfocapsa sulfexigens* Bin134 s_*Desulfocapsa sulfexigens*
Desulfatiglanales NaphS2	delta proteobacterium NaphS2 (GCF_000179315.1)	75 (34)	97 93 93 93	Bin2047 g_NaphS2 Bin505 g_NaphS2 Bin1224 g_NaphS2

Desulfobacterales S5133MH16	*Desulfosarcina* sp. BuS5 (GCF_000472805.1)	74 (31)	93	Bin1209 g_S5133MH16 Bin1110 g_S5133MH16 Bin1728 g_S5133MH16 Bin2047 g_NaphS2 Bin505 g_NaphS2
Syntrophales UBA2210	*Syntrophus aciditrophicus* (NC_007759.1)	70 (17)	92 92	Bin2060 g_UBA2210 Bin962 s_UBA2210 sp1 Bin899 g_UBA6078
	Syntrophus gentianae (GCF_900109885.1)	71 (26)	92	
	Smithella sp. F21 (GCF_000747085.1)	71 (35)	NM	
	Smithella sp. SCADC (GCF_000747625.1)	70 (20)	91 91 91	
Atribacteria 34-128	unclassified *Atribacteria* (GCA_001509285.1)	81 (14)	NM	Bin894 g_34-128 Bin1182 g_34-128 Bin866 g_34-128 Bin2083 g_34-128 Bin1876 p__*Firmicutes*
	Atribacteria bacterium JGI 0000014-F07 (GCA_001509285.1)	81 (18)	NM	
Cloacimonetes JGIOTU-2	unclassified *Cloacimonetes* (JGI OUT-2 – GCF_000493905.1) (TCS61 – GCA_001577125.1)	JGI OUT-2 – 81 (22) TCS61 – 71 (5)	NM	Bin1703 s_JGIOTU-2 sp1 Bin1683 s_JGIOTU-2 sp1 Bin1264 s_JGIOTU-2 sp1 Bin2003 f_TCS61 Bin1346 g_TCS61
Methanomicrobiaceae 1	*Methanoplanus limicola* (GCF_000243255.1)	73 (27)	94	Bin1205 f_*Methanomicrobiaceae*

				Bin2059 s_*Methanomicrobium mobile* Bin1141 f_*Methanomicrobiaceae*
*Methanothrix*_A	*Methanosaeta harundinacea* (GCF_000235565.1)	73 (22)	NM	Bin23 g_*Methanothrix*_A
Parcubacteria	unclassified *Parcubacteria*	NA	NA	Bin1642 g_UBA6065 Bin1194 o_UBA9983 Bin1725 g_2-02-FULL-39-13 Bin1572 g_UBA2196 Bin2081 c_ABY1 Bin1304 s_2-12-FULL-45-10 sp1

Appendix H

Viral data

Table H1. List of specific viral contigs identified in Antarctic metagenomes. [A] The viral cluster or singleton designations were assigned to the viral contigs using the data in the Antarctic virus catalogue. The viral contigs that were not a part of the Antarctic virus catalogue could not be assigned a viral cluster of singleton designation. [B] The viral clades that were not determined by JGI's IMG system were referred to as 'Unknown'. [C] The IMG taxon IDs refer to the metagenome IDs allotted by JGI's IMG system. [D] The complete, circular virus genomes were analysed using the method described in Chapter 3 section 3.2.6.4. The orange-highlighted viral contigs represent a 'huge' phage that contained *cas* genes (Chapter 3 sections 3.2.6.5 and 3.3.5.2) and are also included under the 'Huge' phage genome contigs section in this table. [E] The blue-highlighted cyanophage contig was used for the analysis of potential *Synechococcus* viruses (Chapter 3 sections 3.2.6.2 and 3.3.5.5). **[F]** Various *Chlorobium*-associated viruses were identified among the viral contigs in the Antarctic virus catalogue (Chapter 3 section 3.2.6.1 and Chapter 5 section 5.2.5**)**. Additional potential *Chlorobium* viruses were also identified during the analysis of high abundance Ace Lake viral clusters (Chapter 3 sections 3.2.6.6 and 3.3.5.3). **[H]** 'Huge' phage genome sequences were identified during the analysis of complete circular viruses (orange-highlighted contigs) (Chapter 3 sections 3.2.6.5 and 3.3.5.2). * 'Huge' phage contigs Ga0222637_1000003 and Ga0222637_1000005 from Nov 2013_L1_0.1 µm metagenome together represented the complete phage genome. All other 'huge' phage contigs individually represented the complete phage genome. Ace Lake depths: U1, Upper 1; U2, Upper 2; U3, Upper 3; I, Interface; L1, Lower 1; L2, Lower 2; L3, Lower 3. Filter fractions: 3, 3–20 µm; 0.8, 0.8–3 µm; 0.1, 0.1–0.8 µm. NA, not applicable.

Viral cluster or singleton[A]	Clade[B]	Contig ID	Ace Lake zone	Metagenome (sample collection date, depth, filter fraction)	IMG taxon ID[C]	Contig length (bp)	GC content	Read depth
Complete, circular virus genomes[D]								
cl_39	*Caudovirales*	Ga0208413_1000137	Upper	Nov 2008_U2_0.8 µm	3300025513	61,327	0.4	215
	Caudovirales	Ga0208414_1000255	Upper	Nov 2008_U3_0.8 µm	3300025603	61,327	0.4	254
cl_61	*Caudovirales*	Ga0222634_1000074	Upper	Nov 2013_U3_0.1 µm	3300023235	60,667	0.34	59
cl_86	*Caudovirales*	Ga0222644_1000071	Upper	Dec 2013_U1_3 µm	3300022841	57,734	0.34	17
	Caudovirales	Ga0222646_100034	Upper	Dec 2013_U1_0.1 µm	3300022822	57,734	0.34	19
	Caudovirales	Ga0222652_1000057	Upper	Jul 2014_U2_0.1 µm	3300022853	57,728	0.34	49
	Caudovirales	Ga0222711_1000019	Upper	27Jan 2015_U1_0.1 µm	3300022837	57,728	0.34	32
	Caudovirales	Ga0222668_1000121	Upper	Oct 2014_U2_3 µm	3300022865	57,720	0.34	32
	Caudovirales	Ga0222674_1000083	Upper	Oct 2014_U3_3 µm	3300022848	57,720	0.34	34
cl_88	*Caudovirales*	Ga0222631_1000001	Upper	Nov 2013_U2_0.1 µm	3300022843	198,785	0.36	128
cl_94	*Caudovirales*	Ga0222649_1000036	Upper	Feb 2014_U1_0.1 µm	3300022839	56,488	0.59	638
cl_182	*Caudovirales*	Ga0222686_1000050	Upper	Dec 2014_U2_3 µm	3300023501	61,266	0.48	46
cl_190	*Caudovirales*	Ga0208414_1000068	Upper	Nov 2008_U3_0.8 µm	3300025603	116,917	0.43	35

	Caudovirales	Ga0222630_1000008	Upper	Nov 2013_U2_0.8 µm	3300023243	110,308	0.43	96
	Caudovirales	Ga0222645_100024	Upper	Dec 2013_U1_0.8 µm	3300022833	110,283	0.43	31
cl_294	*Caudovirales*	Ga0222651_1000297	Upper	Jul 2014_U2_0.8 µm	3300022866	35,773	0.37	27
cl_355	*Caudovirales*	Ga0222631_1000054	Upper	Nov 2013_U2_0.1 µm	3300022843	34,439	0.49	272
	Caudovirales	Ga0222652_1000162	Upper	Jul 2014_U2_0.1 µm	3300022853	34,439	0.49	176
cl_415	*Caudovirales*	Ga0222676_1000131	Upper	Oct 2014_U3_0.1 µm	3300023240	32,297	0.47	336
	Caudovirales	Ga0222654_1000364	Upper	Jul 2014_U3_0.8 µm	3300022836	32,288	0.47	71
cl_440	*Caudovirales*	Ga0222648_1000052	Upper	Feb 2014_U1_0.8 µm	3300023054	60,542	0.52	135
cl_468	*Caudovirales*	Ga0208768_1000361	Upper	Nov 2008_U2_3 µm	3300025601	31,641	0.43	19
cl_548	*Caudovirales*	Ga0222652_1000145	Upper	Jul 2014_U2_0.1 µm	3300022853	36,399	0.33	91
	Caudovirales	Ga0222659_1000274	Upper	Aug 2014_U2_3 µm	3300023236	36,399	0.33	61
	Caudovirales	Ga0222660_1000153	Upper	Aug 2014_U2_0.8 µm	3300023239	36,399	0.33	40
	Caudovirales	Ga0222670_1000118	Upper	Oct 2014_U2_0.1 µm	3300023294	36,399	0.33	287
	Caudovirales	Ga0222676_1000104	Upper	Oct 2014_U3_0.1 µm	3300023240	36,399	0.33	190
	Caudovirales	Ga0222689_1000137	Upper	Dec 2014_U3_3 µm	3300023231	36,399	0.33	28
	Caudovirales	Ga0222711_1000047	Upper	27Jan 2015_U1_0.1 µm	3300022837	36,399	0.33	156

cl_563	*Caudovirales*	Ga0222663_1000084	Upper	Aug 2014_U3_0.8 µm	3300022845	71,162	0.34	21
	Caudovirales	Ga0222676_1000026	Upper	Oct 2014_U3_0.1 µm	3300023240	71,162	0.34	36
	Caudovirales	Ga0222690_1000051	Upper	Dec 2014_U3_0.8 µm	3300023227	71,162	0.34	29
	Caudovirales	Ga0222691_1000031	Upper	Dec 2014_U3_0.1 µm	3300022851	71,162	0.34	16
cl_578	*Caudovirales*	Ga0208768_1000193	Upper	Nov 2008_U2_3 µm	3300025601	46,295	0.39	57
	Caudovirales	Ga0222711_1000027	Upper	27Jan 2015_U1_0.1 µm	3300022837	43,333	0.39	84
cl_614	*Caudovirales*	Ga0222629_1000113	Upper	Nov 2013_U2_3 µm	3300022867	32,140	0.38	16
	Caudovirales	Ga0222631_1000063	Upper	Nov 2013_U2_0.1 µm	3300022843	32,140	0.38	146
	Caudovirales	Ga0222652_1000183	Upper	Jul 2014_U2_0.1 µm	3300022853	32,140	0.38	94
	Caudovirales	Ga0222661_1000169	Upper	Aug 2014_U2_0.1 µm	3300023229	32,140	0.38	40
	Caudovirales	Ga0222711_1000069	Upper	27Jan 2015_U1_0.1 µm	3300022837	32,138	0.38	17
	Caudovirales	Ga0208646_1000327	Upper	Nov 2008_U2_0.1 µm	3300025425	31,918	0.38	73
	Caudovirales	Ga0208770_1000334	Upper	Nov 2008_U3_0.1 µm	3300025438	31,918	0.38	59
cl_619	*Caudovirales*	Ga0222629_1000065	Upper	Nov 2013_U2_3 µm	3300022867	40,762	0.34	43
cl_685	*Caudovirales*	Ga0222630_1000110	Upper	Nov 2013_U2_0.8 µm	3300023243	41,001	0.34	78
	Caudovirales	Ga0222644_1000115	Upper	Dec 2013_U1_3 µm	3300022841	41,001	0.34	20

	Caudovirales	Ga0222676_1000088	Upper	Oct 2014_U3_0.1 µm	3300023240	41,001	0.34	66
	Caudovirales	Ga0222688_1000030	Upper	Dec 2014_U2_0.1 µm	3300023293	41,001	0.34	36
	Caudovirales	Ga0222691_1000082	Upper	Dec 2014_U3_0.1 µm	3300022851	41,001	0.34	77
	Caudovirales	Ga0222711_1000034	Upper	27Jan 2015_U1_0.1 µm	3300022837	41,001	0.34	18
	Caudovirales	Ga0222631_1000036	Upper	Nov 2013_U2_0.1 µm	3300022843	40,999	0.34	139
	Caudovirales	Ga0222652_1000113	Upper	Jul 2014_U2_0.1 µm	3300022853	40,988	0.34	121
	Caudovirales	Ga0222661_1000105	Upper	Aug 2014_U2_0.1 µm	3300023229	40,921	0.34	69
	Caudovirales	Ga0222675_1000133	Upper	Oct 2014_U3_0.8 µm	3300023238	40,899	0.34	16
cl_723	*Caudovirales*	Ga0222669_1000080	Upper	Oct 2014_U2_0.8 µm	3300022825	34,304	0.6	39
cl_727	*Caudovirales*	Ga0222690_1000105	Upper	Dec 2014_U3_0.8 µm	3300023227	31,608	0.3	60
cl_811	*Caudovirales*	Ga0222651_1000265	Upper	Jul 2014_U2_0.8 µm	3300022866	37,577	0.38	129
	Caudovirales	Ga0222652_1000135	Upper	Jul 2014_U2_0.1 µm	3300022853	37,577	0.38	231
	Caudovirales	Ga0222653_1000212	Upper	Jul 2014_U3_3 µm	3300022857	37,577	0.38	188
	Caudovirales	Ga0222659_1000254	Upper	Aug 2014_U2_3 µm	3300023236	37,577	0.38	78
	Caudovirales	Ga0222660_1000148	Upper	Aug 2014_U2_0.8 µm	3300023239	37,577	0.38	95
	Caudovirales	Ga0222661_1000134	Upper	Aug 2014_U2_0.1 µm	3300023229	37,577	0.38	147

	Caudovirales	Ga0222668_1000262	Upper	Oct 2014_U2_3 µm	3300022865	37,577	0.38	112
	Caudovirales	Ga0222669_1000067	Upper	Oct 2014_U2_0.8 µm	3300022825	37,577	0.38	34
	Caudovirales	Ga0222670_1000107	Upper	Oct 2014_U2_0.1 µm	3300023294	37,577	0.38	106
	Caudovirales	Ga0222674_1000132	Upper	Oct 2014_U3_3 µm	3300022848	37,577	0.38	86
	Caudovirales	Ga0222675_1000143	Upper	Oct 2014_U3_0.8 µm	3300023238	37,577	0.38	44
	Caudovirales	Ga0222676_1000100	Upper	Oct 2014_U3_0.1 µm	3300023240	37,577	0.38	50
	Caudovirales	Ga0222686_1000106	Upper	Dec 2014_U2_3 µm	3300023501	37,577	0.38	58
	Caudovirales	Ga0222687_1000052	Upper	Dec 2014_U2_0.8 µm	3300022844	37,577	0.38	27
	Caudovirales	Ga0222688_1000043	Upper	Dec 2014_U2_0.1 µm	3300023293	37,577	0.38	63
cl_814	*Caudovirales*	Ga0222674_1000138	Upper	Oct 2014_U3_3 µm	3300022848	36,364	0.5	23
cl_834	*Caudovirales*	Ga0222664_1000268	Upper	Aug 2014_U3_0.1 µm	3300023296	36,977	0.57	166
	Caudovirales	Ga0222676_1000102	Upper	Oct 2014_U3_0.1 µm	3300023240	36,977	0.57	420
cl_843	*Caudovirales*	Ga0208414_1000192	Upper	Nov 2008_U3_0.8 µm	3300025603	69,370	0.43	22
	Caudovirales	Ga0208646_1000071	Upper	Nov 2008_U2_0.1 µm	3300025425	69,370	0.43	91
cl_922	*Caudovirales*	Ga0222634_1000130	Upper	Nov 2013_U3_0.1 µm	3300023235	44,118	0.35	15
cl_925	*Caudovirales*	Ga0208646_1000125	Upper	Nov 2008_U2_0.1 µm	3300025425	55,014	0.43	82

cl_936	*Caudovirales*	Ga0208768_1000068	Upper	Nov 2008_U2_3 µm	3300025601	82,754	0.3	555
	Caudovirales	Ga0208646_1000042	Upper	Nov 2008_U2_0.1 µm	3300025425	82,750	0.3	137
	Caudovirales	Ga0208770_1000054	Upper	Nov 2008_U3_0.1 µm	3300025438	82,750	0.3	94
	Caudovirales	Ga0208903_1000111	Upper	Nov 2008_U3_3 µm	3300025502	82,721	0.3	944
cl_960	*Caudovirales*	Ga0222688_1000057	Upper	Dec 2014_U2_0.1 µm	3300023293	33,844	0.45	247
cl_961	*Caudovirales*	Ga0222644_1000054	Upper	Dec 2013_U1_3 µm	3300022841	65,450	0.35	229
cl_1134	*Caudovirales*	Ga0222652_1000107	Upper	Jul 2014_U2_0.1 µm	3300022853	42,580	0.56	29
cl_1234	*Caudovirales*	Ga0222649_1000109	Upper	Feb 2014_U1_0.1 µm	3300022839	32,238	0.45	192
cl_1255	*Caudovirales*	Ga0208414_1000468	Upper	Nov 2008_U3_0.8 µm	3300025603	40,133	0.45	20
	Caudovirales	Ga0208646_1000215	Upper	Nov 2008_U2_0.1 µm	3300025425	40,133	0.45	105
	Caudovirales	Ga0208768_1000236	Upper	Nov 2008_U2_3 µm	3300025601	40,133	0.45	23
	Caudovirales	Ga0208770_1000240	Upper	Nov 2008_U3_0.1 µm	3300025438	40,133	0.45	91
	Caudovirales	Ga0208903_1000351	Upper	Nov 2008_U3_3 µm	3300025502	40,133	0.45	25
cl_1303	*Caudovirales*	Ga0222652_1000051	Upper	Jul 2014_U2_0.1 µm	3300022853	59,036	0.46	57
	Caudovirales	Ga0222688_1000009	Upper	Dec 2014_U2_0.1 µm	3300023293	59,036	0.46	80
	Caudovirales	Ga0222691_1000045	Upper	Dec 2014_U3_0.1 µm	3300022851	59,036	0.46	73

	Caudovirales	Ga0222661_1000055	Upper	Aug 2014_U2_0.1 µm	3300023229	58,976	0.46	35
cl_1389	*Caudovirales*	Ga0222633_1000195	Upper	Nov 2013_U3_0.8 µm	3300022847	67,914	0.56	25
	Caudovirales	Ga0222632_1000099	Upper	Nov 2013_U3_3 µm	3300022842	67,839	0.56	69
	Caudovirales	Ga0222634_1000056	Upper	Nov 2013_U3_0.1 µm	3300023235	67,839	0.56	44
cl_1609	*Caudovirales*	Ga0208770_1000293	Upper	Nov 2008_U3_0.1 µm	3300025438	35,550	0.41	18
	Caudovirales	Ga0208646_1000275	Upper	Nov 2008_U2_0.1 µm	3300025425	35,530	0.41	22
cl_1614	*Caudovirales*	Ga0208414_1000465	Upper	Nov 2008_U3_0.8 µm	3300025603	40,262	0.56	26
cl_1687	*Caudovirales*	Ga0222649_1000033	Upper	Feb 2014_U1_0.1 µm	3300022839	60,587	0.36	151
cl_1925	*Caudovirales*	Ga0222670_1000134	Upper	Oct 2014_U2_0.1 µm	3300023294	34,611	0.37	43
cl_1931	*Caudovirales*	Ga0222629_1000098	Upper	Nov 2013_U2_3 µm	3300022867	35,224	0.4	30
	Caudovirales	Ga0222631_1000049	Upper	Nov 2013_U2_0.1 µm	3300022843	35,224	0.4	67
	Caudovirales	Ga0222652_1000154	Upper	Jul 2014_U2_0.1 µm	3300022853	35,224	0.4	40
cl_2074	*Caudovirales*	Ga0222632_1000124	Upper	Nov 2013_U3_3 µm	3300022842	57,505	0.55	42
cl_2122	*Caudovirales*	Ga0222651_1000121	Upper	Jul 2014_U2_0.8 µm	3300022866	58,728	0.36	32
cl_2251	*Caudovirales*	Ga0222633_1000388	Upper	Nov 2013_U3_0.8 µm	3300022847	42,580	0.44	31
cl_2260	*Caudovirales*	Ga0222660_1000154	Upper	Aug 2014_U2_0.8 µm	3300023239	36,315	0.31	27

cl_2466	*Caudovirales*	Ga0208770_1000066	Upper	Nov 2008_U3_0.1 µm	3300025438	75,636	0.42	16
cl_2664	*Caudovirales*	Ga0222646_100080	Upper	Dec 2013_U1_0.1 µm	3300022822	35,853	0.42	28
	Caudovirales	Ga0222652_1000149	Upper	Jul 2014_U2_0.1 µm	3300022853	35,853	0.42	57
	Caudovirales	Ga0222661_1000141	Upper	Aug 2014_U2_0.1 µm	3300023229	35,853	0.42	35
	Caudovirales	Ga0222670_1000120	Upper	Oct 2014_U2_0.1 µm	3300023294	35,853	0.42	45
	Caudovirales	Ga0222688_1000048	Upper	Dec 2014_U2_0.1 µm	3300023293	35,853	0.42	68
	Caudovirales	Ga0222711_1000048	Upper	27Jan 2015_U1_0.1 µm	3300022837	35,853	0.42	30
cl_3129	*Caudovirales*	Ga0222650_1000104	Upper	Jul 2014_U2_3 µm	3300023237	76,097	0.33	56
cl_3153	*Caudovirales*	Ga0222631_1000051	Upper	Nov 2013_U2_0.1 µm	3300022843	34,975	0.33	60
	Caudovirales	Ga0222652_1000157	Upper	Jul 2014_U2_0.1 µm	3300022853	34,975	0.33	21
cl_3169	*Caudovirales*	Ga0222688_1000054	Upper	Dec 2014_U2_0.1 µm	3300023293	35,117	0.39	17
cl_3187	*Caudovirales*	Ga0222650_1000438	Upper	Jul 2014_U2_3 µm	3300023237	32,633	0.46	29
	Caudovirales	Ga0222652_1000180	Upper	Jul 2014_U2_0.1 µm	3300022853	32,633	0.46	18
cl_3592	*Caudovirales*	Ga0208414_1000754	Upper	Nov 2008_U3_0.8 µm	3300025603	28,792	0.33	108
cl_3890	*Caudovirales*	Ga0222691_1000046	Upper	Dec 2014_U3_0.1 µm	3300022851	58,779	0.38	46
cl_3895	*Caudovirales*	Ga0222691_1000080	Upper	Dec 2014_U3_0.1 µm	3300022851	41,353	0.32	23

cl_3927	*Caudovirales*	Ga0222649_1000115	Upper	Feb 2014_U1_0.1 µm	3300022839	30,893	0.33	33
	Caudovirales	Ga0222644_1000166	Upper	Dec 2013_U1_3 µm	3300022841	30,887	0.33	138
cl_3933	*Caudovirales*	Ga0222650_1000363	Upper	Jul 2014_U2_3 µm	3300023237	37,577	0.38	253
	Caudovirales	Ga0222654_1000296	Upper	Jul 2014_U3_0.8 µm	3300022836	37,577	0.38	65
	Caudovirales	Ga0222655_1000141	Upper	Jul 2014_U3_0.1 µm	3300023245	37,577	0.38	130
	Caudovirales	Ga0222630_1000128	Upper	Nov 2013_U2_0.8 µm	3300023243	37,509	0.38	22
cl_3945	*Caudovirales*	Ga0222689_1000107	Upper	Dec 2014_U3_3 µm	3300023231	46,233	0.53	57
cl_3980	*Caudovirales*	Ga0222670_1000135	Upper	Oct 2014_U2_0.1 µm	3300023294	34,523	0.62	27
	Caudovirales	Ga0222688_1000055	Upper	Dec 2014_U2_0.1 µm	3300023293	34,523	0.62	61
	Caudovirales	Ga0222661_1000151	Upper	Aug 2014_U2_0.1 µm	3300023229	34,520	0.62	24
cl_4928	*Caudovirales*	Ga0222629_1000080	Upper	Nov 2013_U2_3 µm	3300022867	37,577	0.38	24
cl_4964	*Caudovirales*	Ga0222646_100030	Upper	Dec 2013_U1_0.1 µm	3300022822	61,566	0.42	50
cl_4974	*Caudovirales*	Ga0222707_1000108	Upper	8Jan 2015_U1_0.8 µm	3300022832	35,794	0.31	16
	Caudovirales	Ga0222708_1000050	Upper	8Jan 2015_U1_0.1 µm	3300023242	35,794	0.31	350
	Caudovirales	Ga0222711_1000050	Upper	27Jan 2015_U1_0.1 µm	3300022837	35,794	0.31	59
cl_4978	*Caudovirales*	Ga0222646_100095	Upper	Dec 2013_U1_0.1 µm	3300022822	31,728	0.32	33

cl_5051	Unknown	Ga0222691_1000044	Upper	Dec 2014_U3_0.1 µm	3300022851	59,477	0.56	29
cl_6587	*Caudovirales*	Ga0222708_1000007	Upper	8Jan 2015_U1_0.1 µm	3300023242	76,295	0.63	32
cl_6662	*Caudovirales*	Ga0222646_100076	Upper	Dec 2013_U1_0.1 µm	3300022822	36,531	0.4	15
cl_6676	*Caudovirales*	Ga0222629_1000100	Upper	Nov 2013_U2_3 µm	3300022867	34,975	0.33	54
cl_6750	*Caudovirales*	Ga0222664_1000144	Upper	Aug 2014_U3_0.1 µm	3300023296	55,212	0.37	261
cl_8545	*Caudovirales*	Ga0208768_1000228	Upper	Nov 2008_U2_3 µm	3300025601	40,839	0.39	45
cl_9600	*Caudovirales*	Ga0222689_1000136	Upper	Dec 2014_U3_3 µm	3300023231	36,717	0.41	86
cl_9840	*Caudovirales*	Ga0222644_1000101	Upper	Dec 2013_U1_3 µm	3300022841	44,811	0.59	134
cl_10110	*Caudovirales*	Ga0222707_1000112	Upper	8Jan 2015_U1_0.8 µm	3300022832	35,037	0.56	30
cl_10239	*Caudovirales*	Ga0222670_1000558	Upper	Oct 2014_U2_0.1 µm	3300023294	15,420	0.56	75
sg_8813	*Caudovirales*	Ga0222646_100052	Upper	Dec 2013_U1_0.1 µm	3300022822	44,962	0.51	21
sg_8814	*Caudovirales*	Ga0222646_100054	Upper	Dec 2013_U1_0.1 µm	3300022822	44,811	0.59	151
sg_8907	*Caudovirales*	Ga0222645_100134	Upper	Dec 2013_U1_0.8 µm	3300022833	44,811	0.59	27
sg_9264	Unknown	Ga0222691_1000034	Upper	Dec 2014_U3_0.1 µm	3300022851	68,930	0.53	73
sg_9323	*Caudovirales*	Ga0222652_1000161	Upper	Jul 2014_U2_0.1 µm	3300022853	34,523	0.62	45
sg_9693	*Caudovirales*	Ga0222662_1000451	Upper	Aug 2014_U3_3 µm	3300022885	33,379	0.48	34

NA	Unknown	Ga0222634_1000134	Upper	Nov 2013_U3_0.1 µm	3300023235	43,608	0.55	26
NA	*Caudovirales*	Ga0222629_1000190	Upper	Nov 2013_U2_3 µm	3300022867	23,953	0.29	43
	Caudovirales	Ga0222630_1000353	Upper	Nov 2013_U2_0.8 µm	3300023243	23,953	0.29	64
	Caudovirales	Ga0222649_1000173	Upper	Feb 2014_U1_0.1 µm	3300022839	23,953	0.29	26
	Caudovirales	Ga0222650_1000687	Upper	Jul 2014_U2_3 µm	3300023237	23,953	0.29	23
	Caudovirales	Ga0222652_1000292	Upper	Jul 2014_U2_0.1 µm	3300022853	23,953	0.29	255
	Caudovirales	Ga0222655_1000302	Upper	Jul 2014_U3_0.1 µm	3300023245	23,953	0.29	116
	Caudovirales	Ga0222661_1000259	Upper	Aug 2014_U2_0.1 µm	3300023229	23,953	0.29	65
	Caudovirales	Ga0222670_1000261	Upper	Oct 2014_U2_0.1 µm	3300023294	23,953	0.29	48
	Caudovirales	Ga0222676_1000205	Upper	Oct 2014_U3_0.1 µm	3300023240	23,953	0.29	44
NA	Retrovirales	Ga0222647_1001013	Upper	Feb 2014_U1_3 µm	3300022827	9,276	0.52	47
NA	Microviridae	Ga0222662_1004683	Upper	Aug 2014_U3_3 µm	3300022885	4,292	0.49	28
	Microviridae	Ga0222663_1002047	Upper	Aug 2014_U3_0.8 µm	3300022845	4,292	0.49	35
	Microviridae	Ga0222675_1003300	Upper	Oct 2014_U3_0.8 µm	3300023238	4,292	0.49	13
	Microviridae	Ga0222690_1001523	Upper	Dec 2014_U3_0.8 µm	3300023227	4,292	0.49	14
NA	CressDNAParvo	Ga0222633_1005399	Upper	Nov 2013_U3_0.8 µm	3300022847	3,562	0.5	151

NA	CressDNAParvo	Ga0222648_1005368	Upper	Feb 2014_U1_0.8 µm	3300023054	3,536	0.49	32
NA	CressDNAParvo	Ga0222645_105175	Upper	Dec 2013_U1_0.8 µm	3300022833	3,122	0.51	14
NA	CressDNAParvo	Ga0222630_1006513	Upper	Nov 2013_U2_0.8 µm	3300023243	3,115	0.5	20
NA	CressDNAParvo	Ga0222633_1006260	Upper	Nov 2013_U3_0.8 µm	3300022847	3,081	0.49	39
	CressDNAParvo	Ga0222644_1002881	Upper	Dec 2013_U1_3 µm	3300022841	3,081	0.5	77
	CressDNAParvo	Ga0222645_105245	Upper	Dec 2013_U1_0.8 µm	3300022833	3,081	0.5	168
NA	CressDNAParvo	Ga0208413_1024040	Upper	Nov 2008_U2_0.8 µm	3300025513	2,137	0.39	109
	CressDNAParvo	Ga0208768_1018460	Upper	Nov 2008_U2_3 µm	3300025601	2,137	0.39	279
	CressDNAParvo	Ga0222648_1012621	Upper	Feb 2014_U1_0.8 µm	3300023054	2,137	0.39	63
	CressDNAParvo	Ga0222668_1009307	Upper	Oct 2014_U2_3 µm	3300022865	2,137	0.39	9
	CressDNAParvo	Ga0222707_1004304	Upper	8Jan 2015_U1_0.8 µm	3300022832	2,137	0.39	331
	CressDNAParvo	Ga0222710_1007623	Upper	27Jan 2015_U1_0.8 µm	3300023429	2,137	0.39	84
	CressDNAParvo	Ga0222644_1005372	Upper	Dec 2013_U1_3 µm	3300022841	2,074	0.39	33
NA	Unknown	Ga0222686_1006371	Upper	Dec 2014_U2_3 µm	3300023501	2,033	0.47	30
	Unknown	Ga0222687_1009818	Upper	Dec 2014_U2_0.8 µm	3300022844	2,033	0.48	50
	Unknown	Ga0222690_1005126	Upper	Dec 2014_U3_0.8 µm	3300023227	2,033	0.48	8

cl_711	*Caudovirales*	Ga0208414_1000366	Upper, Interface	Nov 2008_U3_0.8 µm	3300025603	48,437	0.61	110
	Caudovirales	Ga0208770_1000171	Upper, Interface	Nov 2008_U3_0.1 µm	3300025438	48,437	0.61	71
	Caudovirales	Ga0208903_1000269	Upper, Interface	Nov 2008_U3_3 µm	3300025502	48,437	0.61	177
	Caudovirales	Ga0222673_1000112	Upper, Interface	Oct 2014_I_0.1 µm	3300022821	48,437	0.61	31
cl_1926	*Caudovirales*	Ga0222631_1000047	Upper, Interface	Nov 2013_U2_0.1 µm	3300022843	35,853	0.42	75
	Caudovirales	Ga0222634_1000201	Upper, Interface	Nov 2013_U3_0.1 µm	3300023235	35,853	0.42	1023
	Caudovirales	Ga0222664_1000276	Upper, Interface	Aug 2014_U3_0.1 µm	3300023296	35,853	0.42	144
	Caudovirales	Ga0222676_1000105	Upper, Interface	Oct 2014_U3_0.1 µm	3300023240	35,853	0.42	120
	Caudovirales	Ga0222691_1000108	Upper, Interface	Dec 2014_U3_0.1 µm	3300022851	35,853	0.42	426
cl_5848	*Caudovirales*	Ga0222671_1000145	Upper, Interface	Oct 2014_I_3 µm	3300022856	36,717	0.41	20
	Caudovirales	Ga0222674_1000135	Upper, Interface	Oct 2014_U3_3 µm	3300022848	36,717	0.41	26
cl_169	*Caudovirales*	Ga0222673_1000089	Interface	Oct 2014_I_0.1 µm	3300022821	56,215	0.52	43
cl_388	*Caudovirales*	Ga0222667_1000128	Interface	Aug 2014_I_0.1 µm	3300022890	43,076	0.39	19
cl_735	*Caudovirales*	Ga0222628_1000134	Interface	Nov 2013_I_0.1 µm	3300022871	37,031	0.34	45
cl_868	*Caudovirales*	Ga0222658_1000119	Interface	Jul 2014_I_0.1 µm	3300023257	49,083	0.39	48
	Caudovirales	Ga0222667_1000096	Interface	Aug 2014_I_0.1 µm	3300022890	49,083	0.39	54

cl_1230	*Caudovirales*	Ga0222664_1000178	Interface	Aug 2014_U3_0.1 µm	3300023296	47,643	0.39	31
	Caudovirales	Ga0222673_1000118	Interface	Oct 2014_I_0.1 µm	3300022821	47,643	0.39	67
cl_1480	*Caudovirales*	Ga0222665_1000014	Interface	Aug 2014_I_3 µm	3300022864	116,028	0.53	23
cl_2193	*Caudovirales*	Ga0222666_1000279	Interface	Aug 2014_I_0.8 µm	3300024048	34,155	0.46	21
cl_2653	*Caudovirales*	Ga0222673_1000073	Interface	Oct 2014_I_0.1 µm	3300022821	62,977	0.34	65
cl_2987	*Caudovirales*	Ga0222667_1000147	Interface	Aug 2014_I_0.1 µm	3300022890	40,847	0.42	27
cl_3162	*Caudovirales*	Ga0222676_1000050	Interface	Oct 2014_U3_0.1 µm	3300023240	57,415	0.58	16
cl_3886	*Caudovirales*	Ga0222628_1000116	Interface	Nov 2013_I_0.1 µm	3300022871	39,121	0.33	23
cl_3903	*Caudovirales*	Ga0222628_1000072	Interface	Nov 2013_I_0.1 µm	3300022871	49,428	0.3	36
cl_4460	*Caudovirales*	Ga0208647_1000090	Interface	Nov 2008_I_0.1 µm	3300025362	39,252	0.3	131
	Caudovirales	Ga0208901_1000202	Interface	Nov 2008_I_0.8 µm	3300025380	39,252	0.3	74
cl_4998	*Caudovirales*	Ga0222694_1000023	Interface	Dec 2014_I_0.1 µm	3300023292	46,925	0.36	27
cl_6611	*Caudovirales*	Ga0222673_1000151	Interface	Oct 2014_I_0.1 µm	3300022821	41,192	0.41	21
cl_6749	*Caudovirales*	Ga0222673_1000068	Interface	Oct 2014_I_0.1 µm	3300022821	64,142	0.59	20
cl_8415	*Caudovirales*	Ga0222671_1000142	Interface	Oct 2014_I_3 µm	3300022856	37,783	0.31	22
cl_8461	*Caudovirales*	Ga0208647_1000095	Interface	Nov 2008_I_0.1 µm	3300025362	38,594	0.4	31

cl_8535	*Caudovirales*	Ga0208900_1000079	Interface	Nov 2008_I_3 µm	3300025433	57,510	0.52	47
sg_10466	Unknown	Ga0222694_1000048	Interface	Dec 2014_I_0.1 µm	3300023292	35,253	0.31	54
sg_8715	*Caudovirales*	Ga0222673_1000129	Interface	Oct 2014_I_0.1 µm	3300022821	45,604	0.34	24
cl_116	*Caudovirales*	Ga0222626_1000295	Interface, Lower	Nov 2013_I_3 µm	3300022882	32,335	0.31	29
	Caudovirales	Ga0222657_1000246	Interface, Lower	Jul 2014_I_0.8 µm	3300023241	32,281	0.31	57
	Caudovirales	Ga0222696_1000266	Interface, Lower	Dec 2014_L1_0.8 µm	3300023233	32,200	0.31	41
cl_540	*Caudovirales*	Ga0222637_1000124	Interface, Lower	Nov 2013_L1_0.1 µm	3300023435	42,243	0.34	17
	Caudovirales	Ga0222667_1000132	Interface, Lower	Aug 2014_I_0.1 µm	3300022890	42,243	0.34	43
	Caudovirales	Ga0222697_1000106	Interface, Lower	Dec 2014_L1_0.1 µm	3300022868	42,243	0.34	32
cl_714	*Caudovirales*	Ga0208904_1000360	Interface, Lower	Nov 2008_L2_0.1 µm	3300025669	39,410	0.32	155
	Caudovirales	Ga0222628_1000114	Interface, Lower	Nov 2013_I_0.1 µm	3300022871	39,366	0.32	47
	Caudovirales	Ga0222664_1000244	Interface, Lower	Aug 2014_U3_0.1 µm	3300023296	39,366	0.32	36
	Caudovirales	Ga0222667_1000156	Interface, Lower	Aug 2014_I_0.1 µm	3300022890	39,366	0.32	95
	Caudovirales	Ga0222697_1000123	Interface, Lower	Dec 2014_L1_0.1 µm	3300022868	39,366	0.32	80
	Caudovirales	Ga0222637_1000143	Interface, Lower	Nov 2013_L1_0.1 µm	3300023435	39,308	0.32	33
	Caudovirales	Ga0222694_1000034	Interface, Lower	Dec 2014_I_0.1 µm	3300023292	39,294	0.32	21

cl_738	*Caudovirales*	Ga0208904_1000278	Interface, Lower	Nov 2008_L2_0.1 µm	3300025669	44,769	0.38	124
	Caudovirales	Ga0222637_1000115	Interface, Lower	Nov 2013_L1_0.1 µm	3300023435	43,214	0.38	58
	Caudovirales	Ga0222658_1000155	Interface, Lower	Jul 2014_I_0.1 µm	3300023257	43,214	0.38	47
	Caudovirales	Ga0222667_1000127	Interface, Lower	Aug 2014_I_0.1 µm	3300022890	43,214	0.38	49
	Caudovirales	Ga0222679_1000066	Interface, Lower	Oct 2014_L1_0.1 µm	3300022858	43,214	0.38	40
	Caudovirales	Ga0222697_1000099	Interface, Lower	Dec 2014_L1_0.1 µm	3300022868	43,214	0.38	34
cl_753	*Caudovirales*	Ga0222657_1000100	Interface, Lower	Jul 2014_I_0.8 µm	3300023241	63,847	0.34	26
	Caudovirales	Ga0222658_1000072	Interface, Lower	Jul 2014_I_0.1 µm	3300023257	63,847	0.34	127
	Caudovirales	Ga0222667_1000072	Interface, Lower	Aug 2014_I_0.1 µm	3300022890	63,847	0.34	105
	Caudovirales	Ga0222673_1000070	Interface, Lower	Oct 2014_I_0.1 µm	3300022821	63,847	0.34	76
	Caudovirales	Ga0222694_1000013	Interface, Lower	Dec 2014_I_0.1 µm	3300023292	63,847	0.34	104
	Caudovirales	Ga0222697_1000052	Interface, Lower	Dec 2014_L1_0.1 µm	3300022868	63,847	0.34	196
cl_782	*Caudovirales*	Ga0222658_1000118	Interface, Lower	Jul 2014_I_0.1 µm	3300023257	49,360	0.33	38
	Caudovirales	Ga0208647_1000101	Interface, Lower	Nov 2008_I_0.1 µm	3300025362	37,783	0.31	20
	Caudovirales	Ga0208904_1000394	Interface, Lower	Nov 2008_L2_0.1 µm	3300025669	37,783	0.31	109
	Caudovirales	Ga0222673_1000179	Interface, Lower	Oct 2014_I_0.1 µm	3300022821	37,783	0.31	92

	Caudovirales	Ga0222697_1000133	Interface, Lower	Dec 2014_L1_0.1 µm	3300022868	37,783	0.31	35
cl_1928	*Caudovirales*	Ga0222628_1000130	Interface, Lower	Nov 2013_I_0.1 µm	3300022871	37,186	0.4	15
	Caudovirales	Ga0222637_1000162	Interface, Lower	Nov 2013_L1_0.1 µm	3300023435	37,186	0.4	38
	Caudovirales	Ga0222667_1000169	Interface, Lower	Aug 2014_I_0.1 µm	3300022890	37,186	0.4	38
	Caudovirales	Ga0222679_1000092	Interface, Lower	Oct 2014_L1_0.1 µm	3300022858	37,186	0.4	18
	Caudovirales	Ga0222697_1000138	Interface, Lower	Dec 2014_L1_0.1 µm	3300022868	37,186	0.4	32
cl_2172	*Caudovirales*	Ga0208904_1000242	Interface, Lower	Nov 2008_L2_0.1 µm	3300025669	47,696	0.41	50
	Caudovirales	Ga0222658_1000126	Interface, Lower	Jul 2014_I_0.1 µm	3300023257	47,696	0.41	23
NA	*Caudovirales*	Ga0222626_1000161	Interface, Lower	Nov 2013_I_3 µm	3300022882	47,710	0.39	29
	Caudovirales	Ga0222627_1000111	Interface, Lower	Nov 2013_I_0.8 µm	3300023244	47,710	0.39	118
	Caudovirales	Ga0222636_1000113	Interface, Lower	Nov 2013_L1_0.8 µm	3300022854	47,710	0.39	17
	Caudovirales	Ga0222666_1000189	Interface, Lower	Aug 2014_I_0.8 µm	3300024048	47,710	0.39	65
	Caudovirales	Ga0222667_1000103	Interface, Lower	Aug 2014_I_0.1 µm	3300022890	47,710	0.39	46
	Caudovirales	Ga0222696_1000167	Interface, Lower	Dec 2014_L1_0.8 µm	3300023233	47,710	0.39	52
cl_24	*Caudovirales*	Ga0208769_1000001	Lower	Nov 2008_L1_0.1 µm	3300025697	528,260	0.56	78
	Caudovirales	Ga0222679_1000001	Lower	Oct 2014_L1_0.1 µm	3300022858	528,258	0.56	65

	Caudovirales	Ga0222682_1000001	Lower	Oct 2014_L2_0.1 µm	3300023246	528,256	0.56	30
cl_82	*Caudovirales*	Ga0222635_1000049	Lower	Nov 2013_L1_3 µm	3300023234	65,245	0.51	43
cl_113	*Caudovirales*	Ga0208769_1000003	Lower	Nov 2008_L1_0.1 µm	3300025697	185,273	0.4	22
	Caudovirales	Ga0208771_1000009	Lower	Nov 2008_L3_3 µm	3300025698	185,273	0.4	33
	Caudovirales	Ga0208905_1000005	Lower	Nov 2008_L3_0.8 µm	3300025661	185,273	0.4	28
cl_205	*Caudovirales*	Ga0207996_1000318	Lower	Nov 2008_L2_0.8 µm	3300025586	44,112	0.4	91
	Caudovirales	Ga0208648_1000425	Lower	Nov 2008_L2_3 µm	3300025642	44,112	0.4	21
cl_311	*Caudovirales*	Ga0208902_1000117	Lower	Nov 2008_L1_0.8 µm	3300025628	48,707	0.44	21
	Caudovirales	Ga0208279_1000257	Lower	Nov 2008_L1_3 µm	3300025649	48,534	0.44	355
cl_740	*Caudovirales*	Ga0208771_1000154	Lower	Nov 2008_L3_3 µm	3300025698	43,313	0.36	18
cl_866	*Caudovirales*	Ga0208904_1000201	Lower	Nov 2008_L2_0.1 µm	3300025669	51,617	0.55	37
	Caudovirales	Ga0222640_1000018	Lower	Nov 2013_L2_0.1 µm	3300023297	51,617	0.55	47
	Caudovirales	Ga0222679_1000040	Lower	Oct 2014_L1_0.1 µm	3300022858	51,617	0.55	24
cl_872	*Caudovirales*	Ga0222696_1000206	Lower	Dec 2014_L1_0.8 µm	3300023233	40,169	0.31	44
cl_914	*Caudovirales*	Ga0208904_1000193	Lower	Nov 2008_L2_0.1 µm	3300025669	52,303	0.39	22
	Caudovirales	Ga0222679_1000038	Lower	Oct 2014_L1_0.1 µm	3300022858	52,303	0.39	15

cl_1040	*Caudovirales*	Ga0208769_1000193	Lower	Nov 2008_L1_0.1 µm	3300025697	32,656	0.42	27
cl_1137	*Caudovirales*	Ga0207996_1000364	Lower	Nov 2008_L2_0.8 µm	3300025586	41,358	0.35	34
cl_1152	*Caudovirales*	Ga0222685_1000017	Lower	Oct 2014_L3_0.1 µm	3300022874	36,298	0.36	24
cl_1153	*Caudovirales*	Ga0208769_1000085	Lower	Nov 2008_L1_0.1 µm	3300025697	44,334	0.36	17
cl_1270	*Caudovirales*	Ga0222637_1000055	Lower	Nov 2013_L1_0.1 µm	3300023435	59,445	0.31	24
cl_1274	*Caudovirales*	Ga0208769_1000038	Lower	Nov 2008_L1_0.1 µm	3300025697	68,986	0.51	46
cl_1424	*Caudovirales*	Ga0208905_1000064	Lower	Nov 2008_L3_0.8 µm	3300025661	51,499	0.34	33
cl_1429	*Caudovirales*	Ga0208771_1000215	Lower	Nov 2008_L3_3 µm	3300025698	36,811	0.34	20
	Caudovirales	Ga0208905_1000127	Lower	Nov 2008_L3_0.8 µm	3300025661	36,811	0.34	24
cl_1640	*Caudovirales*	Ga0208904_1000412	Lower	Nov 2008_L2_0.1 µm	3300025669	36,601	0.4	23
cl_1869	*Caudovirales*	Ga0222637_1000130	Lower	Nov 2013_L1_0.1 µm	3300023435	41,872	0.39	19
	Caudovirales	Ga0208904_1000324	Lower	Nov 2008_L2_0.1 µm	3300025669	41,796	0.39	36
cl_1870	*Caudovirales*	Ga0222637_1000047	Lower	Nov 2013_L1_0.1 µm	3300023435	65,207	0.39	16
cl_1875	*Caudovirales*	Ga0208769_1000105	Lower	Nov 2008_L1_0.1 µm	3300025697	42,085	0.41	23
cl_1882	*Caudovirales*	Ga0208769_1000053	Lower	Nov 2008_L1_0.1 µm	3300025697	58,972	0.39	38
cl_2151	*Caudovirales*	Ga0208771_1001592	Lower	Nov 2008_L3_3 µm	3300025698	11,801	0.36	20

	Caudovirales	Ga0208905_1001478	Lower	Nov 2008_L3_0.8 µm	3300025661	11,801	0.36	19
	Caudovirales	Ga0307928_10003616	Lower	Nov 2013_L3_0.1 µm	3300031227	11,801	0.36	31
cl_2535	*Caudovirales*	Ga0222679_1000074	Lower	Oct 2014_L1_0.1 µm	3300022858	41,354	0.39	41
cl_2539	*Caudovirales*	Ga0222697_1000080	Lower	Dec 2014_L1_0.1 µm	3300022868	48,241	0.32	32
cl_2543	Unknown	Ga0207997_1000146	Lower	Nov 2008_L3_0.1 µm	3300025736	33,354	0.34	86
cl_2955	*Caudovirales*	Ga0208648_1000482	Lower	Nov 2008_L2_3 µm	3300025642	40,949	0.42	60
	Caudovirales	Ga0207996_1000375	Lower	Nov 2008_L2_0.8 µm	3300025586	40,946	0.42	66
cl_4777	*Caudovirales*	Ga0307928_10000216	Lower	Nov 2013_L3_0.1 µm	3300031227	40,506	0.33	20
cl_6083	*Caudovirales*	Ga0208904_1000262	Lower	Nov 2008_L2_0.1 µm	3300025669	46,043	0.56	23
cl_6100	*Caudovirales*	Ga0208904_1000286	Lower	Nov 2008_L2_0.1 µm	3300025669	43,985	0.42	54
cl_6184	*Caudovirales*	Ga0307928_10000238	Lower	Nov 2013_L3_0.1 µm	3300031227	39,571	0.49	40
cl_6251	*Caudovirales*	Ga0307928_10000150	Lower	Nov 2013_L3_0.1 µm	3300031227	44,919	0.31	32
cl_6647	*Caudovirales*	Ga0307928_10000232	Lower	Nov 2013_L3_0.1 µm	3300031227	39,766	0.36	23
cl_8417	*Caudovirales*	Ga0222634_1000217	Lower	Nov 2013_U3_0.1 µm	3300023235	33,882	0.32	12
cl_8655	*Caudovirales*	Ga0208902_1000132	Lower	Nov 2008_L1_0.8 µm	3300025628	46,752	0.4	23
cl_8662	*Caudovirales*	Ga0208648_1000620	Lower	Nov 2008_L2_3 µm	3300025642	34,039	0.48	20

cl_9532	*Caudovirales*	Ga0307928_10000040	Lower	Nov 2013_L3_0.1 µm	3300031227	67,356	0.54	27
cl_10290	*Caudovirales*	Ga0222679_1000185	Lower	Oct 2014_L1_0.1 µm	3300022858	25,884	0.34	16
	Caudovirales	Ga0222637_1000264	Lower	Nov 2013_L1_0.1 µm	3300023435	25,839	0.34	20
sg_10366	Unknown	Ga0222703_1000045	Lower	Dec 2014_L3_0.1 µm	3300023256	33,260	0.34	16
sg_11172	*Caudovirales*	Ga0307928_10003248	Lower	Nov 2013_L3_0.1 µm	3300031227	12,412	0.35	23
sg_11648	*Caudovirales*	Ga0307928_10000234	Lower	Nov 2013_L3_0.1 µm	3300031227	39,713	0.56	16
sg_2	*Caudovirales*	Ga0222636_1000107	Lower	Nov 2013_L1_0.8 µm	3300022854	48,758	0.6	22
sg_550	*Caudovirales*	Ga0208904_1000255	Lower	Nov 2008_L2_0.1 µm	3300025669	46,474	0.39	47
sg_576	*Caudovirales*	Ga0208904_1000272	Lower	Nov 2008_L2_0.1 µm	3300025669	45,040	0.39	23
sg_637	*Caudovirales*	Ga0208904_1000246	Lower	Nov 2008_L2_0.1 µm	3300025669	47,285	0.31	32
sg_9367	*Caudovirales*	Ga0222679_1000111	Lower	Oct 2014_L1_0.1 µm	3300022858	34,677	0.32	25
NA	*Caudovirales*	Ga0207997_1000112	Lower	Nov 2008_L3_0.1 µm	3300025736	37,866	0.37	17
	Caudovirales	Ga0307928_10000272	Lower	Nov 2013_L3_0.1 µm	3300031227	37,866	0.37	50
NA	*Caudovirales*	Ga0222640_1000087	Lower	Nov 2013_L2_0.1 µm	3300023297	31,376	0.31	35
NA	*Caudovirales*	Ga0207997_1000616	Lower	Nov 2008_L3_0.1 µm	3300025736	18,059	0.35	29
NA	*Caudovirales*	Ga0207997_1001313	Lower	Nov 2008_L3_0.1 µm	3300025736	12,882	0.41	17

	Caudovirales	Ga0307928_10003015	Lower	Nov 2013_L3_0.1 µm	3300031227	12,882	0.41	48
NA	*Caudovirales*	Ga0307928_10003459	Lower	Nov 2013_L3_0.1 µm	3300031227	12,048	0.43	15
NA	*Caudovirales*	Ga0222697_1000929	Lower	Dec 2014_L1_0.1 µm	3300022868	11,901	0.35	14
NA	*Caudovirales*	Ga0207997_1001542	Lower	Nov 2008_L3_0.1 µm	3300025736	11,831	0.44	15
Cyanophage contigs[E]								
Cyanophage	Unknown	Ga0078900_115654	Upper	Dec 2006_U2_0.1 µm	3300016486	548,945	0.28	9
cl_6580	Unknown	Ga0222688_1000642	Upper	Dec 2014_U1_0.1 µm	3300023293	7,681	0.27	9
	Unknown	Ga0302071_100018	Upper	Dec 2006_U1_0.1 µm	3300028228	51,665	0.29	9
cl_6727	Unknown	Ga0222688_1000631	Upper	Dec 2014_U1_0.1 µm	3300023293	7,780	0.27	9
	Unknown	Ga0302071_100023	Upper	Dec 2006_U1_0.1 µm	3300028228	43,753	0.28	10
cl_9495	Unknown	Ga0302066_100481	Upper	Dec 2006_U2_0.1 µm	3300028222	6,313	0.29	4
cl_9892	Unknown	Ga0302071_100165	Upper	Dec 2006_U1_0.1 µm	3300028228	17,793	0.26	9
sg_14929	Unknown	Ga0302071_100266	Upper	Dec 2006_U1_0.1 µm	3300028228	13,303	0.3	11
sg_14949	Unknown	Ga0302071_100224	Upper	Dec 2006_U1_0.1 µm	3300028228	14,834	0.27	10
sg_14969	Unknown	Ga0302071_100180	Upper	Dec 2006_U1_0.1 µm	3300028228	16,925	0.28	9

sg_14971	Unknown	Ga0302071_100029	Upper	Dec 2006_U1_0.1 µm	3300028228	41,278	0.28	14
sg_15003	Unknown	Ga0302071_100139	Upper	Dec 2006_U1_0.1 µm	3300028228	19,294	0.29	10
Potential *Chlorobium* virus contigs[F]								
cl_1024	Unknown	Ga0222689_1000957	Upper	Dec 2014_U3_3 µm	3300023231	6,117	0.48	27
	Unknown	Ga0222690_1000793	Upper	Dec 2014_U3_0.8 µm	3300023227	6,492	0.47	50
	Unknown	Ga0302060_10025	Interface	Dec 2006_I_0.8 µm	3300028201	11,188	0.48	34
	Unknown	Ga0302061_10032	Interface	Dec 2006_I_3 µm	3300028203	8,085	0.49	13
	Unknown	Ga0302067_10039	Interface	Dec 2006_I_0.1 µm	3300028204	6,665	0.45	24
	Unknown	Ga0208900_1004295	Interface	Nov 2008_I_3 µm	3300025433	5,426	0.48	613
	Unknown	Ga0222656_1001806	Interface	Jul 2014_I_3 µm	3300022834	5,624	0.48	29
	Unknown	Ga0222665_1003383	Interface	Aug 2014_I_3 µm	3300022864	5,317	0.47	24
	Unknown	Ga0208904_1006197	Lower	Nov 2008_L2_0.1 µm	3300025669	6,239	0.48	37
	Unknown	Ga0208905_1004525	Lower	Nov 2008_L3_0.8 µm	3300025661	6,310	0.48	46
	Unknown	Ga0222638_1002074	Lower	Nov 2013_L2_3 µm	3300023298	5,603	0.48	28
	Unknown	Ga0222695_1002624	Lower	Dec 2014_L1_3 µm	3300023253	5,952	0.48	30

	Unknown	Ga0222696_1002166	Lower	Dec 2014_L1_0.8 µm	3300023233	5,599	0.48	37
	Unknown	Ga0222699_1002408	Lower	Dec 2014_L1_0.8 µm	3300022846	5,369	0.47	40
cl_1024 matches	Unknown	Ga0222693_103307	Interface	Dec 2014_I_0.8 µm	3300022826	2,461	0.48	204
	Unknown	Ga0222693_105389	Interface	Dec 2014_I_0.8 µm	3300022826	1,845	0.43	247
	Unknown	Ga0222693_109555	Interface	Dec 2014_I_0.8 µm	3300022826	1,307	0.42	192
	Unknown	Ga0222693_111585	Interface	Dec 2014_I_0.8 µm	3300022826	1,152	0.5	162
	Unknown	Ga0222693_113295	Interface	Dec 2014_I_0.8 µm	3300022826	1,048	0.52	162
	Unknown	Ga0222671_1015434	Interface	Oct 2014_I_3 µm	3300022856	1,622	0.43	9
	Unknown	Ga0222628_1006547	Interface	Nov 2013_I_0.1 µm	3300022871	2,995	0.45	242
	Unknown	Ga0222628_1009068	Interface	Nov 2013_I_0.1 µm	3300022871	2,400	0.44	244
	Unknown	Ga0222628_1029116	Interface	Nov 2013_I_0.1 µm	3300022871	1,027	0.5	153
	Unknown	Ga0222626_1005760	Interface	Nov 2013_I_3 µm	3300022882	3,274	0.46	106
	Unknown	Ga0222626_1017698	Interface	Nov 2013_I_3 µm	3300022882	1,399	0.51	83
	Unknown	Ga0222667_1013453	Interface	Aug 2014_I_0.1 µm	3300022890	1,848	0.45	17
	Unknown	Ga0222667_1025494	Interface	Aug 2014_I_0.1 µm	3300022890	1,230	0.47	7
	Unknown	Ga0222657_1006670	Interface	Jun 2014_I_0.8 µm	3300023241	2,965	0.45	21

Unknown	Ga0222627_1005353	Interface	Nov 2013_I_0.8 µm	3300023244	3,111	0.46	197
Unknown	Ga0222658_1008932	Interface	Jun 2014_I_0.1 µm	3300023257	3,093	0.46	18
Unknown	Ga0222658_1012832	Interface	Jun 2014_I_0.1 µm	3300023257	2,415	0.43	18
Unknown	Ga0222694_1001922	Interface	Dec 2014_I_0.1 µm	3300023292	3,576	0.46	218
Unknown	Ga0222694_1008117	Interface	Dec 2014_I_0.1 µm	3300023292	1,568	0.51	203
Unknown	Ga0222666_1006542	Interface	Aug 2014_I_0.8 µm	3300024048	3,036	0.45	16
Unknown	Ga0208647_1001847	Interface	Nov 2008_I_0.1 µm	3300025362	3,923	0.46	412
Unknown	Ga0208647_1011846	Interface	Nov 2008_I_0.1 µm	3300025362	1,173	0.5	367
Unknown	Ga0208901_1003063	Interface	Nov 2008_I_0.8 µm	3300025380	3,061	0.45	586
Unknown	Ga0208901_1005235	Interface	Nov 2008_I_0.8 µm	3300025380	2,150	0.51	420
Unknown	Ga0302067_10108	Interface	Dec 2006_I_0.1 µm	3300028204	3,094	0.45	17
Unknown	Ga0307929_1013488	Interface	Dec 2014_I_3 µm	3300031697	3,034	0.45	561
Unknown	Ga0307929_1022100	Interface	Dec 2014_I_3 µm	3300031697	2,150	0.51	512
Unknown	Ga0222641_1001513	Lower	Nov 2013_L3_3 µm	3300022828	3,169	0.46	15
Unknown	Ga0222641_1008646	Lower	Nov 2013_L3_3 µm	3300022828	1,075	0.51	19
Unknown	Ga0222636_1027756	Lower	Nov 2013_L1_0.8 µm	3300022854	1,148	0.47	8

Unknown	Ga0222677_1004353	Lower	Oct 2014_L1_3 µm	3300022855	3,025	0.45	16
Unknown	Ga0222677_1008770	Lower	Oct 2014_L1_3 µm	3300022855	1,884	0.51	13
Unknown	Ga0222679_1019904	Lower	Oct 2014_L1_0.1 µm	3300022858	1,315	0.46	8
Unknown	Ga0222679_1021467	Lower	Oct 2014_L1_0.1 µm	3300022858	1,249	0.42	12
Unknown	Ga0222698_1004621	Lower	Dec 2014_L2_3 µm	3300022860	3,468	0.46	27
Unknown	Ga0222698_1010393	Lower	Dec 2014_L2_3 µm	3300022860	2,138	0.5	27
Unknown	Ga0222697_1005317	Lower	Dec 2014_L1_0.1 µm	3300022868	3,624	0.46	22
Unknown	Ga0222685_1005680	Lower	Oct 2014_L3_0.1 µm	3300022874	2,835	0.45	17
Unknown	Ga0222701_1002792	Lower	Dec 2014_L3_3 µm	3300022884	4,249	0.47	37
Unknown	Ga0222701_1007002	Lower	Dec 2014_L3_3 µm	3300022884	2,347	0.43	42
Unknown	Ga0222642_1004415	Lower	Nov 2013_L3_0.8 µm	3300022887	3,428	0.46	19
Unknown	Ga0222700_1011970	Lower	Dec 2014_L2_0.1 µm	3300023061	1,563	0.43	14
Unknown	Ga0222700_1013246	Lower	Dec 2014_L2_0.1 µm	3300023061	1,479	0.48	9
Unknown	Ga0222700_1014196	Lower	Dec 2014_L2_0.1 µm	3300023061	1,422	0.35	18
Unknown	Ga0222700_1022360	Lower	Dec 2014_L2_0.1 µm	3300023061	1,095	0.41	10
Unknown	Ga0222635_1003665	Lower	Nov 2013_L1_3 µm	3300023234	3,370	0.46	22

Unknown	Ga0222635_1006398	Lower	Nov 2013_L1_3 µm	3300023234	2,336	0.43	19
Unknown	Ga0222635_1017146	Lower	Nov 2013_L1_3 µm	3300023234	1,209	0.51	22
Unknown	Ga0222682_1006090	Lower	Oct 2014_L2_0.1 µm	3300023246	2,530	0.45	9
Unknown	Ga0222678_1026606	Lower	Oct 2014_L1_0.8 µm	3300023249	1,023	0.4	4
Unknown	Ga0222683_1002296	Lower	Oct 2014_L3_3 µm	3300023251	4,576	0.47	36
Unknown	Ga0222695_1008753	Lower	Dec 2014_L1_3 µm	3300023253	2,285	0.43	28
Unknown	Ga0222703_1004199	Lower	Dec 2014_L3_0.1 µm	3300023256	3,156	0.46	16
Unknown	Ga0222639_1009454	Lower	Nov 2013_L2_0.8 µm	3300023262	3,054	0.45	13
Unknown	Ga0222684_1001894	Lower	Oct 2014_L3_0.8 µm	3300023295	5,033	0.48	53
Unknown	Ga0222640_1009671	Lower	Nov 2013_L2_0.1 µm	3300023297	2,458	0.44	10
Unknown	Ga0222702_1004458	Lower	Dec 2014_L3_0.8 µm	3300023299	3,853	0.47	38
Unknown	Ga0222702_1035347	Lower	Dec 2014_L3_0.8 µm	3300023299	1,123	0.5	22
Unknown	Ga0222680_1002636	Lower	Oct 2014_L2_3 µm	3300023434	4,220	0.47	37
Unknown	Ga0222637_1013402	Lower	Nov 2013_L1_0.1 µm	3300023435	1,902	0.43	12
Unknown	Ga0207996_1009992	Lower	Nov 2008_L2_0.8 µm	3300025586	3,789	0.47	30
Unknown	Ga0208902_1009654	Lower	Nov 2008_L1_0.8 µm	3300025628	3,754	0.46	18

	Unknown	Ga0208648_1008385	Lower	Nov 2008_L2_3 µm	3300025642	5,049	0.48	47
	Unknown	Ga0208648_1067131	Lower	Nov 2008_L2_3 µm	3300025642	1,142	0.5	44
	Unknown	Ga0208279_1006523	Lower	Nov 2008_L1_3 µm	3300025649	4,535	0.47	44
	Unknown	Ga0208769_1009442	Lower	Nov 2008_L1_0.1 µm	3300025697	3,589	0.46	14
	Unknown	Ga0208771_1005090	Lower	Nov 2008_L3_3 µm	3300025698	5,506	0.48	86
	Unknown	Ga0207997_1013304	Lower	Nov 2008_L3_0.1 µm	3300025736	3,540	0.46	34
	Unknown	Ga0307928_10010556	Lower	Nov 2013_L3_0.1 µm	3300031227	6,634	0.47	79
cl_248	Unknown	Ga0222686_1001026	Upper	Dec 2014_U2_3 µm	3300023501	6,903	0.51	150
	Unknown	Ga0222689_1000893	Upper	Dec 2014_U3_3 µm	3300023231	6,437	0.51	926
	Unknown	Ga0222689_1001025	Upper	Dec 2014_U3_3 µm	3300023231	5,833	0.5	954
	Unknown	Ga0222691_1002182	Upper	Dec 2014_U3_0.1 µm	3300022851	5,045	0.52	38
	Unknown	Ga0302060_10018	Interface	Dec 2006_I_0.8 µm	3300028201	17,393	0.5	79
	Unknown	Ga0302061_10026	Interface	Dec 2006_I_3 µm	3300028203	10,117	0.52	28
	Unknown	Ga0302067_10021	Interface	Dec 2006_I_0.1 µm	3300028204	12,817	0.51	66
	Unknown	Ga0222626_1002761	Interface	Nov 2013_I_3 µm	3300022882	6,100	0.51	383
	Unknown	Ga0222627_1001894	Interface	Nov 2013_I_0.8 µm	3300023244	6,106	0.51	365

Unknown	Ga0222628_1002177	Interface	Nov 2013_I_0.1 µm	3300022871	6,106	0.51	796
Unknown	Ga0302056_100137	Lower	Dec 2006_L2_0.8 µm	3300028227	7,336	0.51	6
Unknown	Ga0302055_100061	Lower	Dec 2006_L3_0.8 µm	3300028226	8,566	0.51	10
Unknown	Ga0302070_100065	Lower	Dec 2006_L3_0.1 µm	3300028296	7,664	0.49	6
Unknown	Ga0302070_100157	Lower	Dec 2006_L3_0.1 µm	3300028296	5,159	0.51	7
Unknown	Ga0208279_1004110	Lower	Nov 2008_L1_3 µm	3300025649	6,103	0.51	515
Unknown	Ga0208902_1002585	Lower	Nov 2008_L1_0.8 µm	3300025628	8,161	0.51	136
Unknown	Ga0208769_1002379	Lower	Nov 2008_L1_0.1 µm	3300025697	7,992	0.51	143
Unknown	Ga0207996_1003791	Lower	Nov 2008_L2_0.8 µm	3300025586	7,966	0.51	189
Unknown	Ga0208904_1004243	Lower	Nov 2008_L2_0.1 µm	3300025669	8,059	0.51	284
Unknown	Ga0222635_1001297	Lower	Nov 2013_L1_3 µm	3300023234	7,103	0.51	131
Unknown	Ga0222636_1002821	Lower	Nov 2013_L1_0.8 µm	3300022854	5,518	0.51	104
Unknown	Ga0222637_1002828	Lower	Nov 2013_L1_0.1 µm	3300023435	5,215	0.51	73
Unknown	Ga0222639_1002111	Lower	Nov 2013_L2_0.8 µm	3300023262	8,030	0.51	157
Unknown	Ga0222640_1001468	Lower	Nov 2013_L2_0.1 µm	3300023297	6,934	0.51	138
Unknown	Ga0222641_1000777	Lower	Nov 2013_L3_3 µm	3300022828	5,218	0.52	91

	Unknown	Ga0222679_1001284	Lower	Oct 2014_L1_0.1 µm	3300022858	7,692	0.51	68
	Unknown	Ga0222682_1001695	Lower	Oct 2014_L2_0.1 µm	3300023246	5,038	0.52	108
	Unknown	Ga0222695_1002611	Lower	Dec 2014_L1_3 µm	3300023253	5,972	0.5	264
	Unknown	Ga0222696_1001254	Lower	Dec 2014_L1_0.8 µm	3300023233	8,225	0.5	132
	Unknown	Ga0222698_1001226	Lower	Dec 2014_L2_3 µm	3300022860	7,682	0.5	354
	Unknown	Ga0222698_1002050	Lower	Dec 2014_L2_3 µm	3300022860	5,591	0.51	399
	Unknown	Ga0222699_1001059	Lower	Dec 2014_L2_0.8 µm	3300022846	8,592	0.51	279
	Unknown	Ga0222700_1001281	Lower	Dec 2014_L2_0.1 µm	3300023061	5,063	0.51	82
	Unknown	Ga0222701_1001644	Lower	Dec 2014_L3_3 µm	3300022884	6,103	0.51	265
	Unknown	Ga0222702_1001705	Lower	Dec 2014_L3_0.8 µm	3300023299	6,313	0.51	233
cl_400	Unknown	Ga0208414_1003043	Upper	Nov 2008_U2_0.1 µm	3300025603	10,451	0.34	20
	Unknown	Ga0222689_1000515	Upper	Dec 2014_U2_3 µm	3300023231	10,005	0.33	31
	Unknown	Ga0222691_1000883	Upper	Dec 2014_U2_0.1 µm	3300022851	9,331	0.33	23
	Unknown	Ga0208900_1002555	Interface	Nov 2008_I_3 µm	3300025433	8,218	0.33	308
	Unknown	Ga0208901_1000990	Interface	Nov 2008_I_0.1 µm	3300025380	8,545	0.33	456
	Unknown	Ga0208647_1000742	Interface	Nov 2008_I_0.1 µm	3300025362	7,900	0.33	955

Unknown	Ga0222656_1001064	Interface	Jul 2014_I_3 µm	3300022834	7,890	0.33	82
Unknown	Ga0222658_1003935	Interface	Jul 2014_I_0.1 µm	3300023257	5,310	0.34	481
Unknown	Ga0222666_1002834	Interface	Aug 2014_I_0.1 µm	3300024048	5,311	0.34	279
Unknown	Ga0222673_1001116	Interface	Oct 2014_I_0.1 µm	3300022821	9,687	0.33	30
Unknown	Ga0208279_1002046	Lower	Nov 2008_L1_3 µm	3300025649	10,106	0.33	128
Unknown	Ga0208902_1002371	Lower	Nov 2008_L1_0.1 µm	3300025628	8,544	0.33	257
Unknown	Ga0208769_1002438	Lower	Nov 2008_L1_0.1 µm	3300025697	7,877	0.33	341
Unknown	Ga0207996_1003599	Lower	Nov 2008_L2_0.1 µm	3300025586	8,246	0.33	540
Unknown	Ga0208904_1004387	Lower	Nov 2008_L2_0.1 µm	3300025669	7,890	0.33	1034
Unknown	Ga0208771_1002615	Lower	Nov 2008_L3_3 µm	3300025698	8,545	0.33	312
Unknown	Ga0208905_1003078	Lower	Nov 2008_L3_0.1 µm	3300025661	7,889	0.33	337
Unknown	Ga0207997_1003371	Lower	Nov 2008_L3_0.1 µm	3300025736	7,890	0.33	400
Unknown	Ga0222635_1001119	Lower	Nov 2013_L1_3 µm	3300023234	7,890	0.33	112
Unknown	Ga0222638_1001230	Lower	Nov 2013_L2_3 µm	3300023298	7,890	0.33	82
Unknown	Ga0222677_1001056	Lower	Oct 2014_L1_3 µm	3300022855	7,889	0.33	117
Unknown	Ga0222681_1001368	Lower	Oct 2014_L2_0.1 µm	3300022838	7,890	0.33	59

	Unknown	Ga0222682_1001525	Lower	Oct 2014_L2_0.1 µm	3300023246	5,311	0.34	146
	Unknown	Ga0222685_1000765	Lower	Oct 2014_L3_0.1 µm	3300022874	7,889	0.33	167
	Unknown	Ga0222700_1000514	Lower	Dec 2014_L2_0.1 µm	3300023061	7,890	0.33	148
	Unknown	Ga0222701_1002009	Lower	Dec 2014_L3_3 µm	3300022884	5,311	0.34	132
sg_14554	Unknown	Ga0302067_10019	Interface	Dec 2006_I_0.1 µm	3300028204	14,104	0.51	146
sg_14554 matches	Unknown	Ga0302064_100390	Upper	Dec 2006_U2_0.8 µm	3300028221	5,671	0.5	8
	Unknown	Ga0302071_101098	Upper	Dec 2006_U2_0.1 µm	3300028228	4,692	0.52	9
	Unknown	Ga0222693_108773	Interface	Dec 2014_I_0.8 µm	3300022826	1,380	0.53	461
	Unknown	Ga0222628_1020343	Interface	Nov 2013_I_0.1 µm	3300022871	1,348	0.5	359
	Unknown	Ga0222658_1028294	Interface	Jun 2014_I_0.1 µm	3300023257	1,396	0.52	73
	Unknown	Ga0302067_10094	Interface	Dec 2006_I_0.1 µm	3300028204	3,305	0.52	89
	Unknown	Ga0307929_1025180	Interface	Dec 2014_I_3 µm	3300031697	1,969	0.51	931
	Unknown	Ga0307929_1026701	Interface	Dec 2014_I_3 µm	3300031697	1,893	0.51	466
	Unknown	Ga0307929_1028327	Interface	Dec 2014_I_3 µm	3300031697	1,825	0.52	1794
	Unknown	Ga0307929_1033417	Interface	Dec 2014_I_3 µm	3300031697	1,633	0.52	650
	Unknown	Ga0307929_1033418	Interface	Dec 2014_I_3 µm	3300031697	1,633	0.51	630

Unknown	Ga0307929_1034033	Interface	Dec 2014_I_3 µm	3300031697	1,614	0.49	1200
Unknown	Ga0307929_1039738	Interface	Dec 2014_I_3 µm	3300031697	1,458	0.54	341
Unknown	Ga0307929_1041004	Interface	Dec 2014_I_3 µm	3300031697	1,427	0.52	247
Unknown	Ga0307929_1043591	Interface	Dec 2014_I_3 µm	3300031697	1,371	0.51	1328
Unknown	Ga0222681_1011995	Lower	Oct 2014_L2_0.8 µm	3300022838	1,909	0.49	42
Unknown	Ga0222699_1019844	Lower	Dec 2014_L2_0.8 µm	3300022846	1,365	0.52	203
Unknown	Ga0222698_1019532	Lower	Dec 2014_L2_3 µm	3300022860	1,412	0.51	361
Unknown	Ga0222697_1012543	Lower	Dec 2014_L1_0.1 µm	3300022868	1,997	0.51	126
Unknown	Ga0222697_1013453	Lower	Dec 2014_L1_0.1 µm	3300022868	1,902	0.49	103
Unknown	Ga0222697_1017175	Lower	Dec 2014_L1_0.1 µm	3300022868	1,609	0.47	69
Unknown	Ga0222701_1013523	Lower	Dec 2014_L3_3 µm	3300022884	1,573	0.51	135
Unknown	Ga0222700_1013614	Lower	Dec 2014_L2_0.1 µm	3300023061	1,457	0.52	94
Unknown	Ga0222695_1013787	Lower	Dec 2014_L1_3 µm	3300023253	1,611	0.52	113
Unknown	Ga0222680_1014745	Lower	Oct 2014_L2_3 µm	3300023434	1,606	0.52	122
Unknown	Ga0208905_1016630	Lower	Nov 2008_L3_0.8 µm	3300025661	2,843	0.51	138
Unknown	Ga0208905_1024838	Lower	Nov 2008_L3_0.8 µm	3300025661	2,211	0.5	208

	Unknown	Ga0207997_1031231	Lower	Nov 2008_L3_0.1 µm	3300025736	2,078	0.51	127
	Unknown	Ga0302057_100737	Lower	Dec 2006_L2_3 µm	3300028199	1,897	0.51	5
sg_10581	Unknown	Ga0222684_1001894	Lower	Oct 2014_L3_0.8 µm	3300023295	5,033	0.48	53
sg_14551	Unknown	Ga0302067_10044	Interface	Dec 2006_I_0.1 µm	3300028204	6,155	0.48	127
sg_14796	Unknown	Ga0302068_100085	Lower	Dec 2006_L1_0.1 µm	3300028219	7,610	0.49	5
sg_14959	Unknown	Ga0302071_100505	Upper	Dec 2006_U1_0.1 µm	3300028228	8,621	0.5	7
cl_9176	Unknown	Ga0306906_1001873	NA	Rauer 13 Lake	3300028374	6,481	0.5	35
sg_1370	Unknown	Ga0306906_1000248	NA	Rauer 13 Lake	3300028374	23,985	0.46	27
sg_1648	Unknown	Ga0307254_100878	NA	Deep Lake 24 m 3 µm	3300028435	8,628	0.52	72
sg_1649	Unknown	Ga0307254_100745	NA	Deep Lake 24 m 3 µm	3300028435	10,206	0.53	72
sg_1677	Unknown	Ga0307253_100979	NA	Deep Lake 24 m 0.8 µm	3300028451	7,606	0.53	74
'Huge' phage genome contigs[H]								
cl_24	*Caudovirales*	Ga0208769_1000001	Lower	Nov 2008_L1_0.1 µm	3300025697	528,260	0.56	78
	Unknown	Ga0208904_1000003	Lower	Nov 2008_L2_0.1 µm	3300025669	447,854	0.56	174
	Unknown	Ga0208771_1000001	Lower	Nov 2008_L3_3 µm	3300025698	528,138	0.56	22

	Unknown	Ga0222637_1000003*	Lower	Nov 2013_L1_0.1 µm	3300023435	323,923	0.56	98
	Unknown	Ga0222637_1000005*			3300023435	204,206	0.56	114
	Unknown	Ga0222640_1000001	Lower	Nov 2013_L2_0.1 µm	3300023297	528,282	0.56	44
	Caudovirales	Ga0222679_1000001	Lower	Oct 2014_L1_0.1 µm	3300022858	528,258	0.56	65
	Caudovirales	Ga0222682_1000001	Lower	Oct 2014_L2_0.1 µm	3300023246	528,256	0.56	30
	Unknown	Ga0222700_1000001	Lower	Dec 2014_L2_0.1 µm	3300023061	521,772	0.56	23

Table H2. List of abundant viral clusters. The most abundant viral clusters in Ace Lake were identified from the Antarctic virus catalogue (Chapter 3 section 3.2.6.6). [A] The Ace Lake zone to which most of the contigs of an abundant viral cluster belonged are shown in bold letters in the third column. Ace Lake depths: Upper, upper oxic zone; Interface, oxycline; Lower, lower anoxic zone.

Viral clusters	Probable host	Ace Lake zone (number of contigs)[A]	Total contigs	Contig length range (median length)	Mean GC content	Read depth range (median value)
cl_5	Bacteria	**Upper** (133); Interface (2); Lower (1)	136	5–89 kb (9 kb)	0.34	10–723 (63)
cl_7	Eukarya	**Upper** (111)	111	5–72 kb (11 kb)	0.4	4–647 (107)
cl_9	Eukarya	**Upper** (101); Interface (2); Lower (8)	111	5–42 kb (12 kb)	0.4	4–962 (48)
cl_11	Bacteria	**Upper** (112); Interface (3); Lower (2)	117	5–61 kb (13 kb)	0.56	9–1286 (70)
cl_20	Eukarya	**Upper** (66); Interface (3); Lower (14)	83	5–62 kb (13 kb)	0.4	3–3675 (159)
cl_32	Eukarya	**Upper** (70)	70	6–52 kb (13 kb)	0.41	7–804 (90)
cl_35	Eukarya	**Upper** (56); Lower (3)	59	5–49 kb (6 kb)	0.43	5–1713 (185)
cl_37	Unknown	**Upper** (68)	68	5–57 kb (19 kb)	0.43	9–556 (66)
cl_54	Unknown	**Upper** (42); Interface (1); Lower (18)	61	5–31 kb (8 kb)	0.52	11–1766 (63)
cl_66	Eukarya	**Upper** (54)	54	7–34 kb (12 kb)	0.46	10–705 (137)

cl_89	Unknown	**Upper** (55); Lower (1)	56	5–51 kb (22 kb)	0.43	8–1301 (108)
cl_159	Bacteria	**Upper** (41)	41	5–31 kb (7 kb)	0.3	11–2922 (58)
cl_191	Unknown	**Upper** (39)	39	7–45 kb (15 kb)	0.42	11–1333 (157)
cl_248	Unknown	Upper (4); Interface (6); **Lower** (25)	35	5–17 kb (7 kb)	0.51	6–954 (138)
cl_295	Bacteria	**Upper** (28); Interface (1); Lower (2)	31	5–26 kb (6 kb)	0.34	9–1660 (147)
cl_400	Unknown	Upper (3); Interface (7); **Lower** (16)	26	5–10 kb (8 kb)	0.33	20–1034 (158)
cl_463	Bacteria	**Upper** (13); Interface (3); Lower (6)	22	5–45 kb (24 kb)	0.62	11–6412 (127)

Table H3. List of *Chlorobium* spacer and repeat sequences. The spacer and repeat sequences were obtained from (i) 18 Ace Lake metagenomes — mostly from the anoxic zone including Interface (4), Lower 1 (4), Lower 2 (6), and Lower 3 (3) and one from Upper 3 zone; (ii) three Ellis Fjord metagenomes from 60 m depth; and (iii) three Taynaya Bay metagenomes from 5 m (2) and 11 m (1) depths. The methods used for identification of CRISPR spacer arrays in Ace Lake *Chlorobium* and Ellis Fjord and Taynaya Bay *Chlorobium* are discussed in Chapter 3 section 3.2.6.1 and Chapter 5 section 5.2.5, respectively.

Sequence name	Sequence	Sequence length (bp)
CRISPR spacers		
Spc1	TTGCTTCTATCATGATTTGATTCCTCCTATAAG	33
Spc2	CAGGAAAGATGCGTATGCGTGGCGGAAAGGCT	32
Spc3	TCAGTGCTGGGGTAAAGGCGACGACGGCCGGATA	34
Spc4	TTCTATTAGATCAACTGGAAATGGAGCAGGGTG	33
Spc5	CAATGAATTTCACCAACTCAAATCTGGCATTAA	33
Spc6	TCATGCGCCGCCTGCTCCGCGAGCTGGCAACCA	33
Spc7	GCGATAAAGACCGCGTAGCACAGGAAACTGAGG	33
Spc8	TACAACCTCATAGCTTTTGTAGATTTCTTGCAA	33
Spc9	ACCGCCCCCCGCCCGCATAAGGTCATCAGCCTG	33
Spc10	TGACACAGGGGTTTTGATCGACAAAGTTGTGTG	33
Spc11	TTCTGAGAAGTACTGGATCAGGGTTGACTCTTG	33
Spc12	GGCTAGCCTTAGTGGCCACAAAGACTGGAACCA	33
Spc13	TCACAGTTGACGATCCCTGGTCTGATGCTATGA	33
Spc14	CTTGGGGTGTATCAGGCGTCAGGGTTGACAGATG	34
Spc15	TAGCTTTGCTGTAATATGGTCACCTTATCATCTA	34
Spc16	GGCAGGGATAACAGAGCTGCGCAGTCAAGTAAAA	34
Spc17	CCCAACGCTAACGCTAGTTGATTAGCGTCAGGGA	34
Spc18	ACGCAGTTGAGTATCAAGAAATTACATCCGCGA	33
Spc19	CCCCGCTGGAAGTATCGATTAATGGGAAGCTTG	33
Spc20	CATATCAGCAACAATGGATTGCACCTGTCACTG	33
Spc21	GCGGCGCTGGTTGCCATTGAAAAGGTATCAGCAG	34
Spc22	ACACAATAAAACCGTGGAGGATTTATGCCGTCG	33
Spc23	CTCTGGATGACGGTCAACCCAGCTGCCGGAAGAA	34
Spc24	TAGACTGGGCGGAGTTTGACAAGCTATGCGGGA	33
Spc25	TCACGTTATTTGGCATAAGCCTGGCGGCGGTAT	33

Spc26	GCTGCCGTGCCGTCTGTTCCGCCAACGGCAAAT	33
Spc27	CCACACCAGCTCAGATAGAAGCATGACACCCAA	33
Spc28	TATATGCAGGCTGAGAAAGCGCGGGCGGGTCTA	33
Spc29	CAAGGTCATGGCGGATTCGCTGGCAACTCAGAGC	34
Spc30	CGCTTATAAATTAGAAACGATGCAGTGGGTCAA	33
Spc31	CGAACCAGCCAAACGCCGGTAGCTTTCTGTTCC	33
Spc32	TCGTCTTGCCGCGCCAAGGGAATCAACGCCTAT	33
Spc33	TCCGCATCCCACAAGATCGGTGAGAACCTCGTCG	34
Spc34	TCCATTCGAATGTCCTGCGCCACATGCCTGCCT	33
Spc35	CACCGTCGTCGCCATAGCTTTTAGCTCTGTGAG	33
Spc36	CACCGGCAAAGTCATAGCTTTTAGCTCCGTGAG	33
Spc37	CAAAAGCCGCGTCGAAAGGCACATATACTTCCG	33
Spc38	CGGAAGTATATGTGCCTTTCGACGCGGCTTTTG	33
Spc39	CTCACGGAGCTAAAAGCTATGACTTTGCCGGTG	33
Spc40	CTCACAGAGCTAAAAGCTATGGCGACGACGGTG	33
Spc41	AGGCAGGCATGTGGCGCAGGACATTCGAATGGA	33
Spc42	CGACGCGGCTGGTTTTGGCCTTTCTACGGTCAA	33
Spc43	AATCGCGAGCGAGCGGGCTGGCTCTGGCTGCT	32
Spc44	CCTCTGGCTGATCATCCCAAATGTTCGGAAGC	32
Spc45	CCGGCAAAGTCATAGCTTTTAGCTCCGTGA	30
Spc46	AAAGCCGCGTCGAAAGGCACATATACTTCC	30
Spc47	CTGTATGCCCGGGACACTCGGAGACCTCGGTC	32
Spc48	TACGGGCCCAGAGTCAGGCCGATGTGGAGGGT	32
Spc49	TACCAATCTCCAAGGAACGACCGAAGCCGTG	31
Spc50	GATGCACGCGACATTCCGCGCGCTGGCGAGA	31
Spc51	CGTCATCGCACCACCAGCCAATCCGGTATAA	31
Spc52	CGACGAGGTTCTCACCGATCTTGTGGGATGCGGA	34
Spc53	GTTGCTGACCGCATTTTTACAAAGCTTGACAC	32
Spc54	GCTGATTCACTGGCAACTCAGAGCACTGACAA	32
Spc55	TCTACTTTCGTCTGCGTTGGTATCAGCTCCCA	32
Spc56	TCGCCTGAGTTAAAAGCAAGGCCGTATAAAGT	32
Spc57	TATCAGGTTTCATTTTTTTTCTCTCTCCCTTGA	33
Spc58	TGTCATTTCATGTGTCATTTTTGTATCCTCTTG	33
Spc59	TATTTCTTCGGTGGCATATCAGAATTTGAGCTTA	34
Spc60	TCGCGTCCTCCTCATCCACTATCCCCGCACCGTA	34

Spc61	GGCAGGCTAACGATATGCAATCAGATAGTTGG	32
Spc62	TAAGGCCGTCAGTCTGAGAGATTCGTTCATGTGA	34
Spc63	TAGACTGGGGTGAGTTCGACAAGCTATGTATGA	33
Spc64	TAGATGGCTGGGGTGTCATGCTGCCATCCTCGC	33
Spc65	TGGATACAATGGACGATGGACCGCTGGAAAGGT	33
Spc66	TAGAGAGAGGAACGATCTCCTCGACCTATCCCG	33
Spc67	CAGCAGCAGCGTAGAAAAGCAGCTGCGCATTTTC	34
Spc68	CTGTGCCCGGGGCTTTTTTCCGGGGGTGGGCTTA	34
Spc69	CATGCATCAAGACGTTATCACATCGCTATTTAG	33
Spc70	TGTAGAGAGAGGTCTTCCTGCACTTATTCCGTTTA	35
Spc71	CCACACGAGCAACGATGGCAGCATGACACCCAA	33
Spc72	TGCGGTACGACGCGGATGGCCTAGGGGCCGGGG	33
Spc73	CAAGAGGAGTACGAGATCAAGGCTGAGGAGGAGG	34
Spc74	CTAAGTCTACATCCTCCGCGTCATCAAACAGGG	33
Spc75	TATCTGTCATATCGTCGCACAACAATATAGGCA	33
Spc76	TCAGACTTGTATGTGCTCCCAGCAGGAATAATA	33
Spc77	CGAAAGGGCCTTGAACGGGCATACTGGGGTAGC	33
Spc78	GGCAATTAGGTTTTTAAGTCCGCTCATTGCAG	32
Spc79	TGGAACAAGCACAGAGGGAGCGATAATGGCCGCA	34
Spc80	TAAGCCCCCCGGAAATAAGCCCCCGGACCTCTCG	34
Spc81	CTAAGATATAGCTTGCAGGTTAATTATATTTTG	33
Spc82	CCAACAACAACCATGAGACACTACTACGCTTTA	33
Spc83	TTGCACATGTTTTTATATCAGGCTTCTTCGGGTG	34
Spc84	CTTCAGGGGAGCATGGAACTCGCTTTCCGGGGC	33
Spc85	CAAAAGCCAGATCATCCTCACCCGACACCATAC	33
Spc86	TGTATACGCGTCAATTTTTTACAATGACACTCC	33
Spc87	CTGCTACGGAGGCGTTGCGGAGCAGGCGCAAAGC	34
Spc88	TCTCCGACTCTGTCAAAGAAAGCCTTGCCGGAG	33
Spc89	CCCAAGGTGGAACCCTGTCGA	21
Spc90	CACCACGTCGCTCTCGTCGACGCTTAAAGCCAT	33
Spc91	CCAGGCGTGGATATGGCCGGCGATAGCCTTCCG	33
Spc92	TTATCGGACTCGCGACGGAGCTGCTCAGTCTCG	33
Spc93	TGGGATCAATGTCGTCGTCGGCAAGACCGATGC	33
Spc94	TGGAGATCAAGCTACTGAGCCTCCATCTCAAAA	33
Spc95	CGGGAGATTTATGGGACAAGAAATGACGGCAGG	33

Spc96	CAGCCGTGCGCTCGGTATCGGAGTATGTTGCAA	33
Spc97	TCAGAAAGAACGCAAGCACTGGCGATGCTGAAG	33
Spc98	CGTTCTGGCGGCCGTCTTCGACTTCGCCACTGGGA	35
Spc99	TAGAGAGCTCAGGGCGGAGTGGGCCATCGTCAA	33
Spc100	CATCGGCATATTTGACGCTATCAACCTCGTCGT	33
Spc101	TACGGAGCCCGACACCTCCGCGCTTGAAGCCGA	33
Spc102	CGGCAAGACGACGATGGACGAGCTTGGGTCCAA	33
Spc103	CTCCACATCGCTCTCGTCGACGCTTAAAGCCAT	33
Spc104	CGACAAAGCGCTATCAGTGTGCCACCCGAACGA	33
Spc105	CTCGACGGAGTTGATGAAGTCGGACACGACCGA	33
Spc106	TAGGCCTCGTACACGATGGTGTGCCCGCCACGG	33
Spc107	CGACAGCCCTCATTTTCTCGGCACTTGTCGAAT	33
Spc108	TACTTTCCGTATCGATGTGGGGGGTGATTCCGA	33
Spc109	CAAGTCCCTCCTCTTCCATATACTTGAACTTCT	33
Spc110	TTATCTTGCCTGTCTCGGCCTGCTCTTTTGCA	32
Spc111	TGCTACCGGGCCGGAATCGACAGAAAAGGCATG	33
Spc112	CATGAGCTCGTCCCGATGCAAAAGCCTCTCCTG	33
Spc113	TCTCCATCAGCCACCCTACCCGAATCGCCGCACG	34
Spc114	CAAAAAGCCGGAAAATCGGGATCAAATTTCTCA	33
Spc115	CGAGACGGGCGGGGGGCTGACTGGCGCGCTGGA	33
Spc116	TAGTAAAGACTCGCCCGATCCATGCTGCGTCAGG	34
Spc117	CACCCGAAGAAGCCTGATATAAAAACATGTGCAA	34
Spc118	TAAAGCGTAGTAGTGTCTCATGGTTGTTGTTGG	33
Spc119	CAAAATATAATTAACCTGCAAGCTATATCTTAG	33
Spc120	CGAGAGGTCCGGGGGCTTATTTCCGGGGGGCTTA	34
Spc121	TGCGGCCATTATCGCTCCCTCTGTGCTTGTTCCA	34
Spc122	CTGCAATGAGCGGACTTAAAAACCTAATTGCC	32
Spc123	GCTACCCCAGTATGCCCGTTCAAGGCCCTTTCG	33
Spc124	TATTATTCCTGCTGGGAGCACATACAAGTCTGA	33
Spc125	TGCCTATATTGTTGTGCGACGATATGACAGATA	33
Spc126	CCCTGTTTGATGACGCGGAGGATGTAGACTTAG	33
Spc127	CCTCCTCCTCAGCCTTGATCTCGTACTCCTCTTG	34
Spc128	CCCCGGCCCCTAGGCCATCCGCGTCGTACCGCA	33
Spc129	TTGGGTGTCATGCTGCCATCGTTGCTCGTGTGG	33
Spc130	TAAACGGAATAAGTGCAGGAAGACCTCTCTCTACA	35

Spc131	CTAAATAGCGATGTGATAACGTCTTGATGCATG	33
Spc132	TAAGCCCACCCCCGGAAAAAAGCCCCGGGCACAG	34
Spc133	GAAAATGCGCAGCTGCTTTTCTACGCTGCTGCTG	34
Spc134	CGGGATAGGTCGAGGAGATCGTTCCTCTCTCTA	33
Spc135	ACCTTTCCAGCGGTCCATCGTCCATTGTATCCA	33
Spc136	GCGAGGATGGCAGCATGACACCCCAGCCATCTA	33
Spc137	TCATACATAGCTTGTCGAACTCACCCCAGTCTA	33
Spc138	TCACATGAACGAATCTCTCAGACTGACGGCCTTA	34
Spc139	CCAACTATCTGATTGCATATCGTTAGCCTGCC	32
Spc140	TACGGTGCGGGGATAGTGGATGAGGAGGACGCGA	34
Spc141	TAAGCTCAAATTCTGATATGCCACCGAAGAAATA	34
Spc142	CAAGAGGATACAAAAATGACACATGAAATGACA	33
Spc143	TCAAGGGAGAGAGAAAAAAAATGAAACCTGATA	33
Spc144	ATAGGCGTTGATTCCCTTGGCGCGGCAAGACGA	33
Spc145	GGAACAGAAAGCTACCGGCGTTTGGCTGGTTCG	33
Spc146	TTGACCCACTGCATCGTTTCTAATTTATAAGCG	33
Spc147	GCTCTGAGTTGCCAGCGAATCCGCCATGACCTTG	34
Spc148	TAGACCCGCCCGCGCTTTCTCAGCCTGCATATA	33
Spc149	TTGGGTGTCATGCTTCTATCTGAGCTGGTGTGG	33
Spc150	ATCAGGTTTCATTTTTTTTCTCTCTCCCTTGA	32
Spc151	GTCATTTCATGTGTCATTTTTGTATCCTCTTG	32
Spc152	ATTTCTTCGGTGGCATATCAGAATTTGAGCTTA	33
Spc153	CGCGTCCTCCTCATCCACTATCCCCGCACCGTA	33
Spc154	GCAGGCTAACGATATGCAATCAGATAGTTGG	31
Spc155	AAGGCCGTCAGTCTGAGAGATTCGTTCATGTGA	33
Spc156	AGACTGGGGTGAGTTCGACAAGCTATGTATGA	32
Spc157	AGATGGCTGGGGTGTCATGCTGCCATCCTCGC	32
Spc158	GGATACAATGGACGATGGACCGCTGGAAAGGT	32
Spc159	AGAGAGAGGAACGATCTCCTCGACCTATCCCG	32
Spc160	AGCAGCAGCGTAGAAAAGCAGCTGCGCATTTTC	33
Spc161	TGTGCCCGGGGCTTTTTTCCGGGGGTGGGCTTA	33
Spc162	ATGCATCAAGACGTTATCACATCGCTATTTAG	32
Spc163	GTAGAGAGAG	10
Spc164	TTCAAGCGCAGCGCATCCGTCCCGGTCACATAT	33
Spc165	GGAACGGCAGCGGTCAATATCGTTAAGGGAGCA	33

Spc166	ACGTGCTATACTGCGCCTGGGCCTGCTG	28
Spc167	TCCGCGAAGAGGTGGCCAAAGACTACCT	28
Spc168	ACCAGCTTCCCTTTAATCGATACATCCAGAGGCACTAGTTTTTGGCTA	48
Spc169	GACAGCTTCCCATCAATCGATACTTCCAGTGGGGAAAATTTTGGTTA	47
Spc170	TGCTACCGGGCCGGAATCGACAGAAAAGGCAT	32
Spc171	CATGAGCTCGTCCCGATGCAAAAGCCTCTCCT	32
Spc172	TCTCCATCAGCCACCCTACCCGAATCGCCGCAC	33
Spc173	TGCTGCGTGGCAGGAGTATAGCCGCGGGGTAAG	33
Spc174	CTACGGGCCCAGAGTCAGGCCGATGTGGAGGGT	33
Spc175	TTGCTGGAAGATAAACCAGAGATAGCCGGTCA	32
Spc176	GTCAGCGCGGGTCCGTCGTAAACTGATGAAGC	32
Spc177	GGAATGGGGCTGTGGCAAGCTATGGTACTCT	31
Spc178	ATAACTGCACATCACTAACCAGCTTCCCCTT	31
Spc179	ACCCTCCACATCGGCCTGACTCTGGGCCCGTAG	33
Spc180	CCTGACGCAGCATGGATCGGGCGAGTCTTTACTA	34
Spc181	TCCAGCGCGCCAGTCAGCCCCCCGCCCGTCTCG	33
Spc182	TGAGAAATTTGATCCCGATTTTCCGGCTTTTTG	33
Spc183	CGTGCGGCGATTCGGGTAGGGTGGCTGATGGAGA	34
Spc184	CAGGAGAGGCTTTTGCATCGGGACGAGCTCATG	33
Spc185	CATGCCTTTTCTGTCGATTCCGGCCCGGTAGCA	33
Spc186	TGCAAAAGAGCAGGCCGAGACAGGCAAGATAA	32
Spc187	AGAAGTTCAAGTATATGGAAGAGGAGGGACTTG	33
Spc188	TCGGAATCACCCCCCACATCGATACGGAAAGTA	33
Spc189	ATTCGACAAGTGCCGAGAAAATGAGGGCTGTCG	33
Spc190	CCGTGGCGGGCACACCATCGTGTACGAGGCCTA	33
Spc191	TCGGTCGTGTCCGACTTCATCAACTCCGTCGAG	33
Spc192	CTTACCCCGCGGCTATACTCCTGCCACGCAGCA	33
Spc193	TCGTTCGGGTGGCACACTGATAGCGCTTTGTCG	33
Spc194	ATGGCTTTAAGCGTCGACGAGAGCGATGTGGAG	33
Spc195	TTGGACCCAAGCTCGTCCATCGTCGTCTTGCCG	33
Spc196	TCGGCTTCAAGCGCGGAGGTGTCGGGCTCCGTA	33
Spc197	ACGACGAGGTTGATAGCGTCAAATATGCCGATG	33
Spc198	TTGACGATGGCCCACTCCGCCCTGAGCTCTCTA	33

Spc199	CATAGTGCGTCCTGATCATCGCTCCGAAGCTGT	33
Spc200	TCCCAGTGGCGAAGTCGAAGACGGCCGCCAGAACG	35
Spc201	CCTGCCGTCATTTCTTGTCCCATAAATCTCCCG	33
Spc202	TTTTGAGATGGAGGCTCAGTAGCTTGATCTCCA	33
Spc203	GCATCGGTCTTGCCGACGACGACATTGATCCCA	33
Spc204	CGAGACTGAGCAGCTCCGTCGCGAGTCCGATAA	33
Spc205	CGGAAGGCTATCGCCGGCCATATCCACGCCTGG	33
Spc206	ATGGCTTTAAGCGTCGACGAGAGCGACGTGGTG	33
Spc207	CCACTTCCGAATGGCCCTGATAATCTTCTTATTG	34
Spc208	CGACACCAACGGGCAGGGTGCCCTACAGTCAGG	33
Spc209	CTAAATCGGCAAGATTGCTCGTTCTCCGTGCCA	33
Spc210	TCGACAGGGTTCCACCTTGGG	21
Spc211	CTCCGGCAAGGCTTTCTTTGACAGAGTCGGAGA	33
Spc212	ATCTTCTCGTCAAGCCGGTTGATCGCTGTCACA	33
Spc213	ACGACGGTGCGCAGCACCGAGATTTGCTGCCGG	33
Spc214	GTACTGCTTGCAAAGCGGCGTCCTGACCTTTGA	33
Spc215	AGTCAGGCCGATGTGGAGGGTTATGAGCAGCA	32
Spc216	CGGGACCAGAAACGTACTTGACGACCACGCCTTA	34
Spc217	CCCTGAAAACTCCCTACCGTCGCACCGAAATCG	33
Spc218	CCCCATGGCTCACCCCGATCTTCAACGCCGCCG	33
Spc219	ATCACAAACCTTGTCGAAAAGCCCCGTGAATGG	33
Spc220	CAGGAAGTTTATCGTCGTTCGTCACGAAGCCAG	33
Spc221	GGATAGCTGCGCTACTTCTTGTCGCCCTCACGA	33
Spc222	CTCCGCCGCAACGAGGCAGCAATCGCGGCAGTG	33
Spc223	CGCCTGGTGTATGTGCCGCTGACGCAGGGGCAG	33
Spc224	GCTTTGCGCCTGCTCCGCAACGCCTCCGTAGCAG	34
Spc225	GGAGTGTCATTGTAAAAAATTGACGCGTATACA	33
Spc226	GTATGGTGTCGGGTGAGGATGATCTGGCTTTTG	33
Spc227	GCCCCGGAAAGCGAGTTCCATGCTCCCCTGAAG	33
Spc228	GGTTGGCCAACTCTCGCCGGTCGGACGCGTTCGG	34
Spc229	TGACGCGTCGGCGGGGTCGCTATGTCGCCCGTGG	34
Spc230	TCTGCACGGTATACAATCCCCCGCCCCCGGTCA	33
Spc231	ACGAGTTTCGGGCCATTTAGGGCGGGGGTGAG	32
Spc232	CCCAGCGGGTGGTTGTCTTGGCGGCGTGCAGC	32
Spc233	CTGGCGGATCTCAGAGCGTGGCGGCTCGGGTGC	33

Spc234	CGCTGTGTCGATAACGACCGCAATCTCATCTAG	33
Spc235	TGCCTGTTGAATAATCGTAAACGCGTTAAATGA	33
Spc236	TTCGATACCGCGATTGTTTTGAGTGGTGTTCAG	33
Spc237	TAAGCCCTCGCTTAGTAGGTATTCTTTCCCGTCA	34
Spc238	TGTAGTTACCATTAGCCTGTCTATTTTTACATA	33
Spc239	CTATTTTCTATCTTTTTTCTTCGTGCCCCAGCC	33
Spc240	CTTATCTTCTGGAAAAGAAAAAGATGTCGTAGA	33
Spc241	CAAAGCCGAGTTCTACCGCCGCGCCGAGGAGAAG	34
Spc242	TGGTATGGAGAATTGACATCCTTTGATCCAAAA	33
Spc243	TTGATATTGAGAAGTTGAAAGGGGAGGTTGATC	33
Spc244	CATACGACGCGCACAATACATTTTAGCGACGAG	33
Spc245	CAATATCTTGGTCAAAGGGACCAGCAAGATCTCA	34
Spc246	TGGGAAGCTGGTTAGTGATGAGCAGCCACGCCA	33
Spc247	CAAGAGCCCGACGACCGTTTTTGCTCTTATTGT	33
Spc248	TGAAGATAGTGAGGTACGCCAGTAGCACGGTTG	33
Spc249	TTCTAGGTTGATACGATGGCAGAGAAGATCCACC	34
Spc250	TTACGGCTTCACAAGTAACGCATCTGTCACCACA	34
Spc251	CTACGACATTGACGATTTCACCATTGGCTATAT	33
Spc252	CAAGCTTCCCATTAATCGATACTTCCAGCGGGA	33
Spc253	CTAAACGAGGAGCACAGCATGAATCATGAACAG	33
Spc254	TGGCACGGAGAACGAGCAATCTTGCCGATTTAG	33
Spc255	CCTGACTGTAGGGCACCCTGCCCGTTGGTGTCG	33
Spc256	CAATAAGAAGATTATCAGGGCCATTCGGAAGTGG	34
Spc257	ACAGCTTCGGAGCGATGATCAGGACGCACTATG	33
Spc258	CTACGGGCCCAGAGTCAGGCCGATGTGGAGGGT	33
	CRISPR repeats	
Rpt1	GAAACACCCCCACGAGCGTGGGGAAGAC	28
Rpt2	AAAACACCCCCACGAGCGTGGGGAAGAC	28
Rpt3	GTCTTCCCCACGCTCGTGGGGGTGTTTC	28
Rpt4	GTCTTCCCCACGCTCGTGGGGGTGTTCA	28
Rpt5	GTCTTCCCCACGCTCGTGGGGGTGTTTCT	29
Rpt6	GGTCTTCCCCACGCTCGTGGGGGTGTTTCCA	31
Rpt7	GAGCGTGGGGAAGACGA	17
Rpt8	GAAACACCCCCACGAGCGTGGGGAAGACGC	30
Rpt9	GAAACACCCCCACGAGCGTGGGGAAGACAC	30

Rpt10	GCGTGGGGAAGAC	13
Rpt11	AGAAACACCCCCACGAGCGTGGGGAAGAC	29
Rpt12	GGAAACACCCCCACGAGCGTGGGGAAGAC	29
Rpt13	GTCTTCCCCACGCTCGTGGGGGTGTTTCG	29
Rpt14	GTCTTCCCCACGCTCGTGGGGGTGTTTCC	29
Rpt15	GTCTTCCTGCACTTAT	16
Rpt16	CTCGTGGGGGTGTTTCCCT	19
Rpt17	CCGGTCTTCCCCACGCTCGTGGGGGTGTTTCCCT	34
Rpt18	CCGGTCTTCCCCACGCTCGTGGGGGTGTTTCCAT	34
Rpt19	CGCGTGGTATGGCTGCTCATCACTA	25
Rpt20	CGCGTGGTATGGCTGCTCATCGCTA	25
Rpt21	CGCGTGGTTTAGCTGCTCATCGCTA	25
Rpt22	AGTCTTCCCCACGCTCGTGGGGGTGTTTC	29
Rpt23	GGTCTTCCCCACGCTCGTGGGGGTGTTTC	29
Rpt24	AGGAAACACCCCCACGAGCGTGGGGAAGAC	30
Rpt25	GGGAAACACCCCCACGAGCGTGGGGAAGAC	30
Rpt26	AAGAAACACCCCCACGAGCGTGGGGAAGAC	30
Rpt27	GAAACACCCCCACGAGCGTGGGGAA	25
Rpt28	GTCTTCCCCACGCTCGAGGGGGTGTTTCC	29

Appendix I

Physical characteristics of Ace Lake and environmental factors associated with the Vestfold Hills

Table I1: The physicochemical characteristics of Ace Lake and environmental characteristics of the Vestfold Hills. * The background colours of the sample collection dates in the first column reflect the season — summer (red), winter (blue), and spring (green). Ace Lake physical characteristics included lake depth, salinity, lake temperature, DOC, and ice cover thickness. DOC was normalized across sample collection time-periods by keeping DOC at 5 m depth as 100% and recalculating the DOC for the rest of the depths in a time period. Environmental factors measured at Davis Station in East Antarctica included air temperature, sunlight hours, and maximum wind velocity (data obtained from Australian Antarctic Data Centre) and daylength (data obtained from a web service https://www.timeanddate.com). The monthly average values were the means of the values observed in a month (Chapter 3 section 3.2.4.2) and were used for the statistical analysis of the Ace Lake metagenomes using PRIMER v7. NM, not measured.

Sample collection date*	Depth (in m)	Salinity (‰)	Lake temperature (°C)	DOC (%)	On sample collection date			Monthly average values			Maximum wind velocity (km/h)	Ice cover thickness
					Air temperature (°C)	Sunlight (h)	Daylength (h)	Air temperature (°C/day)	Sunlight (h/day)	Daylength (h/day)		
20 Dec 2006	5	22	1	100	-1	9	24	-0.1	10	24	24	NM
20 Dec 2006	11.5	22	0.3	94	-1	9	24				24	
20 Dec 2006	12.7	28	3	45	-1	9	24				24	
20 Dec 2006	14	32	2	11	-1	9	24				24	
20 Dec 2006	18	35	3	2	-1	9	24				24	
20 Dec 2006	23	42	3	-2	-1	9	24				24	
19 Nov 2008	5	22	-0.4	100	-5	0.2	22	-4	4	21	80	Ice thickness 1.8 m
21 Nov 2008	11.8	22	-0.3	95	-3	1	22				70	
21 Nov 2008	12.8	26	3	22	-3	1	22				70	

21 Nov 2008	14.1	31	3	7	-3	1	22				70	
21 Nov 2008	18	34	3	6	-3	1	22				70	
23 Nov 2008	23	40	3	11	-2	0	23				81	
24 Nov 2013	5	21	-0.2	100	-5	4	24				31	
25 Nov 2013	12.5	23	1	104	-3	2	24				39	
26 Nov 2013	13.5	30	3	84	-2	18	24	-4	8	21	55	Completely covered by thick ice
26 Nov 2013	15	33	4	18	-2	18	24				55	
26 Nov 2013	19	36	3	0	-2	18	24				55	
27 Nov 2013	24	42	2	0	1	3	24				72	
17 Dec 2013	0	16	NM	NM	-1	1	24	-0.4	10	24	22	Completely covered by thick ice
15 Feb 2014	0	7	3	NM	0.5	8	17	-2	7	17	43	Half covered by ice
2 Jul 2014	5	15	NM	NM	-22	0	0				26	
3 Jul 2014	12.5	21	NM	NM	-22	0	0	-20	1	2	24	Completely covered by thick ice
3 Jul 2014	13.5	29	NM	NM	-22	0	0				24	
20 Aug 2014	5	19	2	100	-25	1	8				26	
21 Aug 2014	13	24	4	78	-23	0	8	-17	3	7	57	Completely covered by

21 Aug 2014	14.5	27	4	80	-23	0	8				57	more than 1 m ice
20 Oct 2014	5	19	1	100	-10	8	16	-10	7	15	31	Completely covered by ~2 m ice
20 Oct 2014	12	21	2	90	-10	8	16				31	
21 Oct 2014	13	24	4	90	-10	11	16				35	
21 Oct 2014	16	27	4	10	-10	11	16				35	
21 Oct 2014	19	25	3	6	-10	11	16				35	
21 Oct 2014	24	34	2	2	-10	11	16				35	
4 Dec 2014	5	21	3	100	-3	11	24	0.3	10	24	26	~1.8 m ice, very north edge starting to melt behind island
4 Dec 2014	12	22	2	100	-3	11	24				26	
4 Dec 2014	13.4	29	5	96	-3	11	24				26	
4 Dec 2014	14	31	5	88	-3	11	24				26	
3 Dec 2014	19	35	3	2	-1	6	24				55	
3 Dec 2014	24	40	2	0	-1	6	24				55	
8 Jan 2015	0	5	NM	NM	2	0	24	0.8	9	23	74	Mostly covered in poor quality ice
27 Jan 2015	0	10	2	NM	3	9	20	0.8	9	23	65	No Ice