From organism diversity to micro-heterogeneity: confident assessment of fine-scale variation within metagenomic data

Author:
Amos, Timothy

Publication Date:
2011

DOI:
https://doi.org/10.26190/unsworks/15386

From Organism Diversity to Micro-heterogeneity: Confident Assessment of Fine-scale Variation within Metagenomic Data

Timothy Graham Amos

A thesis submitted in fulfilment of the requirements for
the degree of Master of Science (Research)

School of Biotechnology and Biomolecular Sciences
Faculty of Science
The University of New South Wales
Sydney, Australia

April 2011

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.’

Signed …….....

Date …….....

Copyright Statement

Signed …….....

Date …….....

Authenticity Statement

Signed …….....

Date …….....

Abstract

The metagenome of a microbial community contains a large quantity of information about the inter-strain genetic variation present in that community. Genome assemblers using algorithms designed for use with isolate genomes obscure the inter-strain variation within metagenomic data. Analysing this variation in metagenomic data is further complicated by sequencing errors that add noise to the system by making base assignments ambiguous.

In order to develop improved computational methods for metagenome analysis, simulations were performed using genome data of individual species. A software program, *MetaSim*, was used to generate simulated reads. Assemblies of these reads were used to investigate the development of an error model to confidently identify SNPs (Single Nucleotide Polymorphisms). This approach proved limited due to the nature of the *MetaSim* software and the insufficient availability of consistent, well-documented data.

As an alternative approach, a graphical analysis of unitigs (high confidence contigs) was developed. This approach provided accurate predictions of whether each unitig in an assembly of simulated reads consisted of only one strain, or more. The approach included developing a system of rules describing the relationship between the number and proportions of strains in an assembly and the positioning of clusters in scatter plots. The differences in densities of clusters were used to help distinguish between ambiguous cluster patterns. Idealised assemblies of simulated reads without sequencing errors were produced, to examine how sequence quality affects the ability to make inferences about inter-strain variation. Computational clustering was investigated as a means of automating the analysis.

Having established an approach to analyse unitigs, environmental metagenome data was analysed. This graphical analysis provided a well-supported and parsimonious interpretation of the number of strains present in metagenome data of an Antarctic lake community, and their proportions.

Acknowledgements

Thank you to my Supervisors Matt, Rick, Fede, and for a little while Torsten. Matt, I've really appreciated all your help. Thanks, for all the help with day to day stuff, for all your guidance and for being enthusiastic about my project. Thanks, for staying back late the day before multiple presentations. Thanks, for fixing *toAmos* and for all your proofreading. Thanks Rick, for finding me a project and a scholarship. Thank you for help with presentations and proofreading even though I rarely explained things well and didn't always take your advice. I appreciate it. Thanks for helping guide the overall project and for persevering with me. Thanks Fede, for proofreading, even though you were really busy, and including me in your group meetings. Thanks Torsten, for having an open door and a listening ear.

To my awesome girlfriend Laura: I don't know how I would have managed without you. Thanks for advice on presentations and supervision, encouragement to work hard, constant support, a listening ear even when you didn't always follow, advice and help with thesis writing etc. etc. Thank you for all the proofreading and for all the help making my sentences less awkward. I'm sorry I'm such a slow learner sometimes.

Thanks Sheree, for listening to what I'm up to and for proofreading my thesis. Thanks for discussions about reports, presentations and supervision. Thanks for patiently and sympathetically listening to me during the highs, lows and in betweens of life. Thanks John, for always responding enthusiastically to my *M. frigidum* questions. Thanks Mark, for being willing to chat about 16S stuff. Thanks David, for offering to collaborate and for organising journal clubs. Thanks Wei Hua and Shaun, for discussing maths with me and being interested in my project. Thanks to Daniel Huson, for patiently emailing me in response to what I thought were bugs in *MetaSim*. Thanks to the research students at New College Village who listened to me practice presentations and gave good advice. Thanks Travis and the coffee cart, for my exceptional daily coffee.

Thank you all my friends and my siblings, for providing some fun and relaxation to break up the study and keep me sane. Thanks Drew, for being keen to celebrate with me, sorry I kept having to postpone. Thanks dad, for offering to proofread even though

it’s a bit out of your area of specialisation. Thanks mum and dad, for encouraging me to work hard and being there to chat when I needed it. Thanks, for spoiling me during holidays. Thanks for all your love and prayers. Thanks to everyone who encouraged me and prayed for me while I was stressing out. Thank you God, most of all, for all your love and provision.

Table of Contents

Abbreviations

AMOS	A Modular, Open-Source whole genome assembler
AUC	Area Under ROC Curve
ANI	Average Nucleotide Identity
BLAST	Basic Local Alignment Search Tool
BLASTX	*BLAST* using a translated nucleotide query
BOG	Best Overlap Graph unitigger
$\overline{D}$	Discrepancies in contig per unit of contig length
$\overline{D}'$	Discrepancies in unitig per unit of unitig length
DDH	DNA-DNA Hybridization
FP	False Positive rate
GS	Genome Sequencer
GSB	Green Sulfur Bacteria
IID	Internal IDentifier
JAZZ	JGI in-house ASSembler
JCVI	J. Craig Venter Institute, USA
JGI	Joint Genome Institute, US Department of Energy
JTC	The Joint Technology Center, USA
LUCY	Less Useful Chunks Yank
MEGAN	MEtaGenome ANalyzer
MER	oligoMER overlapper
MNFS	Mean Negative Flow Signal
μ	Mean

MUMmer	Maximal Unique Matches
NCBI	The National Center for Biotechnology Information, National Institute of Health, USA
NR	Non-Redundant protein sequences database
OTUs	Operational Taxonomic Units
PHRAP	PHRagment Assembly Program or PHil's Revised Assembly Program
$\bar{R}$	Reads in contig per unit of contig length
$\bar{R}'$	Reads in unitig per unit of unitig length
ROC	Receiver Operating Characteristic
S	Estimate of the number of Strains that contributed a significant proportion of reads to a unitig
SDNFS	Standard Deviation for Negative Flow Signals
σ	Standard deviation
SNP	Single Nucleotide Polymorphism
SSDM	Signal Standard Deviation Multiplier
SSDSRM	Scale Standard Deviation with Square Root of Mean
TCAG	The Center for the Advancement of Genomics, USA
TIGR	The Institute for Genomic Research, USA
TP	True Positive rate
XML	eXtensible Markup Language

List of Figures

List of Tables

Chapter 1

General Introduction

1.1 Metagenomics

The traditional approach of studying cultivatable microorganisms is limited due to the fact that the majority of microorganisms from natural environments are not able to be cultivated (Kalyuzhnaya *et al.* 2008). Genome databases are thus heavily biased towards certain taxa (Handelsman 2004, Kunin *et al.* 2008). By sequencing microbial communities directly from the environment, metagenomics (alternatively called environmental genomics, ecogenomics or community genomics) provides the opportunity to study underrepresented and uncultured taxa.

Metagenomics also allows the structure and interactions of microbial communities to be investigated. One reason this is important is because microorganisms underpin the majority of the geochemical cycles (Handelsman 2004). As carbon dioxide levels and temperatures continue to increase, understanding how environmental changes will affect these communities will become more important. For example, a quarter of the natural release of the greenhouse gas methane is from understudied permafrost Archaea (Wagner *et al.* 2005). Microbial communities like these could have important feedback effects on the extent of climate change.

With the advent of high-throughput DNA sequencing, abundant species in low to medium complexity microbial communities can be sequenced to a high read depth. This provides the opportunity for unprecedented analyses of the genomic heterogeneity and evolution of microorganisms (Handelsman 2004). To obtain genetic sequence data for an entire genome, a high number of reads is required for each position, on average. If less abundant species are to be assembled, then the more abundant species are consequently sampled deeply.

Metagenomic data has traditionally been assembled by algorithms designed for single clonal genomes (Raes *et al.* 2007). For example, Venter *et al.* (2004) used the *Celera assembler* and Tyson *et al.* (2004a) used *JAZZ* (JGI in-house ASSembler). Other

assemblers used for metagenomes include *PHRAP* (PHRagment Assembly Program or PHil's Revised Assembly Program) and *Arachne* (Raes *et al.* 2007). The *Celera assembler*, *PHRAP* and *Arachne* use the overlap-layout-consensus approach (Zerbino and Birney 2008), as does *JAZZ (Aparicio et al. 2002)*. However, *PHRAP* does not perfectly follow this paradigm (Kunin *et al.* 2008). Assemblers using this approach conventionally assume that any disagreements in the assembled sequence are due to sequencing or assembly errors rather than real genetic variation. Thus, these conventional assemblers focus on consensus sequence as the end product, obscuring intra-strain variation.

1.2 Species, Strains and OTUs

The best way to define the species concept in Bacteria and Archaea is still being debated (Deloger *et al.* 2009, Richter and Rosselló-Móra 2009). The traditional species definition for bacteria is of at least one shared distinguishing phenotypic trait and 70% DDH (DNA-DNA Hybridization) (Wayne *et al.* 1987). Whilst this definition is pragmatic and can be applied across the entire bacterial domain, it has several limitations. These include impractical experiments, a lack of overlap with the Eukaryotic species concepts, inapplicability to metagenomics and too much phenotypic variation within named species (Konstantinidis *et al.* 2006).

Multiple other definitions have been proposed. ANI (Average Nucleotide Identity) is the most promising of these, as it correlates well with DDH (Goris *et al.* 2007, Richter and Rosselló-Móra 2009). Other methods include DNA content, which measures the percentage of DNA that is conserved between genomes (Deloger *et al.* 2009). This can be calculated from either the whole genome or the protein coding portion (Goris *et al.* 2007). However, it has been found to not correlate well with DDH or ANI (Konstantinidis *et al.* 2006). Alternatively, there is maximal unique matches, a method that correlates well with ANI (Deloger *et al.* 2009) but which cannot be used with incomplete draft genomes (Richter and Rosselló-Móra 2009).

Oligonucleotide frequencies can also be used for distinguishing species (Richter and Rosselló-Móra 2009). This involves comparing the frequency of short sequences of bases (most commonly, tetranucleotides) between genomic sequences, and is useful for metagenomics, as it is superior to alternative methods (Teeling *et al.* 2004a).

A strain encompasses separate populations within a species that are distinct groups based on genotypic or phenotypic differences (Dijkshoorn *et al.* 2000). In theory, two strains could be separated by a single SNP. In practice, the equilibrium between selection, genetic drift and recombination between distinct populations should increase divergence beyond the minimum. There are no widely accepted guidelines as to the expected quantity of genetic differences between strains. One pragmatic way of classifying whether genetic sequences belong to the same strain is observing whether they co-assemble using a particular algorithm and parameters. For example, the *Celera assembler* has multiple error rate parameters that can be used to adjust the level of similarity required for sequences or their aggregates to be co-assembled. Even with high error rates, some conserved regions of separate strains will probably co-assemble, which could provide additional information for how similar two organisms are. Chapter 3 describes how the adjustment of these parameters can provide information about the strains present in a metagenome.

OTUs (Operational Taxonomic Units) are groupings of similar genetic sequences within a dataset that can be used as an alternative to taxa such as species and strain. They allow sequences from related organisms to be divided in a self-consistent manner that is free from the biases in the taxonomic tree. However, OTUs still require a phylogenetic classification if they are to be connected to related organisms from other studies, which is a task that currently lacks a consistent method (Schloss and Westcott 2011).

1.3 Variation within Species

In metagenomic sequencing projects, the DNA that is sequenced comes from so many different cells that it is unlikely that any two sequences come from the same individual (Eppley *et al.* 2007). Thus, a higher read depth for a species means more data about microheterogeneity (intra-species variation) is available for analysis.

In 2004, metagenomics was moving from smaller-scale targeted sequencing (Stein *et al.* 1996) to bulk random sequencing of an entire community (Tyson *et al.* 2004a, Venter *et al.* 2004). The comparison of single-strain genome projects had shown that there can be high variation within microbial species (Rodríguez-Valera 2004). Metagenomics had also shown that even populations that appeared uniform could

contain substantial microheterogeneity (Schleper *et al.* 1998). At that time, there was concern about the extent to which metagenomic sequences could be assembled (Tringe *et al.* 2005). This concern was due both to the variation between the species, and within them (Béjà 2004, Rodríguez-Valera 2004). It was known that eukaryotes have similar gene content within and between species, and variation in the sequence of those genes. For example, a comparison of four genomes from the yeast genus *Saccharomyces* found five to nineteen unique genes per species compared to around 6000 shared ones (Kellis *et al.* 2003). It was also known that even closely related bacterial strains could have a large difference in the complement of genes. For example, when the genomes of three *Escherichia coli* strains were compared, only about 40% of the genes were found in all three (Welch *et al.* 2002). Thus, the gene pool of a species could be "orders of magnitude larger than the genome of one strain" (Rodríguez-Valera 2004). Together, this meant that intra-species variation was considered a problem that would hinder the study of microorganisms in their environment (Béjà 2004).

Since then, bulk random shotgun metagenomic sequencing has become widespread and increased in scale largely due to improved sequencing technology (Petrosino *et al.* 2009). The techniques used to sort and assemble metagenomic datasets have been evaluated and refined over time (Mavromatis *et al.* 2007, Kunin *et al.* 2008). Together, this means that microheterogeneity in microbial communities can now be seen as the focus of scientific study rather than a potential obstacle to it.

1.4 Microheterogeneity in Metagenomic Data

For metagenomes of very simple communities, microheterogeneity can be manually analysed. For example, Tyson *et al.* (2004b) analysed an acid mine drainage community with a very dominant species (75%) that had low polymorphism. The low complexity in this case allowed the reads to be divided into groups with similar SNPs by the use of custom scripts. The prevalence of one to three such groups throughout this genome shows that there were approximately three distinct strains present in this metagenome.

Other studies have investigated microheterogeneity on a gene by gene or gene island basis without attempting to separate a population into its composite strains (Coleman *et al.* 2006, Allen *et al.* 2007). Using the data from Tyson *et al.*, a custom-built program, *Strainer*, has been developed to automate the analysis of microheterogeneity through the use of patterns of SNPs to trace strain variants through co-assembled data (Eppley *et al.* 2007). *Strainer* was designed for use with data from a

very simple community and to utilise long, high quality, mated reads. As the authors suggest, it could theoretically be expanded to work with shorter, lower quality, unmated reads. *Strainer* also utilises quality scores which can be less meaningful and accurate in next generation sequence data (Brockman *et al.* 2008). Eppley *et al.* used manual curation on some of their datasets before analysis with *Strainer* and also recommend manual curation to "resolve complicated regions", both of which may prove less viable with larger more complex next generation sequencing datasets.

1.5 Aims

The aim of this thesis was **to determine if DNA microheterogeneity could be detected and accurately quantified in metagenome data from environmental microbial communities**. Chapter 2 describes the investigation of a statistical model of microheterogeneity in metagenomic datasets. Chapter 3 describes inferences about whether metagenomic assembly contigs are clonal. In Chapter 4, inferences about the number of closely related organisms (or strains) in a metagenomic assembly contig are described. Estimations of the relative abundances of those strains are also described. Chapter 5 summarises the findings of this thesis and describes future directions. Appendix A contains supplementary material for Chapter 2, Appendix B for Chapter 3, Appendix C for Chapter 4 and Appendix D (supplied as an enclosed CD) contains source code for the custom scripts mentioned in this thesis.

Chapter 2

Error Model Development

2.1 Summary

The aim of the work in this chapter was to investigate a statistical model of microheterogeneity in metagenomic datasets. The purpose of the model was to assess whether discrepancies (potential SNPs) in the datasets were real variation or systemic noise. This was to be done by assigning a probability to each discrepancy describing how confidently it can be classified as real variation.

Reads from nine different Sanger genome sequencing projects were assembled and analysed. Assemblies of equivalent simulated reads were produced to allow calibration of sequencing errors. *MetaSim* was used to generate simulated reads from genome sequence data. The proportion of substitutions, insertions and deletions could all be calibrated to a single sequencing project. However, the high variation in error rates and missing metadata for the available Sanger reads prevented a general calibration and thus limited the usefulness of this methodology.

To overcome this problem, the calibration was repeated with three assemblies of data produced on 454 Genome Sequencer (GS) FLX machines. For 454 data, it was not feasible to calibrate the different kinds of sequencing errors, thus only the combined error rate was calibrated. This was due to the complex non-linear parameters for modelling error rates in *MetaSim*-produced 454 data. To overcome these problems, a different strategy was developed, as described in Chapter 3. If a strong signal of variation could be detected then a detailed understanding of the nature and quantity of systemic noise would not be necessary.

2.2 Introduction

2.2.1 Sequencing Technologies

2.2.1.1 Sanger Dideoxy Sequencing

Sanger sequencing can provide long reads up to ~1000 bp in length and up to 99.999% per-base 'raw' accuracies (Shendure and Ji 2008). Sanger sequencing can also provide mated reads with a large range of insert sizes.

Given the expense of reagents (especially dyes) in Sanger sequencing, there is pressure to adjust protocols in order to be more economical. As decreasing dye usage can produce significant cost savings in a large facility, there is incentive to use as little as possible, even to the extent that quality might suffer. This can lead to significant drops in sequence quality (DeMaere, personal communication). Other factors have been shown to affect the outcome of Sanger sequencing. For example, the presence of secondary structures in primers or templates can be disruptive (Hirao *et al.* 1992).

2.2.1.2 454 Pyrosequencing

Next generation sequencing technologies like 454 pyrosequencing (454 Life Sciences, Branford, CT, USA) are providing low cost, ultra-high throughput alternatives to Sanger sequencing. The GS FLX system produced 250 bp reads (Shendure and Ji 2008), but this sequencer has been superseded by the GS FLX Titanium (Petrosino *et al.* 2009). 454 technology did not originally provide mated reads (Margulies *et al.* 2005), but they are now available (Mardis 2008).

Compared to Sanger sequencers, 454 sequencers are more likely to misjudge the length of homopolymer repeats. Whilst true substitution errors are very rare in 454 data, adjacent deletions and insertions can be interpreted as substitution errors, and are more common (Mardis 2008, Balzer *et al.* 2010, Balzer *et al.* 2011). All the next generation technologies have much higher error rates than Sanger. Raw Sanger base calls are on average ten times more accurate (Shendure and Ji 2008).

The high through-put capabilities of next generation sequencing technologies make them attractive for metagenomics. The resultant greater read depth provides much more information on variation within species.

2.2.2 Discrepancies and Variation

When assemblers tile reads together they often overlap imperfectly matching reads. For each column of bases from different reads, traditional assemblers assign a consensus base. If there is disagreement between the reads, then these assemblers attempt to choose the optimal base using either sequence identity percentages, quality values, or both (Myers *et al.* 2000, Denisov *et al.* 2008).

Discrepancies (non-unanimous consensus bases) can be due to a variety of causes. Any cause other than microheterogeneity (real mutations within the species of interest) can be treated as part of the noise of the system. A minority base (a base at a discrepancy that does not match the consensus base) may or may not be a real SNP. The probability that this base is a real SNP is increased if that exact base is found in more reads. Thus, if the probability of a particular base being affected by noise can be quantified, then this indicates how confidently it can be regarded as real variation.

Factors that contribute to the noise of the system include sequencing errors, inter-species chimerism and misassembly. The dominant cause for a difference between aligned reads is sequencing errors. Alternatively, the aligned reads may come from different species. Reads not from the species of interest may come from a species native to the environment sampled or from a contaminant. Lastly, the aligned reads may be from distinct but similar regions of the genome and thus incorrectly assembled together. For example, the two regions may contain paralogous genes.

There is another level of variation within a species that can also complicate assembly. Strains from the same species may differ due to deletions, insertions, duplications and rearrangements of varying size (Allen *et al.* 2007). These differences can range from a few bases to the level of genes or even genomic islands containing multiple genes. Heterogeneity on this scale can decrease the degree and accuracy of assemblies and make down-stream analysis more problematic.

Metagenomic samples provide a clear advantage for studying inter-strain variation over isolate sequence data. When strains are assembled together discrepancies are made that can be regarded as high quality SNPs. These SNPs can be mined for information about evolution and population structure (Kunin *et al.* 2008). They contain information about which strains are present in a sample, and how and why they differ (Whitaker and Banfield 2006).

Distinguishing real SNPs from sequencing errors is challenging because the SNPs

can occur at low frequencies whilst the errors often occur at frequencies which are orders of magnitude higher (Shen *et al.* 2010). Furthermore, there are regions of genomes that are much more likely than others to be sequenced incorrectly. For example, in Titanium data, a homopolymer of length five will have a mean length of 4.95 and a standard deviation of 0.39 (Balzer *et al.* 2011). It seems reasonable to assume that older FLX data is even more error prone in this area. The higher frequencies of errors in these areas increase the uncertainty concerning what is true microheterogeneity. Next generation sequencing methods are more prone to errors than Sanger sequencing and compensate for this with higher read depths. These two factors together produce more duplicated errors and thus make the data more complex to interpret.

2.2.3 Simulations

Simulations allow metagenomic microheterogeneity to be explored systematically. Statistical measurements from real sequencing projects can be used to set some of the variables in the simulations and thus make them more realistic. For example, the mean and standard deviation of read length from a real project can be used. Likewise, the rates and kinds of sequencing errors should be controllable and able to be calibrated. Variables that require investigation such as the quantity and proportions of the strains used in the simulation can also be controlled. It should be possible to track every read, simulated mutation and simulated sequencing error from its origin in a genome to its location in a scaffold. For example, the strain of origin of each read can be tracked. The methods needed to extract all the information on microheterogeneity from a noisy metagenomic sample are yet to be developed. By using a simulation, methods can be tested in a system where the amount of intra-species variation is known. Thus, these methods can be evaluated and further developed.

2.2.4 Error Modelling

The aim described in this chapter was to investigate a statistical model of discrepancies. *MetaSim* (version 0.9.1; www-ab.informatik.uni-tuebingen.de/software/metasim; Richter *et al.* 2008) was used to produce datasets of simulated DNA-sequencing reads.

To model discrepancies, the calibration of the proportions of the different kinds of sequencing errors in the simulations was attempted. The model would involve assigning probabilities of whether discrepancies are real variation or systemic noise. This could be done by identifying a threshold above which any microheterogeneous signal can be confidently identified as real variation.

2.3 Materials and Methods

For a simulation to be useful in developing a realistic error model, it requires calibration to data from real sequencing projects. To calibrate *MetaSim* to this kind of data, two data analysis pipelines were constructed. The first was constructed to assemble simulated Sanger reads. This pipeline was also used to assemble experimentally-derived, and simulated, 454 reads. The second pipeline was created to assemble experimentally-derived reads sets downloaded from The National Center for Biotechnology Information (NCBI) trace archive (www.ncbi.nlm.nih.gov; Wheeler *et al.* 2006). These pipelines combine custom *Python* (version 2.4; www.python.org) scripts with third party code. Unit testing on the custom scripts was performed using *PyUnit* (http://pyunit.sourceforge.net).

2.3.1 The 454 and Simulated Sanger Sequence Pipeline

The 454 and simulated Sanger sequence pipeline (*run_pipeline.py*; Figure 2.1) uses *MetaSim* generated FASTA output as its input. *MetaSim* provides only embedded mate-pair data and does not provide any quality scores. Therefore, custom scripts were written to create matching quality (*make_quality.py*) and mate-pair (*make_mate-pair.py*) data files. The mate-pair information for *MetaSim* generated reads was embedded in the identifiers of those reads. For *make_quality.py*, correct bases were assigned a score of 40, while erroneous bases were assigned a score of five. A score of five ($\sim\frac{1}{3}$ probability of error) is just below the most common value for low quality scores and a score of 40 (10^{-4} probability of error) is approximately the most common value for high quality scores (Ewing and Green 1998). Thus, the scores chosen are reasonable scores for individual bases and represent a best case scenario in aggregate. The source code for all custom software mentioned in this thesis is available in Appendix D (supplied as an

enclosed CD).

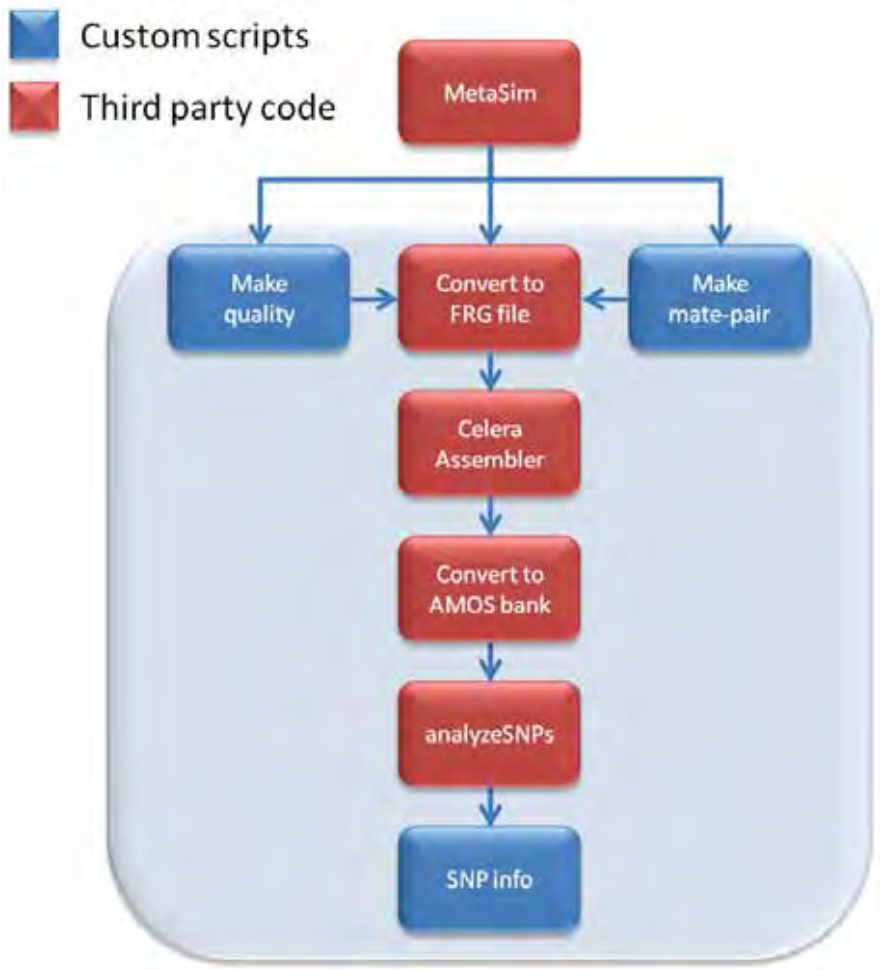

Figure 2.1: The 454 and simulated Sanger sequence pipeline *(run_pipeline.py)*.
Blue boxes denote custom *Python* scripts. Red boxes denote third party computer software. See text for details.

In this pipeline, the sequence, quality and mate-pair information were combined into a FRG file using *convert-FASTA-to-v2.pl*. This script is part of the *Celera Assembler* package (version 5.4; wgs-assembler.sourceforge.net; Myers *et al.* 2000). FRG files are the input format expected by the *Celera Assembler*. Information on the mean and standard deviation of the insert sizes was also incorporated into each FRG file. For 454 data, both the mean and standard deviation were set to zero as none of the 454 reads have mates. The reads in the FRG file were then assembled by the *Celera Assembler*. For 454 data, the *BOG* (Best Overlap Graph) unitigger and *MER* (oligoMER) overlapper were used, as recommended by Celera (http://sourceforge.net 2009Brzuszkiewicz *et al.* 2006). Since *MetaSim* creates reads that do not need trimming, overlap trimming was disabled.

The assembly was converted into an *AMOS* (A Modular, Open-Source whole genome assembler) bank using *toAmos* and *bank-transact*. It was then analysed using

analyzeSNPs. *analyzeSNPs* is one of multiple programs requiring the conversion to the *AMOS* format. These three programs are all parts of the assembly validation pipeline *amosvalidate* (http://sourceforge.net/projects/amos; Phillippy *et al.* 2008) and included in the *AMOS* Assembler package (version 2.0.8). a*nalyzeSNPs* was set to report discrepancies with no minimums for the cumulative quality values or the number of consistent disagreeing reads. These reports were set to display IIDs (Internal IDentifiers) for the contigs and read identifiers.

The list of SNPs reported by *analyzeSNPs* was further analysed by another custom Python script: *SNP_info.py*. This script counts the proportions of substitutions, deletions and insertions in each assembly. It also counts the number of discrepancies where the same minority base appears in at least two reads. Lastly, *SNP_info.py* also calculates the total number of discrepancies and the number per kilobase of consensus sequence.

When experimentally-derived 454 data was used in this pipeline, the quality data used was also experimentally-derived. Sequence was obtained in FASTQ format. This was converted into FASTA and quality files by a custom script *FASTQ_splitter.py* that utilised *Biopython* (http://biopython.org; Cock *et al.* 2009). Multiple FASTQ files were provided for each species. The reads in these files were given unique identifiers by another in-house script (*FASTQ_read_renamer.py*) before they were combined.

2.3.2 The Experimentally-derived Sanger Sequence Pipeline

To assemble Sanger reads from real sequencing projects, a second pipeline was created (*run_pipeline2.py*) (Figure 2.2). This pipeline trims, assembles and analyses experimentally-derived data to allow calibration of *MetaSim*'s parameters. A custom script, *change_identifiers.py*, was used to standardise the read identifiers in the sequence and quality data to match the XML (eXtensible Markup Language) metadata. *LUCY* (Less Useful Chunks Yank) (version 1.19; http://sourceforge.net/projects/lucy; Chou and Holmes 2001) was used to calculate where reads needed trimming. It also listed the low quality reads that should be discarded. The custom script *trim_LUCY.py* was used to trim and filter these reads. This was performed for both the sequence and quality files according to *LUCY*'s calculations. The custom script *cull_LUCY.py* was used to remove the mate-pair information for discarded reads from the mate-pair data

file. The data was then combined and assembled in the same manner as for the 454 and simulated Sanger sequence pipeline.

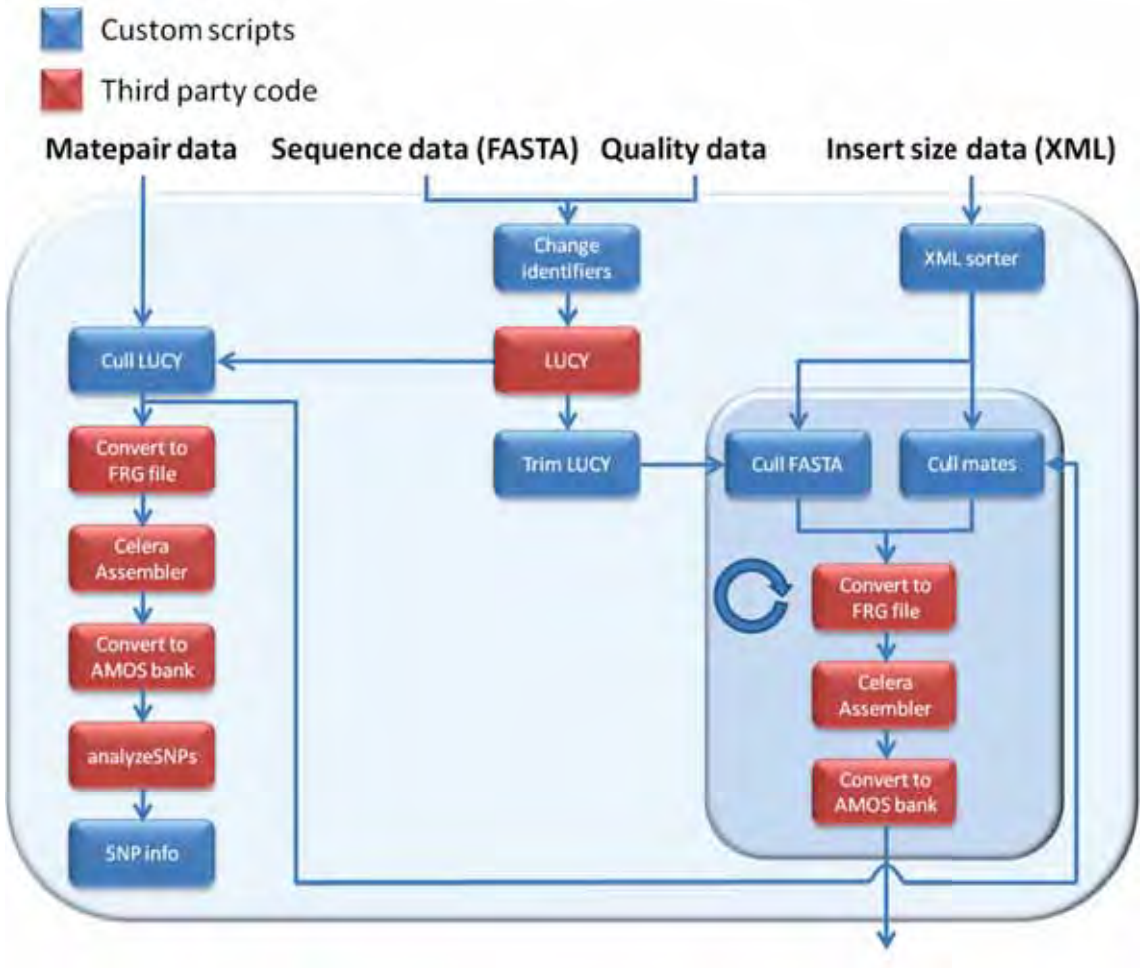

Figure 2.2: The experimentally-derived Sanger sequence pipeline *(run_pipeline2.py)*.
Blue boxes denote custom *Python* scripts. Red boxes denote third party computer software. See text for details.

Where possible, the sizes of the inserts used to create mate-pairs were obtained from the XML metadata file. The metadata file should list the library each read came from and that library's mean insert size. Libraries with the same insert size listed were designated as a group of libraries. Some of these groups contained only one library. Every genome project investigated contained libraries with different insert sizes and thus multiple groups. A custom script, *XML_sorter.py*, was created to sort the reads into these groups of libraries. Separate assemblies were made for each group in the same manner as for the 454 and simulated Sanger sequence pipeline. Another in-house script, *cull_FASTA.py*, was used to remove all reads that did not belong to a particular group. These reads were removed from the trimmed sequence and quality data for that assembly. Likewise, references to these reads were removed from the quality-culled

mate-pair data by the custom script *cull_mates.py*. The assembly of each group of libraries was used to obtain the mean and standard deviation of the insert sizes for that group. It was also used to obtain the group's numbers of mated and unmated reads. These statistics were obtained from *Hawkeye* (http://sourceforge.net/projects/amos; Schatz *et al.* 2007), part of the *AMOS Assembler* package. *MetaSim* also requires the mean and standard deviation of the read lengths for each group simulated. These numbers were obtained by the *Perl* script *getlengths*, also part of the *AMOS Assembler* package. This third party script was called by the custom script *get_stats.py*. *get_stats.py* took the statistics from *Hawkeye* and converted them for use in *MetaSim*. The numbers of mated and unmated reads were converted into the number of mate-pairs and proportion of reads with mates. *get_stats.py* outputs all this information in the *.mprs* format used by *MetaSim*. These statistics could then be used for replicating assemblies of experimentally-derived sequence data in a simulated assembly.

To investigate the reasons for differences in error rates between assemblies, the GC content and coverage of each assembly were calculated. The GC content and coverage of each genome project was calculated using numbers from *Hawkeye*. The mean read length in each assembly for mated and unmated reads were multiplied by the number of reads in their respective categories. These combined read lengths for each category were added together to give the combined length of all reads, i.e. mean read length times number of reads. The combined read length for each assembly was then divided by the corresponding total number of consensus bases to give the coverage in that assembly. The GC content for each category in an assembly was multiplied by the combined read length for that category. The resulting scaled GC contents were then added and divided by the combined read length for the entire assembly to give the GC content of that assembly.

2.4 Results and Discussion

2.4.1 Sanger Dideoxy Sequence

2.4.1.1 Calibration to a Single Genome Sequencing Project

The calibration of sequencing errors in *MetaSim* to experimentally-derived genome sequencing data was necessary to allow the detection of confident SNPs in metagenomic data. As a first step, *MetaSim* was calibrated to a single genome project. *E. coli* HS trace files (Rasko *et al.* 2008) were downloaded from the NCBI trace archive, assembled and simulated. Individual sequencing project data obtained from this archive usually contained plasmid libraries with two to seven different insert sizes. Each project's metadata file was used to divide the reads into their component groups of libraries. Each group contains one or more libraries from the same project that share the same mean insert size. These groups were then assembled separately and the statistics required to simulate the combined assembly were calculated. The statistics required were the number of mate-pairs, the proportion of reads that were mated, the mean and standard deviation of the observed insert lengths and the mean and standard deviation of the read lengths. This information was obtained from or derived from *Hawkeye*'s graphical user interface and the script *getlengths*.

Third party scripts were investigated as a method of automating the process of obtaining this information. However, these scripts generated unreliable data. The scripts *getlengths*, *astats* and *insert-sizes* are all from the *AMOS Assembler* package and all produce output in plain text. If the information was obtainable in plain text format, then these scripts could have been placed in a pipeline with in-house code.

Insert-sizes gave slightly higher insert lengths than *Hawkeye* regardless of which settings were used. *Astats* gave nonsensical output, for example, reporting that a group of libraries had more mates than reads. *Castats* from the *AMOS* package was also investigated and this gave mate-pair information that was different again. Given these errors and disagreements, *Hawkeye* was used instead of *astats*, *castats* or *insert-sizes*. *Hawkeye* is a more widely used program and since these programs are open source this means there have been more opportunities for *Hawkeye*'s users to detect and correct any software bugs. When a comparison was made between *insert-sizes* and *Hawkeye*, *MetaSim* produced simulations that better matched the original data.

MetaSim provides four parameters which together control the types and locations of

sequencing errors introduced into the simulated Sanger reads. These four parameters are: the proportion of deletion and insertion errors and the error rates at the first and last position in each read. *MetaSim* calculates the proportion of substitution errors from the first two proportions (Richter *et al.* 2009).

MetaSim was used to produce simulated reads for each group of libraries separately. These datasets were then pooled and assembled. This assembly was then compared to the assembly of the undivided sequencing project data. *MetaSim* parameters (including the four error parameters) were adjusted iteratively to make the two assemblies as similar as possible. All but one of the observed statistics for the simulation could be adjusted to within 1% of the values from a real sequencing project (Table 2.1). The proportion of insertions was adjusted to within 3%.

Table 2.1: Simulation of an *E. coli* assembly for error calibration.

	Real	***MetaSim***	**% Change**
Unmated Reads	24,130	24,168	0.16
Mated Reads	37,306	37,230	-0.20
Insert Length Mean	7281.93	7256.65	-0.35
Insert Length Standard Deviation	3055.29	3025.80	-0.97
Total Discrepancies	53,428	53,549	0.23
Discrepancies per kb	11.427	11.500	0.64
Proportion of Substitutions	0.406	0.405	-0.25
Proportion of Deletions	0.446	0.443	-0.67
Proportion of Insertions	0.148	0.152	2.70

As confirmation that the error rate for each type of error could be calibrated, a variational study was conducted for each *MetaSim* error parameter in isolation. Such a calibration would be dependent on finding an appropriate set of sequencing projects. A simple linear relationship between the proportion of insertion or deletion errors in a simulation, and proportion in the corresponding assembly was found (R^2 = 0.9999 and 1.0000, respectively) (Figure 2.3A and B). The other two error parameters “Error rate at start of read” and “Error rate at end of read” were kept in their original 1:2 ratio and varied as a pair. The relationship between the settings in *MetaSim* and the number of errors added (as reported by *MetaSim*) was linear (R^2 = 0.9999). When these two error rates were set at twice their defaults (0.02, 0.04), the extra errors were not directly detectable in the assembly. This is because they caused an increase in the number of

reads that were labelled as singletons and rejected, and a decrease in the amount of assembly. For all lower settings measured, the relationship between the error rate set and the error rate observed in the assembly was linear ($R^2 = 0.9997$).

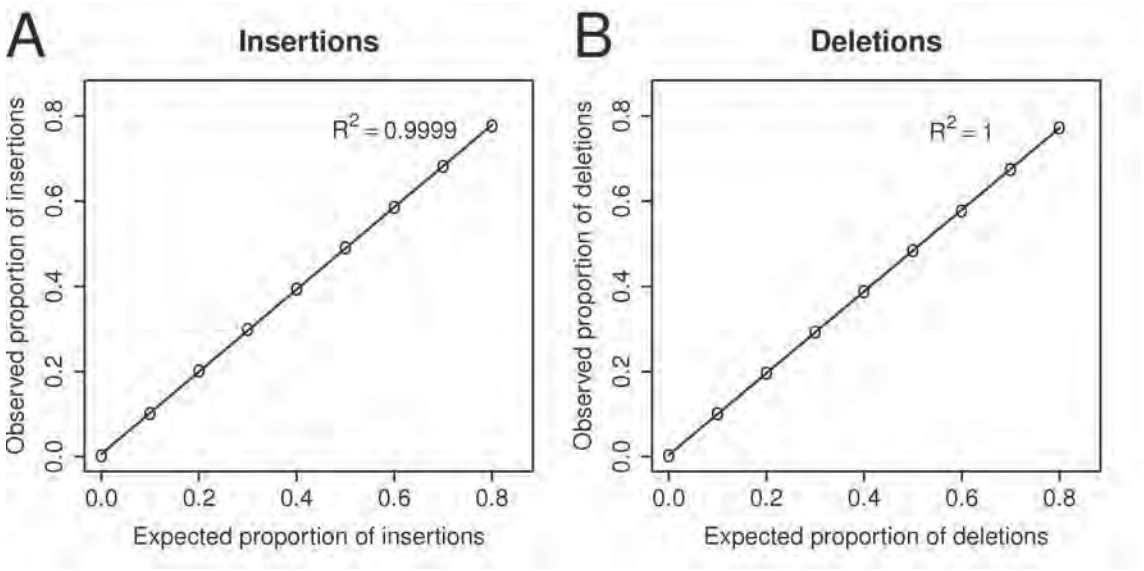

Figure 2.3: *MetaSim* error rates for Sanger reads can be accurately controlled.
Relationship between expected and observed levels of **A)** insertions and **B)** deletions in simulated *E. coli* Sanger Sequence

The numbers of discrepancies that resemble basic evidence for SNPs (i.e. at least two minority bases) in the *E. coli* assemblies were compared. This comparison found a large difference between the simulated and experimentally-derived assemblies with 145 and 1634 such discrepancies, respectively. *MetaSim* produces read errors at random unbiased locations (though it does allow for adjustment of error likelihood of the two ends of the reads separately). The pattern of locations of errors in experimentally-derived reads is more complex. For example, experimentally-derived reads have higher sequencing errors around homopolymer runs. This result suggested that *MetaSim* may not be suitable, if this could not be adjusted.

Two methods of dealing with this difference were considered. One method would be to improve *MetaSim*'s error model to be more realistic. The second option was to abandon *MetaSim* and to work solely with experimentally-derived reads. To inform a decision, the comparison of experimentally-derived and simulated data was repeated with more species.

2.4.1.2 Calibration to Multiple Sequencing Projects

To test the calibration of simulations, eight bacterial species were chosen, each from a different phylum. Phylum was used as a proxy for any genomic variable that might affect assembly, such as GC content or genome size. Some of these species could not be used due to various issues with the data and metadata and thus replacements were required. The original set chosen included: *Anabaena variabilis* ATCC 29413 (Copeland *et al.* 2005), *Bacillus anthracis Ames (Read et al. 2003b), Chlamydophila caviae* GPIC (Read *et al.* 2003a), *Deinococcus geothermalis* DSM 11300 (Copeland *et al.* 2006), *Hydrogenobaculum* sp. Y04AAS1 (Lucas *et al.* 2008a, Reysenbach *et al.* 2009), *Neisseria meningitidis* MC58 (Tettelin *et al.* 2000), *Thermotoga petrophila* RKU-1 (Zhaxybayeva *et al.* 2009) and *Treponema denticola* ATCC 35405 (Seshadri *et al.* 2004). Problems with data and metadata (outlined below) were found with *A. Variabilis, B. anthracis, T. petrophila and T. denticola.* The following potential replacements were also found to be problematic: *Chlamydia muridarum Nigg (Read et al. 2000), Sphingomonas wittichii* RW1 (Miller *et al.* 2010), *Staphylococcus aureus* subsp. *aureus* COL (Gill *et al.* 2005) and *Thermotoga* sp. RQ2 (Copeland *et al.* 2008a). Ultimately, *Borrelia burgdorferi* ZS7 (Fraser-Liggett *et al.* 2008), *Dictyoglomus turgidum* DSM 6724 (Lucas *et al.* 2008b), *Mycoplasma arthritidis* 158L3-1 (Dybvig *et al.* 2008) and *Prochlorococcus marinus* AS9601 (Kettler *et al.* 2007) were chosen as the final replacements.

Various problems with data, metadata and simulation led to genomes being replaced. *A. variabilis* had no information on insert length. This sequencing project also had reads which were recorded as having more than one mate. Both *B. anthracis* groups of libraries had no remaining mate-pairs after read trimming and culling by LUCY. *T. petrophila* had some insert lengths marked as zero. For the *T. denticola* traces, the XML metadata file had seven different insert sizes. Some of these groups of libraries (those marked 32 kb and 60 kb) did not have any of their reads listed in the mate-pair data. However, when all the groups were assembled together, four insert lengths of approximately these sizes were found (29 kb and 62 kb). Thus, the number of mate-pairs in these groups for the simulation was increased from zero to two. Unfortunately, *MetaSim* did not generate any reads for the *T. denticola* group marked 1200 bp. Downloading the mate-pair data for *S. wittichii* and *C. muridarum* was not possible. *S. aureus* and *Thermotoga* sp. RQ2 did not assemble.

Some issues could be resolved. For the *N. meningitidis* MC58 traces, 95% of the insert sizes were recorded as 300 or 700 bp in the XML file. This seems unlikely since the mean read length was 360 bp, and inserts should contain sequence from two reads and intervening sequence. The mean observed insert size in the assembly for these groups of libraries was 1652 bp. It would appear that the experimenters mistook the "insert length" field for "read length". The reads for the insert size group marked as having 8 kb inserts had an observed mean insert length of 861 bp in the assembly, albeit with a very small sample size. Due to this inconsistency and the single peak in the overall assembly insert size distribution, this assembly was simulated with a single group of libraries. The *C. caviae* simulation did not produce the expected number of reads; instead it produced two orders of magnitude less. This was explained by *MetaSim* designer Daniel Huson as resulting from long reads with high standard deviations on short inserts (personal communications). If the reads were too long for the inserts, they were not produced. Thus, the standard deviation for the read lengths in these simulations were both decreased to 150 bp (from 206.16 and 199.44 bp, respectively) to allow the production of the expected quantity of reads. Likewise, the simulation for *N. meningitidis* was producing too few reads. Therefore, the standard deviations for both insert length and read length were decreased from 322.49 and 141.29 to 200 and 80, respectively.

The simulation and comparison to data from real sequencing projects was performed for eight different datasets each containing a species from a different phylum (Table 2.2). There was a much greater variability in the number of errors per kb in the experimentally-derived data in comparison to the simulated data. These results were produced using the error settings calibrated to *E. coli* (Table 2.3).

Table 2.2: Simulation of eight assemblies for error comparison

Phylum	Species	Data Source	Discrepancies per kb		Total Bases
			Total	>2[a]	
Aquificales	*Hydrogenobaculum* sp. Y04AAS1	Real	28.497	1.695	2396541
		MetaSim	5.287	0.020	2167789
Chlamydiae	*C. caviae*	Real	41.755	1.255	1479123
		MetaSim	4.922	0.009	1506662
Cyanobacteria	*P. marinus*	Real	16.293	0.800	2745613
		MetaSim	5.671	0.009	2424352
Deinococcus-Thermus	*D. geothermalis*	Real	10.414	0.162	3939084
		MetaSim	5.281	0.011	3293592
Dictyoglomi	*D. turgidum*	Real	95.711	5.280	2228630
		MetaSim	5.598	0.010	2630906
Proteobacteria	*N. meningitidis*	Real	7.612	0.072	2693295
		MetaSim	5.959	0.018	2324094
Spirochaetes	*B. burgdorferi*	Real	32.706	3.260	1421068
		MetaSim	6.976	0.017	1496373
Tenericutes	*M. arthritidis*	Real	186.706	26.319	919297
		MetaSim	8.427	0.047	1334471

[a]Refers to discrepancies filtered so that only those with at least two matching minority bases are listed.

Table 2.3: Error settings calibrated to *E. coli*

Error rate at first position	6.432×10^{-4}
Error rate at last position	1.2768×10^{-3}
Proportion of Substitutions	0.406
Proportion of Deletions	0.446
Proportion of Insertions	0.148

To determine if this variability was linked to how the data were generated, the sequencing facility, date of sequencing, sequencing machine type and base-calling software were investigated for each data set. A limited amount of data was available (Table 2.4).

Table 2.4: Sequencing Project Metadata

Species	Facility	Load Date	Run Date	Date Submitted	Base-caller
B. burgdorferi	TCAG JCVI JTC	15th Jun 2007		17th Sep 2008 15th Oct 2008	*Jtrace* 3.10 *Tracetuner* 3.0
C. caviae	TIGR	28th Mar 2003		8th Apr 2002 29th Oct 2002	*Phred* 0.960108.C
D. geothermalis	JGI	29th Mar 2005		25th Apr 2006 9th May 2006	0.990772.G
D. turgidum	JGI	7th Oct 2008		4th Dec 2008 14th Dec 2008	*KB* 1.3.0
E. coli	JCVI	6th Oct 2005		13th Aug 2007 13th Sept 2007	*Phred*
Hydrogenobaculum sp. Y04AAS1	JGI	24th Aug 2007		7th Aug 2008	*Phred* 0.990772.G
M. arthritidis	TIGR	26th Mar 2003		1st Apr 2008 29th Jun 2008	*Phred* 0.990722.G
N. meningitidis	TIGR	15th Jun 2005		17th Mar 2000 19th Sept 2001 18th May 2005	*Phred*
P. marinus	JCVI	15th Nov 2007	5th July 2005	6th Nov 2006 22nd Jan 2007	*KB* 1.1.2 *Tracetuner* 2.0.1

Abbreviations: JCVI (J. Craig Venter Institute, USA) formerly TIGR (The Institute for Genomic Research, USA), JGI (Joint Genome Institute, US Department of Energy), TCAG (The Center for the Advancement of Genomics, USA). Load date and run date refer to NCBI trace archive. Date submitted refers to NCBI website.

Sequencing machine type was not stated for any of the eight projects. Since this was not available, the year the DNA was sequenced could have provided an indication of which machine was used. However, the actual sequencing date was only recorded for one species. For six out of eight genomes, the dates specified regarding when data was loaded into the NCBI trace archive are earlier than those provided on the NCBI website for when the data was submitted. For multiple projects the dates provided span over five years. The addition of new data to old sequencing projects can explain some of the variability in the dates for each project. The oldest date provided for each genome did not correlate well with the error rates in the assemblies of experimentally-derived data (Figure 2.4). Updates to older genomes could explain lower sequencing errors but they do not explain high sequencing errors. Whilst multiple database submission dates were available, this only provides a minimum age for the data. Legacy data may have been only recently uploaded. The base-calling software used by different projects was

recorded but multiple versions of various software programs were used in a variety of combinations. Thus, base-caller information was of limited use for determining the age of the assemblies. Even if the same machines and base-callers were used there could still be large differences in sequence quality due to cost-reducing variations in protocols.

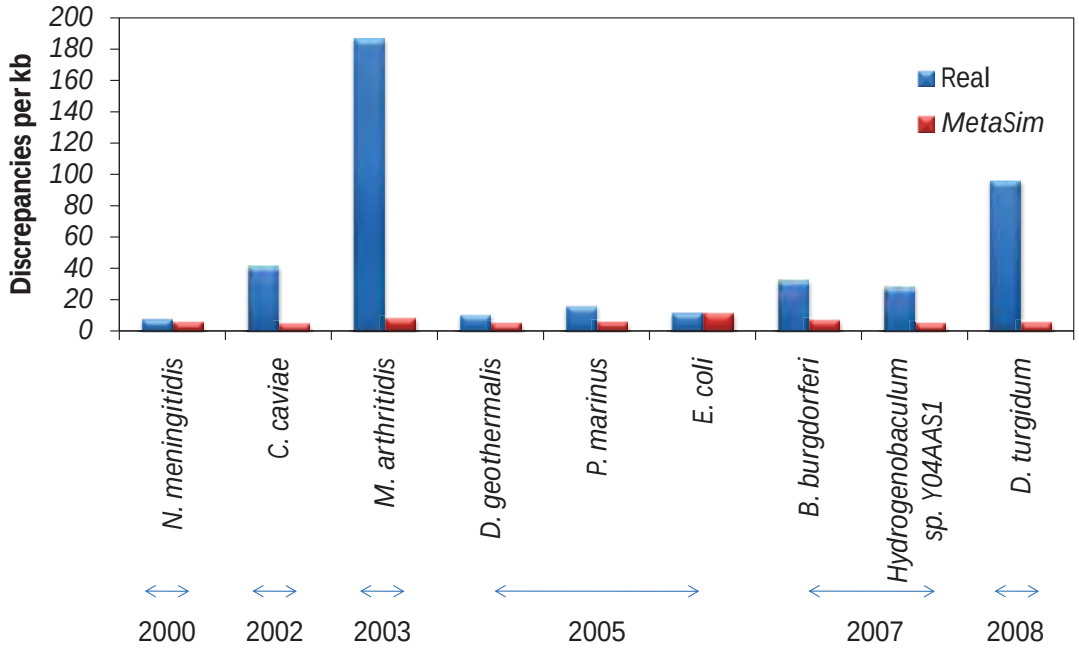

Figure 2.4: Variation in observed error rates in genome sequencing projects does not correlate with project age.
Observed error rates from experimentally-derived sequence data ("Real") are depicted in blue. Observed error rates in assemblies of simulated data ("*MetaSim*") are depicted in red. Sequencing projects are sorted, left to right, by the earliest date recorded for that project (run date, load date or submission date). Error rates in simulated reads are calibrated to *E. coli*.

2.4.1.3 Discussion

The use of *MetaSim* for simulating and understanding microheterogeneity in Sanger metagenomic datasets is still uncertain. Calibrating the settings requires a set of consistent sequencing projects, which was not available. It may have been possible to select a consistent set if sufficient metadata had been available. The sequencing institution from which the data originated could potentially be used to sort genome projects and obtain a consistent set. However, all eight sequencing projects analysed were sequenced by only two institutions (JGI and JCVI/TIGR), both of which produced projects with one of the highest and one of the lowest observed error rates. The sequencing machine used should have been a better discriminator of projects. The

machines used at each institution presumably change over time and more than one type of machine may be used at the same time at the same institution. The machine used for some of the projects should be available in the associated literature. However, many genome projects do not currently have any associated publications (eg. Copeland *et al.* 2006, Lucas *et al.* 2008b, Ward *et al.* 2010a). Extremes of GC content could affect the efficacy of Sanger sequencing and thus explain some of the large variability in sequencing error rates. Whilst the genomes with the highest error rates did have low GC contents (31% for *M. arthritidis* and 35% for *D. turgidum*), the genomes with the lowest (29% for *B. burgdorferi*) and highest (66% for *D. geothermalis*) GC contents had comparatively low error rates (Table A.1). Assembly coverage does explain a proportion of the variability in error rates. Coverage and sequencing error rate have a weak linear relationship (R^2 = 0.8257). However, this does not explain all of the variation in error rates. Even when normalised by coverage, there is still a large difference in error rates (from 1.88 to 11.21) (Table A.2).

Another problem is that the frequencies of more than two errors at a discrepancy are too low in the simulations and these are the ones that, in experimentally-derived data, provide better evidence for real SNPs (Table 2.2). Thus, it would have been desirable to make changes to *MetaSim*'s error model to make it more realistic. However, this would have required access to *MetaSim*'s source code, which was not available.

2.4.2 454 Pyrosequencing

The genomes of *Brucella abortus* NCTC 8038 (Ward *et al.* 2009), *Lactobacillus crispatus* CTV-05 (Ward *et al.* 2010a) and *Neisseria gonorrhoeae F62* (Ward *et al.* 2010b) were investigated using data produced on 454 GS FLX machines. These genomes were chosen because they were sequenced using the same technology at the same institute (Broad Institute of MIT (Massachusetts Institute of Technology) and Harvard, USA). Completed genome sequences were not available for these genomes. Therefore, the simulation of this data was produced from the scaffolds assembled by the *Celera Assembler*. The scaffolds longer than 3 kb from each species were concatenated together to approximate their respective genomes. The lengths of these pseudo-genomes were compared with the genome lengths of related strains to make sure that they were approximately the right size.

Since the parameters in *MetaSim* that control 454 simulation are so different from those used to control the Sanger simulations, only the total number of errors was calibrated instead of trying to fit the ratio of error types as well. The calibration process revealed that the parameter variously called “Proportionality Constant for Std. Deviation” or “Signal Std. Deviation Multiplier” (SSDM) was the main controlling parameter. To confirm this, SSDM and two other parameters were plotted, namely “Lognormal Distribution Mean”/“Mean Negative Flow Signal” (MNFS) and “Lognormal Distribution Std. Deviation”/“Std. Deviation for Negative Flow Signals” (SDNFS), to show that adjustments of these parameters had only small effects when the total errors added were appropriately low (Figure 2.5). The fourth error parameter “Scale Standard Deviation with Square Root of Mean” (SSDSRM) is a binary parameter which produces the least errors with its default setting.

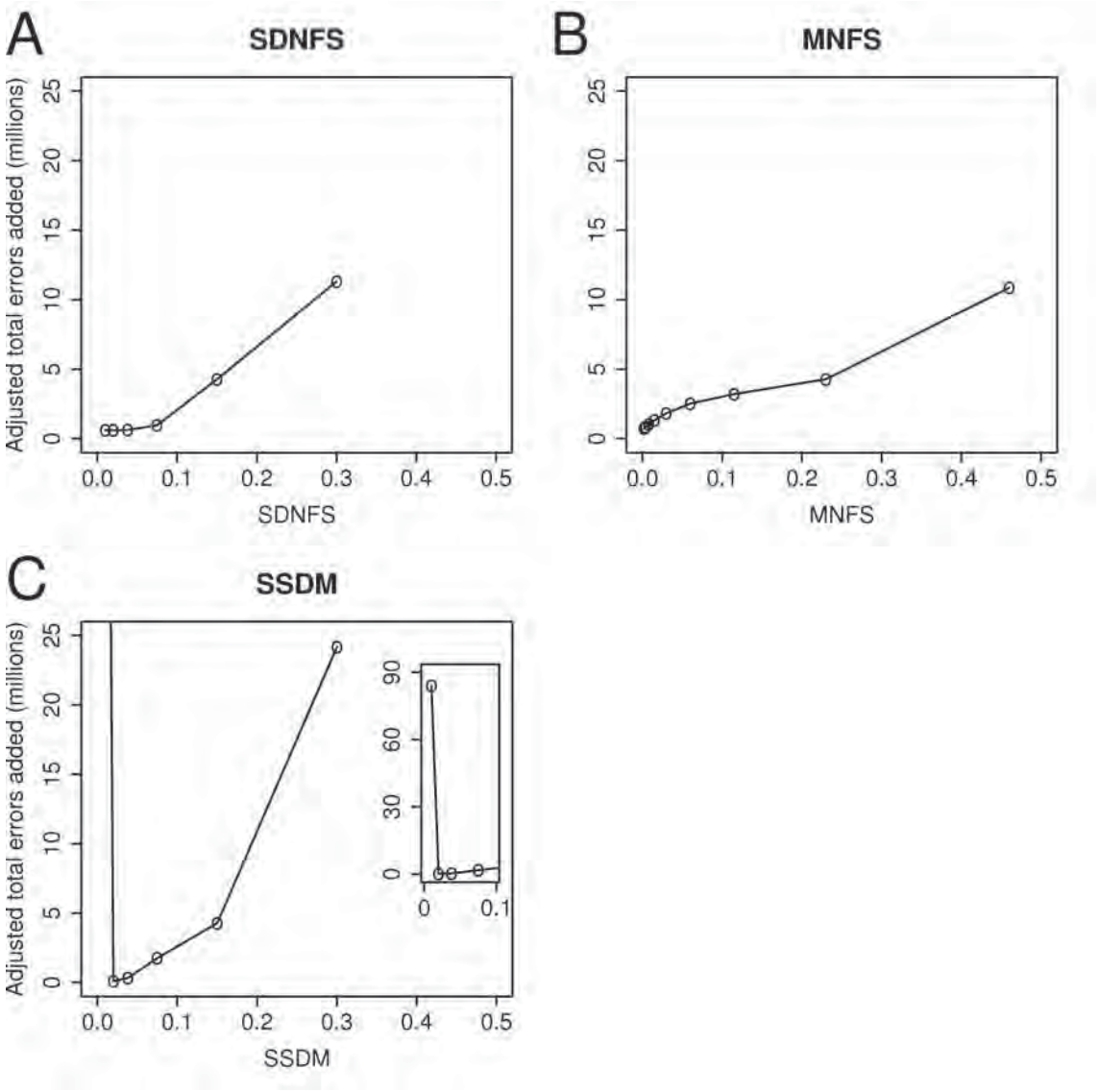

Figure 2.5: 454 error rates in *MetaSim* cannot be conveniently controlled.
Effect of varying: **A)** Signal Standard Deviation Multiplier (SDNFS), **B)** Mean Negative Flow Signal (MNFS) and **C)** Standard Deviation for Negative Flow Signals (SSDM) in *B. abortus*. For the value of the first data point in Subfigure C see insert.

The result of the variational study on the 454 parameters was that, when varied independently, the degree of effect of each parameter was comparable. A combination of these parameters (fit_1: MNFS = 0.05, SDNFS = 0.01 and SSDM = 0.1396) allowed the creation of assemblies that were calibrated to the corresponding genome sequencing projects (Figure 2.6). Since these calibrations only considered total number of errors, other combinations of these parameters could have been used. If setting the proportions of error types for 454 reads in *MetaSim* was to be pursued then all three parameters would need to be further investigated. Since the effect of the parameters is comparable they should be treated equally.

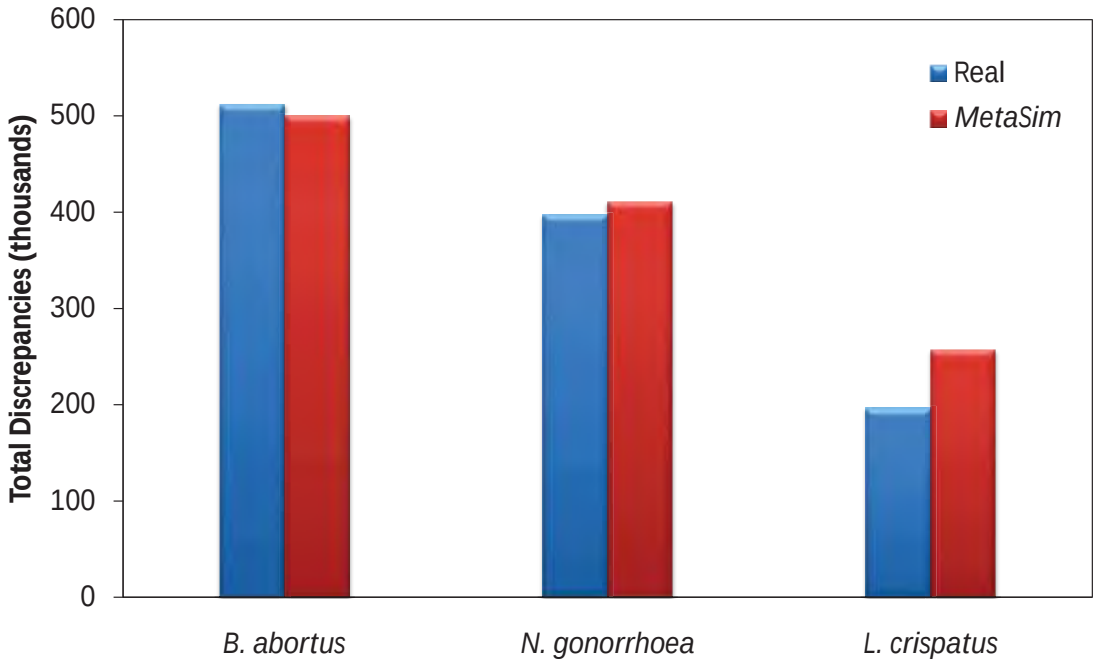

Figure 2.6: The total errors in *MetaSim* 454 data could be calibrated to experimentally derived data.
Total discrepancies in three assemblies of 454 reads from different species. Blue bars denote data from real genome sequencing projects ("Real"). Red bars denote simulated data ("*MetaSim*").

When SSDM was lowered to 0.01 and the other error parameters were kept at their defaults, the number of errors added rose to 84 million instead of decreasing (Figure 2.5C). This aberration appears to be due to a defect in *MetaSim* and thus outside of the usable range of the variable.

Despite this constraint on low values, SSDM is probably still the strongest controlling parameter. Measured in terms of the number of errors that *MetaSim* adds, it has the largest range of errors despite a smaller range of values being investigated. However, it is only marginally better and a future analysis should probably investigate the combined effects of all three parameters.

The relationships of all three parameters to the total number of errors produced are non-linear. The parameters provided in *MetaSim* for controlling 454 sequencing error are ambiguous and difficult to calibrate. The parameters for controlling sequencing error in Sanger reads were much clearer. The differences may allow more realistic simulations of the 454 technology. The non-linearity of the effect of parameters and their apparently complex combined effects add to this. This complexity is hinted at by the two data points investigated on how these parameters interact: the combination

arrived at by the above investigation (fit_1: MNFS = 0.05, SDNFS = 0.01 and SSDM = 0.1396) and previously calibrated settings (Lauro, unpublished work) (fit_2: MNFS = 0.1, SDNFS = 0.05 and SSDM = 0.05). The effect of lowering all three parameters by 50% to 67% from the default settings lowered the total errors to 954. This is much lower than the 500,378 for fit_1 where one of the parameters was decreased by a larger amount. An automated investigation of these parameters would probably be required to match the proportions of the different kinds of errors.

2.5 Conclusion

The calibration of an error model was problematic for both Sanger and 454 data. For the Sanger data, this was largely due to high variation in sequencing error rates in the available genome projects. This variation may have been dealt with if the necessary metadata on sequencing machines was available. Likewise, information on the date of sequencing may have helped but this was not explicitly stated. Multiple submission and upload dates were given, which may not correspond to the date of sequencing.

The 454 data available were more consistent and better documented. This allowed the calibration of the total number of discrepancies in each assembly. The individual calibration of insertion, substitution and deletion errors was not possible because of the issues involved in modelling 454 errors in *MetaSim*. This program uses different parameters for adjusting error rates in 454 and Sanger reads. For Sanger reads, these parameters are straight forward, have a simple relationship and linear effects on the errors introduced. For the 454 data, the parameters in *MetaSim* had non-linear relationships with the numbers of errors added. These parameters also had a complex relationship among themselves. Therefore, a direct search for a strong signal of inter-strain variation was performed.

An error model would have allowed a fine level of control over the FP (False Positive rate). Continuing without the model, the only way to control the FP was to look only at signal that was well above noise. A strong signal can be detected by looking for a pattern in the data that correlates with the desired information in simulations. Filtering that improves the readability of this patterned structure becomes a substitute for probability assignment and a probability cut-off. Chapter 3 describes the detection and filtering of such a signal.

Chapter 3

Detecting Chimeric Contigs

3.1 Summary

To help identify a strong signal of inter-strain variation, a method was developed for graphically representing data from a simulated metagenomic dataset. This involved filtering the data by contig length and selecting informative variables. The strain of origin of each read in each assembly was tracked to show a correlation between clusters in scatter plots of the data and contig chimerism. The effects of varying assembly parameters and coverage on the resultant scatter plots were investigated. Large quantities of reads were found to be unreported in some *Celera Assembler* contigs. This was corrected by using unitigs (high confidence contigs) as the basis of the analysis rather than contigs. A method of binning unitigs based on their chimerism was developed. Predictions could then be made about whether contigs in an assembly were clonal using logit regression and ROC (Receiver Operating Characteristic) plots.

3.2 Introduction

3.2.1 Chimerism and Contig Read Depths

For the purpose of this thesis, chimerism is defined as reads from multiple strains having been assembled together. Patterns in the read depths of contigs in an assembly contain information about the chimerism of each contig and the assembly as a whole.

In an assembly of reads from a clonal sample it would be expected that most contigs would have approximately the same read depth of N. However, there are regions in many genomes that are repetitive, which would create regions of higher read depth. Similarly, consider an assembly of two clonal strains of the same species, each sequenced to a read depth of N. This assembly should contain regions where the two strains were unique and thus there should be contigs with a read depth of N. There should also be regions that were common to both strains and these regions should have approximately twice that read depth, i.e. $2N$. Likewise, in an assembly of three such strains, there would be regions common to one, two and three strains. These regions should have read depths of N, $2N$ and $3N$, respectively. Such a combinatorial pattern could provide the strong signal necessary to determine the number of distinct strains in the microbial community that was sequenced. Alternatively, this pattern could be used to determine whether a species in a microbial community is clonal. Additionally, predictions could be made about whether individual contigs contain sequence from more than one strain.

3.2.2 Unitigs

In the *Celera assembler*, unitigs are a precursor of contigs. They can be considered as high confidence contigs and thus provide an alternative to contigs as a basis of analysis. A unitig of maximum length should either be unique sequence spanning up to repeat boundaries or almost the entire span of a genomic repeat. If a unitig is unique it should not have more than a read length of repeat on either end (Myers *et al.* 2000; http://sourceforge.net 2010, Fraley and Raftery 2006). Because a unitig can be a repeat, it can be classed as a surrogate (a collapsed repeat) and removed from the final assembly. This can occur even when the unitig has been placed into a contig. In such a

case the reads from that unitig are removed and the consensus sequence still spans the gap.

3.2.3 Logit Regression

Contigs, and clusters thereof, can be divided into two categories: clonal and chimeric. Logit Regression (a.k.a. logistic regression analysis) provides an optimal method for the analysis of binary dependent variables (Allison 1999). It uses training data to create a classifier for similar datasets. Thus, given appropriate training data, contigs can be classified according to whether they are clonal.

3.2.4 ROC Curves and AUCs

ROC plots allow classifiers, such as a logit regression of the chimerism of contigs, to be visualised and evaluated. Logit regressions assign scores to each data point along a continuum. This continuum is similar to the probability that a given data point belongs to a particular class. However, a score of 0.5 does not necessarily correlate to a probability of 0.5. Thus, an appropriate cut-off for dividing this continuum into two categories is required. ROC plots allow the optimum cut-off to be selected for a classifier by displaying how the false positive rate (FP) and true positive rate (TP) change as the cut-off value is varied.

AUCs (Area Under ROC Curve) allow different classifiers to be compared and for the best one to be selected. Thus, logit regressions using different training data but evaluating the same test data can be ranked using AUCs. A classifier's AUC value is equivalent to measuring its ability to rank a randomly chosen correct data point higher than a randomly chosen incorrect one (Fawcett 2004). Since an AUC value is a proportion of the area of a square its value is always between zero and one. A ROC curve produced by random guessing would be a diagonal line with an AUC of 0.5.

3.2.5 Aims

The aim in this chapter was to infer whether contigs from assemblies of simulated reads from the same species were clonal. To achieve this aim, informative variables were

chosen to allow the graphical analysis of microheterogeneity in these assemblies. The number of reads and discrepancies in each contig were used as a starting point for such variables. Due to unreported reads in contigs, unitigs became the basis of analysis. Once a strong relationship was shown between the chosen variables and contig chimerism, logit regression could be used to infer the chimerism of these contigs. ROC plots allowed the optimum cut-offs in the logit output to be selected.

3.3 Materials and Methods

3.3.1 Strains

Strains of *E. coli*, *N. meningitidis* and *S. aureus* were selected for use in simulated metagenomic assemblies. These three species are distantly related, had genomes from multiple strains available and have been well-studied. A standard set of strains was used for each of these species. Genomes were downloaded from NCBI by *MetaSim*. Since reads from the different strains from the same species were assembled together, knowledge about how similar these strains are was required. *dnadiff* (version 1.2), from the *MUMmer* (Maximal Unique Matches) package (http://mummer.sourceforge.net; Kurtz *et al.* 2004), was used to compare the genomes of strains within these species. Percentage alignments and identities were calculated for each pair of strains in a species. The alignment values reported for each pair of strains used the percentage of bases of one strain aligned to the other. Likewise, the reported identity values for each pair used the average one-to-one identity of one strain in reference to the other. For both alignment and identity values, the first strain alphabetically was used as the reference strain.

The percentage identity and percentage alignment of the strains from *E. coli* (Table 3.1 and Table 3.2), *N. meningitidis* (Table 3.3 and Table 3.4) and *S. aureus* were used to determine the similarity of the chosen strains. The percentage alignment of *S. aureus* COL (Gill *et al.* 2005) and *S. aureus* JH1 (Copeland *et al.* 2007) was 95.35. The percentage identity of these two strains was 98.92. The main requirement for strain selection was that all strains for a species had similar values for their identity and alignment. In practice, this meant keeping the range of alignment values under 12.5% and the range in identity values under 1.8%. Thus, *E. coli* CFT073 (Welch *et al.* 2002) was not chosen as it had the highest identity and alignment values to the other four *E. coli* strains. These values were 3.86% and 0.12% above the next highest, respectively. Discarding *E. coli* APEC O1 (Johnson *et al.* 2007), the strain with the lowest identity and alignment values, would not have decreased the range values considerably. In this case, the respective decreases in range would only have been 1.4% and 0.01%.

Table 3.1: Percentage alignments of five strains of *E.coli*

E. coli	536	55989	APEC O1	ATCC 8739
55989	83.21			
APEC O1	91.13	81.35		
ATCC 8739	80.08	84.77	78.68	
CFT073	94.81	81.11	91.06	84.69

Table 3.2: Percentage identities of five strains of *E.coli*

E. coli	536	55989	APEC O1	ATCC 8739
55989	97.14			
APEC O1	98.90	97.13		
ATCC 8739	97.18	98.62	97.17	
CFT073	98.98	97.15	99.02	97.17

Table 3.3: Percentage alignments of four strains of *N. meningitidis*

N. meningitidis	053442	FAM18	MC58
FAM18	94.38		
MC58	95.36	94.19	
Z2491	95.51	93.48	92.5

Table 3.4: Percentage identities of four strains of *N. meningitidis*

N. meningitidis	053442	FAM18	MC58
FAM18	97.43		
MC58	97.17	97.23	
Z2491	97.32	97.32	97.13

Assemblies of different quantities of strains were produced to find relationships between strain count and the way contig data-points cluster in scatter plots. For *E. coli*, the two-strain assemblies used the strains *E. coli* APEC O1 and *E. coli* ATCC 8739 (Copeland *et al.* 2008b). Three-strain assemblies also included *E. coli* 55989 (Touchon *et al.* 2009). Four-strain assemblies additionally included *E. coli* 536 (Brzuszkiewicz *et al.* 2006). For *N. meningitidis*, the two-strain assembly used *N. meningitidis* 053442 (Peng *et al.* 2008) and *N. meningitidis* FAM18 (Bentley *et al.* 2007). The three-strain assembly also included *N. meningitidis* MC58 (Tettelin *et al.* 2000). The four-strain additionally included *N. meningitidis* Z2491 (Parkhill *et al.* 2000). *S. aureus* assemblies used *S. aureus* COL and *S. aureus* JH1.

3.3.2 *MetaSim*

For all assemblies of simulated reads, *MetaSim* produced 454 reads calibrated to the real genome sequencing projects that were used to calibrate *MetaSim*'s error parameters for 454 reads in Chapter 2, namely *B. abortus*, *L. crispatus* and *N. gonorrhoeae*. The error calibration set MNFS to 0.05, SDNFS to 0.01, SSDM to 0.1396 and SSDSRM to true. The number of simulated reads per strain used in assemblies was 433,590, the average number of reads in the three genome projects. This allowed the creation of multi-strain assemblies with high, yet technically feasible, coverage. *MetaSim* was set to create reads with an expected read length of 258 bp (101 flow cycles). This number was calculated by averaging the read length means of the three sequencing projects and rounding to the nearest flow cycle. *MetaSim*'s other parameters were kept at their default settings.

3.3.3 Assembly

The *Celera assembler* has a default unitigger error rate of 1.5%. In this thesis, this rate was standardised to a higher value of 4% to increase the degree of assembly of reads from different strains. Ideally, this would be set so that contigs with all possible combinations of strains would be present in equal proportions. Read overlaps above the unitigger error rate are not used to construct unitigs. The assembler's overlapper error rate and consensus error rate both have a default setting of 6%. The unitigger error rate cannot be set higher than either of these rates. Overlaps with an error rate above the overlapper error rate are not computed. The consensus error rate describes the approximate error rate in alignments expected by the consensus stage of the assembler. When the unitigger error rate was set above 6%, the overlapper error rate and consensus error rate were also increased to the same value.

A separate FRG file was produced for each strain and these were then concatenated. This required use of the custom *Python* script *combine_FASTA.py* to ensure that the read identifiers remained unique. A new version of *run_pipeline.py* (*run_pipeline_beta.py*) with extra options was created to facilitate this. The creation of two FRG files for the one assembly allowed the tracking of the number of reads from each strain in each contig. Extra functionality was added to *SNP_info.py* to calculate statistics, including source strain proportions, on every contig in each assembly.

3.3.4 Unitigs and Contigs

Strong relationships were found between the chosen variables, reads per unit of contig length ($\bar{R}$) and discrepancies per unit of contig length ($\bar{D}$), and the chimerism of contigs. However, some contigs had a very large proportion of their reads unreported in the assembler output. This is because the *Celera assembler* designates unitigs with uncharacteristically large read depths as surrogates because they are likely to be stacked repeats. Surrogates are used to extend the consensus sequence in contigs but their reads are not included in the contig. However, in metagenomic assemblies, such areas are often due to a species in high abundance or a section of a genome that was very similar across multiple strains. Therefore, assemblies were analysed with the -utg setting in *toAmos*, so that the list of contigs in the *AMOS* bank used during SNP analysis was replaced by unitigs. The custom script *SNP_info_unitig.py* was created to replace contigs with unitigs as the basic unit of analysis. Total reads per unit of unitig length ($\bar{R}'$) and total discrepancies per unit of unitig length ($\bar{D}'$) were used instead of $\bar{R}$ and $\bar{D}$.

3.3.5 Unitig Binning

The creation of training data required the classification of that data into discreet groups. Unitigs were binned based on the number of strains that contributed a substantial number of reads to that unitig. The number of bins is equal to the number of strains. A method for binning these unitigs that was tolerant to some variation in the quantity and source of reads was developed. This method took into account the proportions of reads from each contributing strain and not just the number of contributing strains. It assigned a number to each unitig along a continuum which was then brought back into discrete bins by the use of cut-off values. The equation that governed bin assignment is:

$$S = n + f(p_1, n) + f(p_2, n) + \ldots + f(p_N, n) \qquad \textbf{(3.1)}$$

Where S estimates the number of strains that contributed a significant proportion of reads to a unitig and how close to equal proportions those strains are; n is the number of strains that contributed at least one read to that unitig and N is the total number of

strains in the assembly; p_1, p_2, ... and p_N are the percentages of reads from strains s_1, s_2, ... and s_N in the unitig; where n and $N \in \mathbb{Z}^+$; S, p_1, p_2 and $p_N \in \mathbb{R}$; and

$$\boldsymbol{f(x,n) = \begin{cases} nx - 1, & 0 < x < \frac{1}{n} \\ 0, & otherwise \end{cases}} \quad \textbf{(3.2)}$$

If a unitig from a three-strain assembly had 50 reads from one strain and 50 from another, it would be correctly assigned the number two ($2 + 0 + 0 + 0 = 2$). A similar unitig with 50 reads from one strain, 49 from another and one from a third would be assigned: $3 + 0 + 0 + \left(3 \times \frac{1}{50+49+1} - 1\right) = 2.03$. Thus, this method assigned similar numbers to similar unitigs and consequently allowed them to be binned together by using an appropriate cut-off value. It was implemented using a series of custom *R* (version 2.12.1) functions.

For dichotomous predictions, a simplified version of the above formula was used. All unitigs not classified as clonal were classified as chimeric. Unitigs with less than 75% of reads from one strain were classified as chimeric and those with greater than 75% as clonal.

3.3.6 Normalisation of Coverage

To test whether plots from different species could be made more similar, assemblies were made with the number of reads used for each strain normalised by the length of each genome. The lengths of plasmids were not included as this sequence was not used by *MetaSim*. The average coverage for each strain was calculated as: number of reads times read length divided by genome length. The mean read lengths for the reads *MetaSim* produced for each strain varied within a small range of values. Thus, the exact mean value for each strain was used in the coverage calculation for that strain. The combined two-strain coverage for *S. aureus* and *N. meningitidis* were adjusted to match the coverage of *E. coli*, namely 46.605×.

3.3.7 Logit Regression and ROC Plots

Logit regression was performed in *R* to make predictions about whether unitigs were clonal. The *ROCR* package in *R* was used to create ROC plots (version 2.12.1). The formula $Cost = FP + (1 - TP)$ was used to calculate the cut-off value for the logit regression output (adapted from Provost and Fawcett 2001). This was equivalent to choosing the point on the ROC curve closest to the top left corner. $Cost$ was minimised in order to minimise the proportion of possible type one and type two errors.

3.4 Results and Discussion

3.4.1 Choice of Variables

The expected combinatorial pattern in the read depths of contigs indicates that the number of reads in each contig is a potential indicator of inter-strain variation. In a mixed system, a contig that has reads from multiple strains should have more reads than similar clonal contigs. Likewise, if reads from different strains were assembled together, the probability of discrepancies would be higher than if the reads were from the same strain. Additionally, adding extra reads in a contig, even if they were sequenced from identical regions of DNA, would add extra discrepancies due to sequencing errors. Thus, the number of discrepancies in each contig is also a potential indicator of inter-strain variation. Therefore, the number of reads and discrepancies in each contig were used as the basis of informative variables, which were in turn used for a graphical analysis of chimerism.

The total number of reads versus the total number of discrepancies for each contig was not sufficiently informative. For example, a three-strain assembly of *E. coli* with a 7% unitigger error rate (three-strain 7% *E. coli* assembly) has a single cluster longitudinally dispersed along y = x (Figure 3.1).

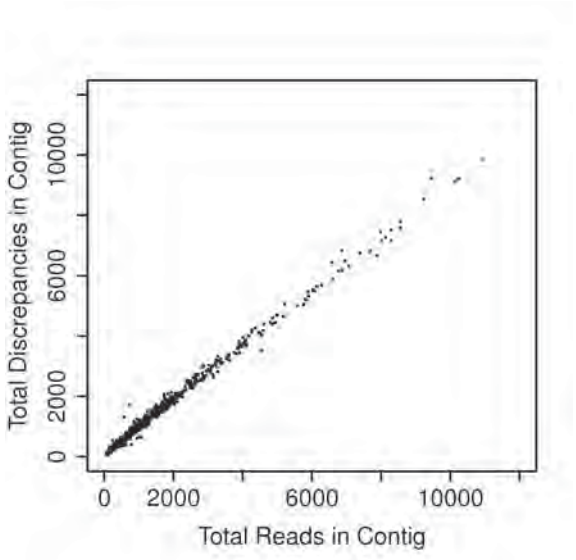

Figure 3.1: Read and discrepancy counts are not sufficiently informative variables.
Three-strain 7% *E. coli* assembly, unfiltered and unnormalised.

The number of reads in a contig is highly dependent on the length of that contig. Ten kilobase contigs almost always have more reads than 100 bp contigs. The number

of discrepancies in a contig is also highly dependent on length. Thus, the numbers of reads and discrepancies in each contig were normalised by dividing by the contig length. These plots were more informative for extracting information about contig chimerism. The equivalent scatter plot of the 7% *E. coli* assembly had two poorly separated clusters with different shapes surrounded by many scattered observations (Figure 3.2).

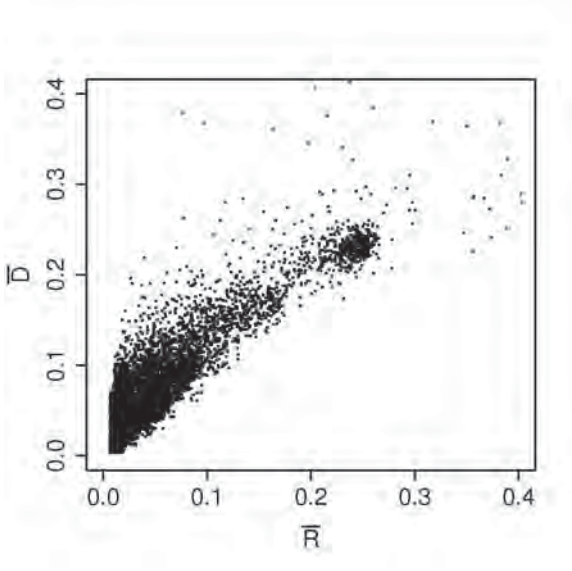

Figure 3.2: Normalising read and discrepancy counts make these variables more informative.
Three-strain 7% *E. coli* assembly, length filtered and unnormalised.
$\bar{R}$: reads per unit of contig length. $\bar{D}$: discrepancies per unit of contig length.

A further improvement of the chosen variables for detecting chimerism was to filter out the shortest contigs, i.e. those less than 1 kb. $\bar{R}$ and $\bar{D}$ are both statistical in nature. Both refer to an average value over the length of that contig. This means that contigs with only a few reads had a small sample size for these values and therefore their derived parameters were less reliable. The filtered version of the 7% *E. coli* assembly had three small well-separated elliptical clusters (Figure 3.3).

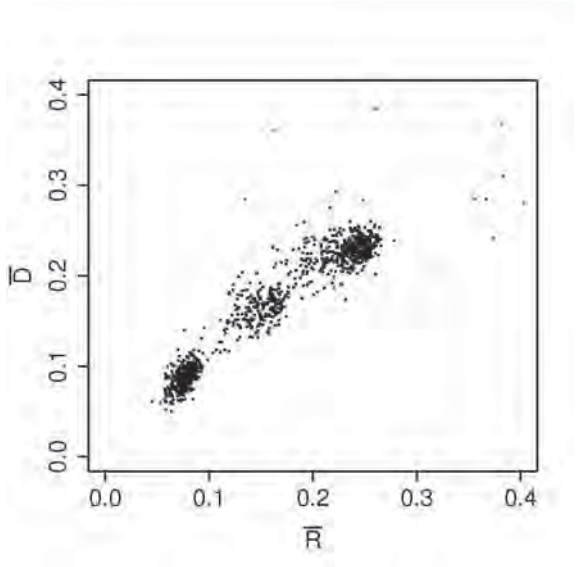

Figure 3.3: Length filtering of $\bar{R}$ and $\bar{D}$ allows clear evenly spaced clusters to be seen.
Three-strain 7% *E. coli* assembly, length filtered and normalised.
$\bar{R}$: reads per unit of contig length. $\bar{D}$: discrepancies per unit of contig length.

A correlation matrix was used to confirm this choice of variables. Since both variables were being used to measure chimerism, at least a moderate correlation was expected. If the correlation was very high then there would be no extra information added by using a second variable, although it would still provide a confirmation. A moderately high correlation of 87% was found (Figure 3.4).

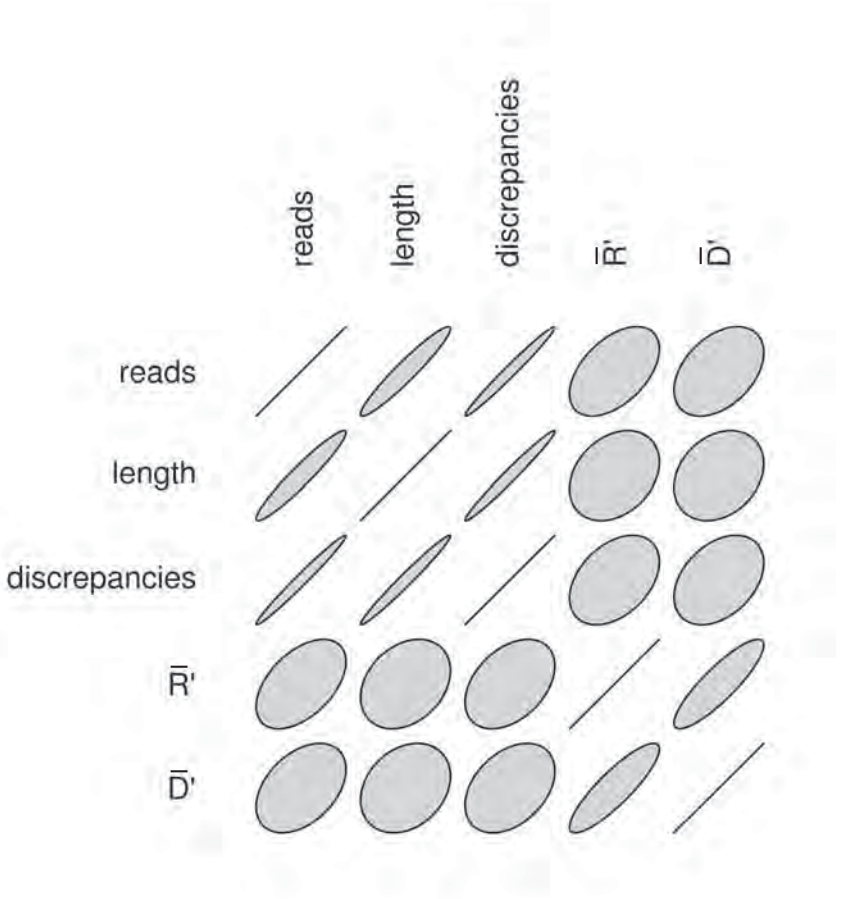

Figure 3.4: $\bar{R}'$ and $\bar{D}'$ are appropriate variables, since have a moderately high correlation.
Correlation matrix of variables in a two-strain 8% assembly of *N. meningitidis*.
$\bar{R}'$: reads per unit of unitig length. $\bar{D}'$: discrepancies per unit of unitig length.

3.4.2 Read Tracking

The systematic naming of reads in *MetaSim* made the tracking of reads from different strains possible. This in turn allowed the creation of the script *is_chimeric.py*. This script counted the numbers of reads from each strain in each contig and thus assigned a percentage value to each contig. This value recorded what proportion of a contig's reads come from the first strain and therefore how chimeric it is. With the use of this script, every contig on a scatter plot could be coloured according to this percentage. Contigs in a two-strain *E. coli* assembly were divided into groups based on this percentage value. The extra information on chimerism correlated very closely with the two clusters (Figure 3.5).

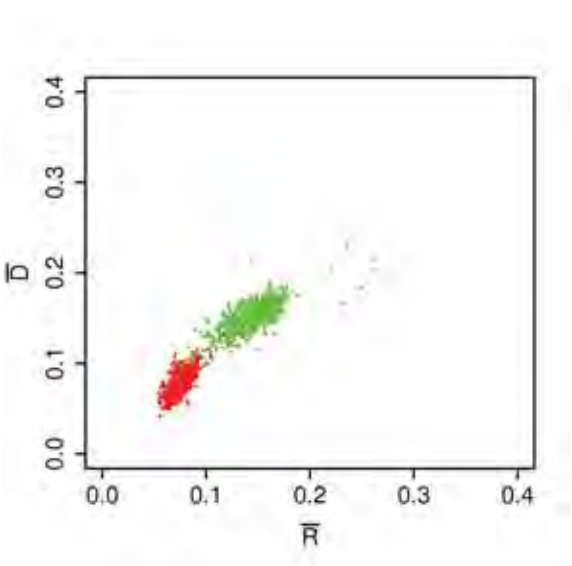

Figure 3.5: Tracking reads allows clusters to be coloured by strain count.
Coloured two-strain *E. coli* assembly. Red contigs: **S** < 1.5. Green contigs: **S** > 1.5.
$\bar{\boldsymbol{R}}$: reads per unit of contig length. $\bar{\boldsymbol{D}}$: discrepancies per unit of contig length.

This colouring method shows that the filtering and normalisation applied to $\bar{R}$ and $\bar{D}$ results in more informative scatter plots (Figure 3.6). The correlation between strain count bins and clusters in Figure 3.5 and Figure 3.6C shows that the binning has worked well.

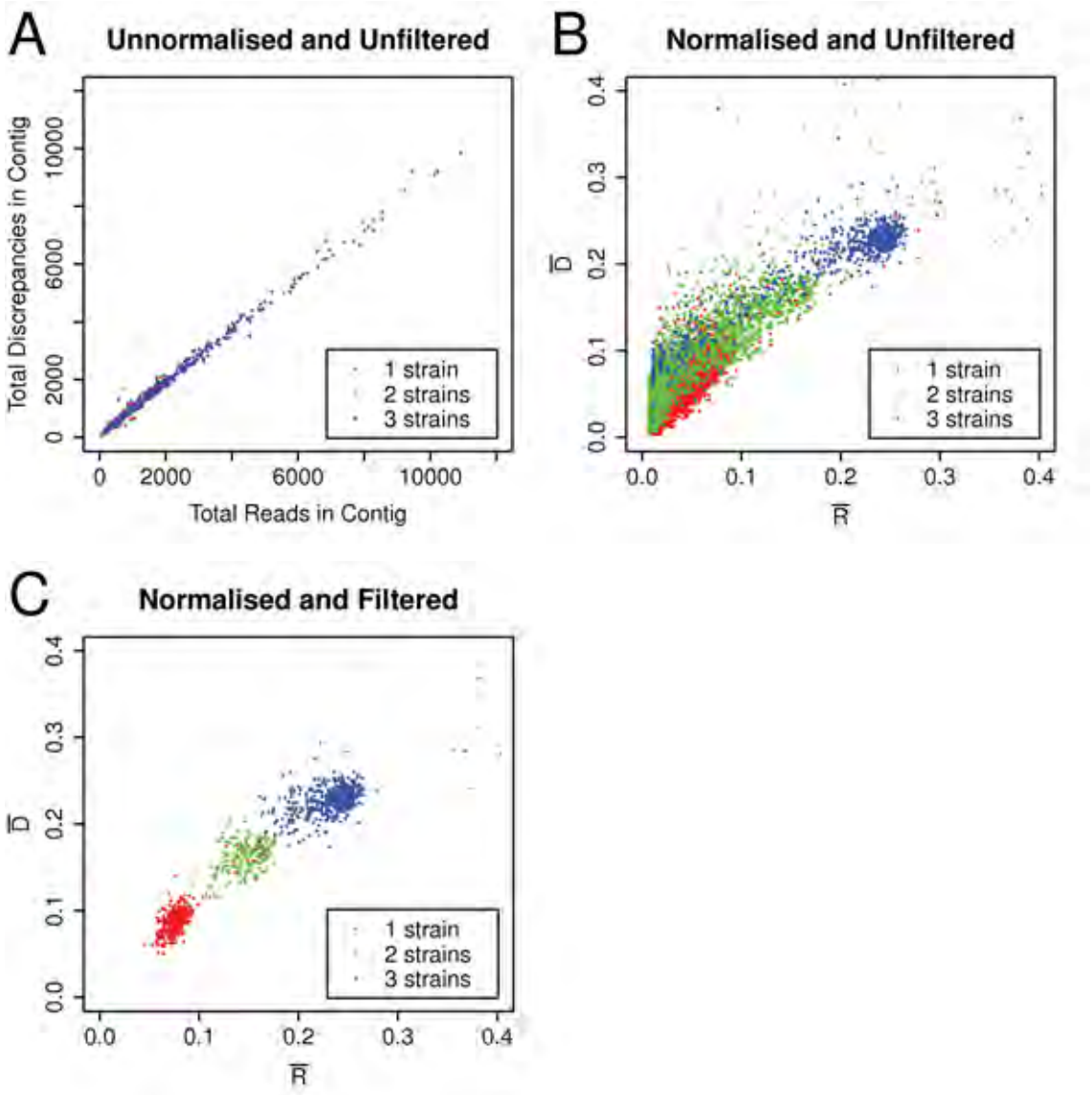

Figure 3.6: The chosen variables create clusters that correlate well with strain count.
Coloured three-strain 7% *E. coli* assembly. Normalisation of variables is by length. Filtering removed contigs < 1 kb in length.
Red contigs: $\boldsymbol{S} < 1.5$. Green contigs: $1.5 < \boldsymbol{S} < 2.5$. Blue contigs: $\boldsymbol{S} > 2.5$.
$\overline{\boldsymbol{R}}$: reads per unit of contig length. $\overline{\boldsymbol{D}}$: discrepancies per unit of contig length.

3.4.3 Understanding Cluster Locations

As a control, a one-strain (*E. coli* APEC O1) assembly was produced. This assembly contained one distinct cluster, as expected (Figure 3.7A). This assembly was compared to equivalent two- and three-strain assemblies (Figure 3.7B and C). The bottom-left cluster, that denotes non-chimeric contigs, was very similarly placed in all three scatter plots. Its size was also very similar between the two- and three-strain assemblies. The one-strain assembly had the least reads and therefore had less contigs and a smaller

cluster. The top-right cluster in the two-strain assembly has a corresponding cluster in the three-strain plot, both of which denote a roughly even mixture of two strains.

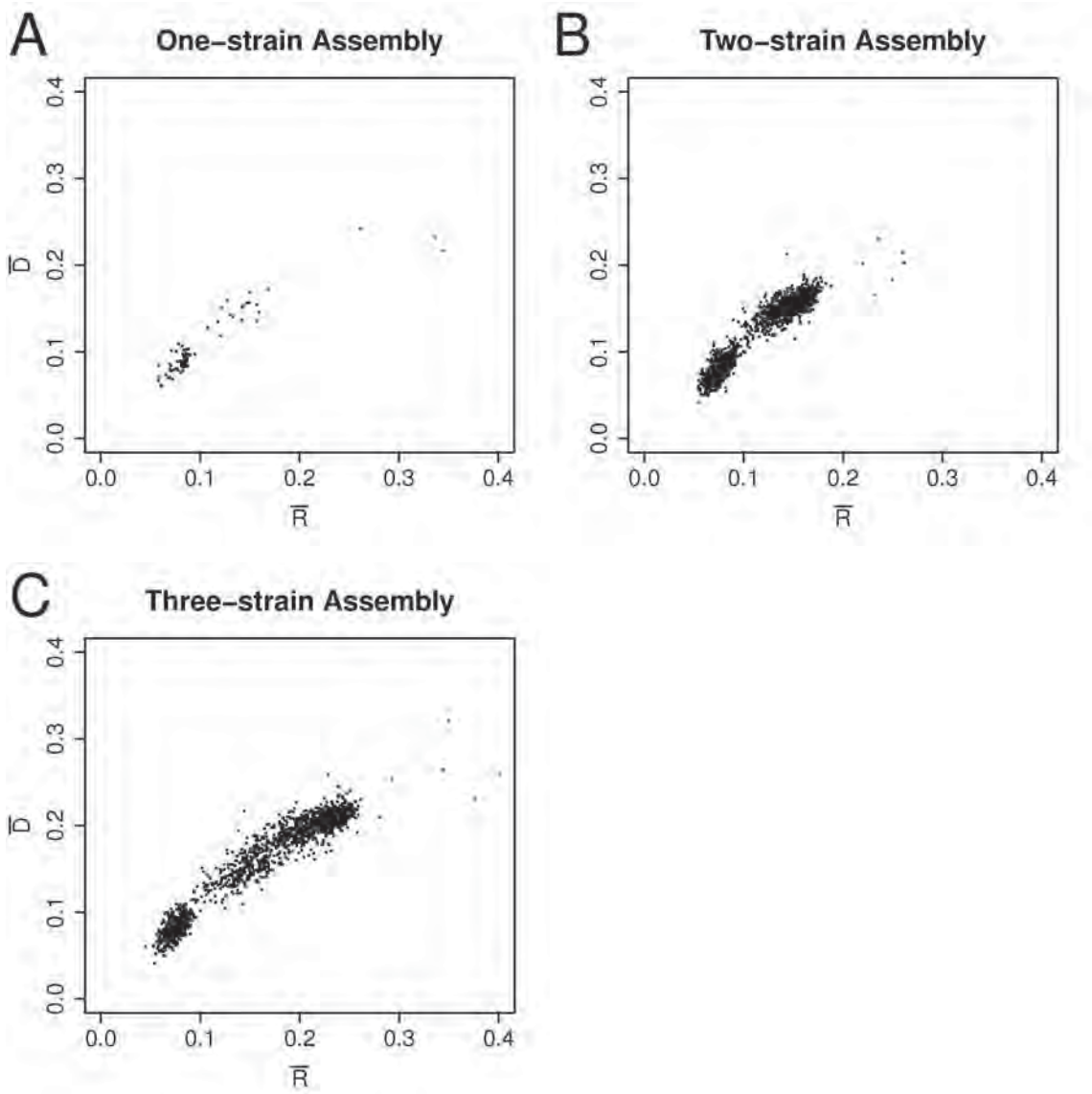

Figure 3.7: The size and positions of clusters with the same strain count are similar across assemblies with different strain counts.
E. coli assemblies with different strain quantities.
$\bar{R}$: reads per unit of contig length. $\bar{D}$: discrepancies per unit of contig length.

In each of these three assemblies, there are contigs with higher $\bar{R}$ values than any of the distinct clusters. In the one-strain assembly, these contigs look like a weak two-strain cluster and in the two-strain assembly they look like an even weaker three-strain cluster. In the one-strain assembly, there are even a few points located where a three- or four-strain cluster would be located. Many genomes contain highly conserved regions that have similar sequence to other parts of that genome. These regions can be

erroneously assembled together leading to unitigs of greater read depth. If the sequence of a clonal contig can be found with sufficient similarity in two locations in a genome then it should have twice the read depth of other clonal contigs. Likewise, if that sequence is located in three or four locations then it should have three or four times the read depth. Thus, a small number of clonal contigs can imitate the location on a scatter plot of chimeric contigs. It is also possible for chimeric contigs to have increased read depth due to this effect, if they contain sequence that is conserved in at least one of their constituent strains. In the two-strain plot (Figure 3.7B), four of the six contigs with higher $\bar{R}$ values than the two-strain cluster are clonal, and two are chimeric. In the three-strain assembly (Figure 3.7C), two out of six are clonal, two are two-strain unitigs and two are three-strain.

Two-strain and three-strain assemblies of *N. meningitidis* were produced to determine if the trend of one cluster per strain in assemblies of *E. coli* held in other species. The same correspondence of clustering was observed between these assemblies, though the boundary between the two- and three-strain clusters in the three-strain assembly was not clearly defined (Figure 3.8). These clusters have higher $\bar{R}$ values since the same number of reads was used with a smaller genome. The *E. coli* strains have a mean genome size of 4.981 Mb compared to 2.201 Mb for the *N. meningitidis* strains.

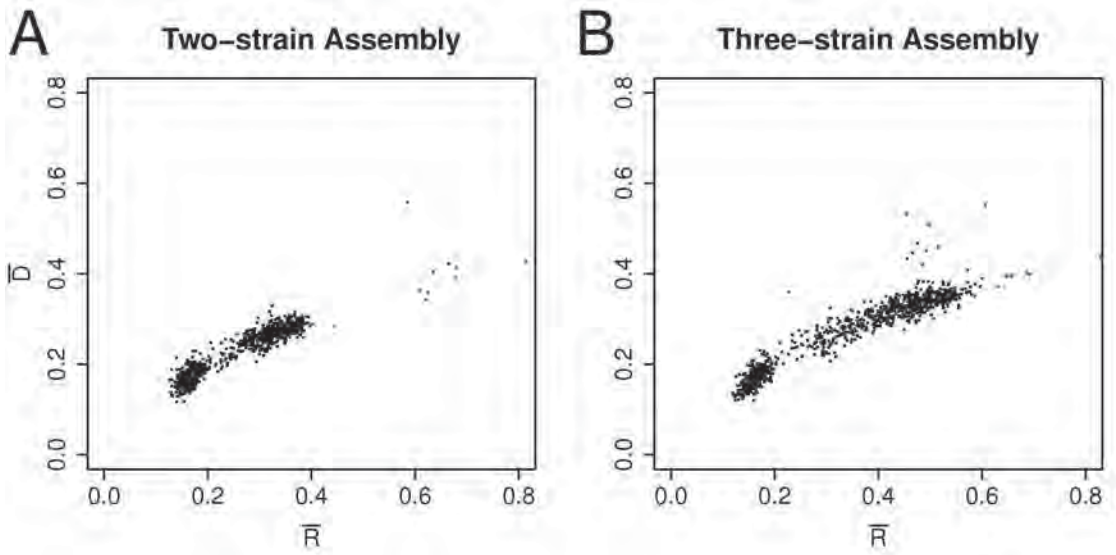

Figure 3.8: The same clustering is apparent in assemblies of a distantly related species.
N. meningitidis assemblies with different strain quantities.
$\bar{R}$: reads per unit of contig length. $\bar{D}$: discrepancies per unit of contig length.

These two assemblies were reproduced with a higher 8% unitigger error rate which

made the cluster boundaries more distinct (Figure 3.9). The reproduced assemblies used unitigs rather than contigs as the scatter plot data points, as explained in the next section. The use of unitigs is denoted by the symbols $\overline{R}'$ and $\overline{D}'$.

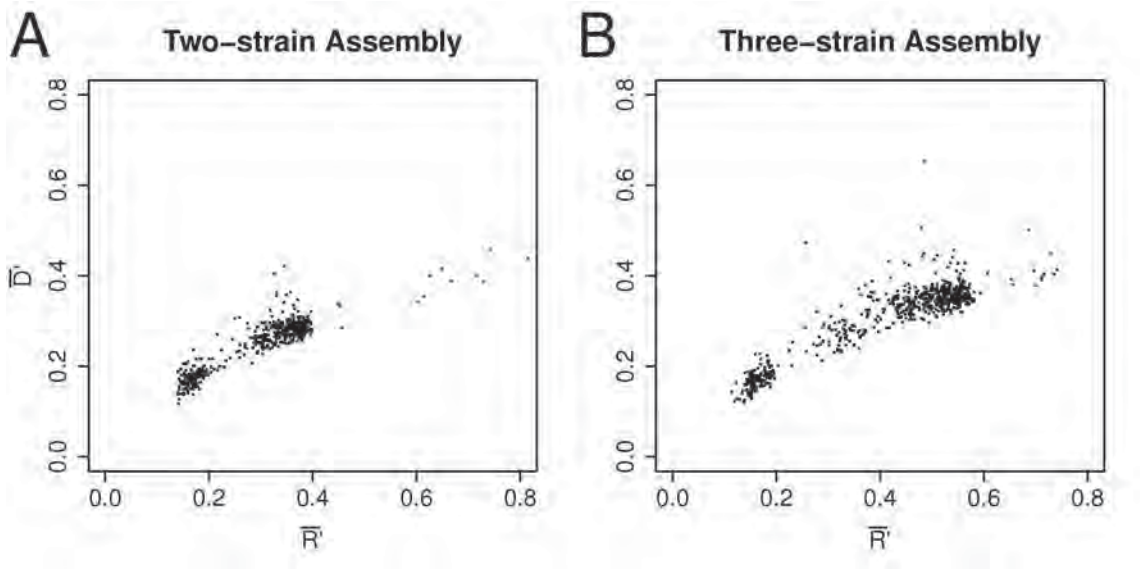

Figure 3.9: Clustering of *N. meningitidis* assemblies is more apparent at a higher unitigger error rate.
N. meningitidis 8% assemblies with different strain quantities:
$\overline{R}'$: reads per unit of unitig length. $\overline{D}'$: discrepancies per unit of unitig length.

3.4.3.4 Understanding the Effect of Coverage on Cluster Locations

In *S. aureus* two-strain assemblies, the number of reads used per strain was varied from 10% to 100% of the standard number in 10% intervals. As the number of reads decreased, the clusters moved closer together and closer to the origin (Figure 3.10). For the 10% reads assembly, the clusters were so close that further magnification was required to distinguish them (Figure 3.10A). The equivalent spacing of clusters across assemblies of different coverage showed that the chosen variables possessed a robust relationship across this coverage range.

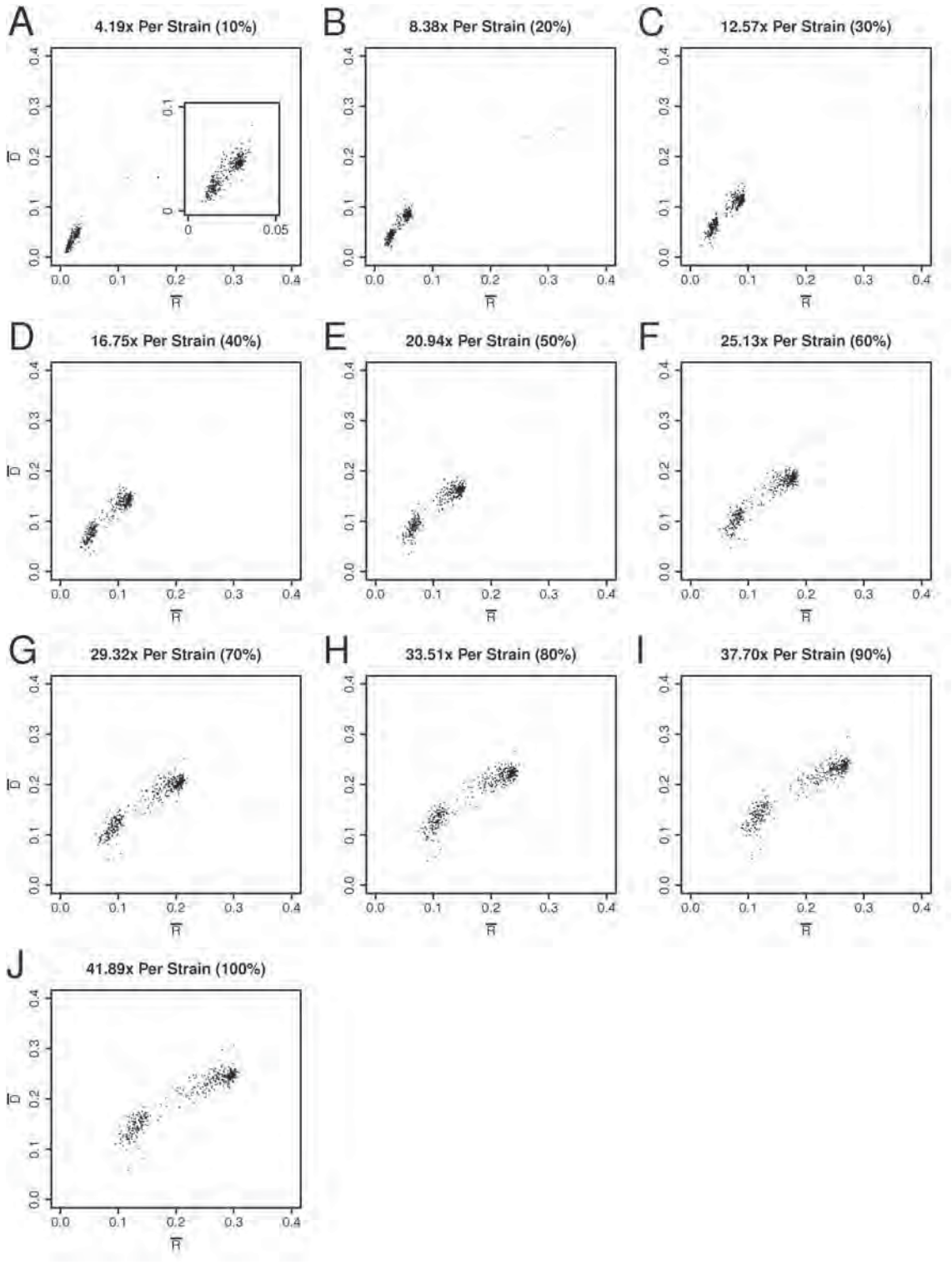

Figure 3.10: Clustering is independent of coverage over a wide range of coverage.
S. aureus two-strain assemblies with varying coverage: 10% (4.19× per strain) coverage to 100% (41.89× per strain) coverage.
$\overline{R}$: reads per unit of contig length. $\overline{D}$: discrepancies per unit of contig length.

3.4.3.5 Understanding the Effect of Assembly Tolerances on Clusters

The three-strain assembly of *E. coli* was repeated using unitigger error rates in the *Celera assembler* between 1% and 8% (Figure 3.11). Scatter plots were produced for these assemblies. As the unitigger error rate was increased, the number of clusters increased. The 1% assembly plot (Figure 3.11A) displays a strong clonal cluster, a less dense two-strain cluster and almost no obvious three-strain unitigs. The 2% plot shows the second cluster elongating and, in the 3% plot, the second cluster starts to resolve into two clusters (Figure 3.11C). In the 5% plot, the two-strain cluster becomes obvious and it is well resolved in the 6% plot (Figure 3.11E and F). After this, the second cluster decreases in density in the 7% and 8% plots (Figure 3.11G and H). While this clustering was dependent on an assembly parameter, since this is part of the analysis, it should be possible to use the most appropriate settings in all situations. For this assembly, the plots showed 7% as the optimal error rate. When contour plots were used, 6% was the setting where there were three peaks with the most similar heights (Figure B.1). However, the setting of 4% was still used as the standard setting for consistency.

As the unitigger error rate is increased, the density of the three-strain cluster increases and the one- and two-strain clusters decrease in density. This does not apply to the two-strain cluster at very low tolerances. At a rate of 4% or lower, the two- and three-strain clusters are still forming. At higher rates, the positions of the clusters are relatively stable. At these rates, the main effect of varying the tolerances is the changes in density of these clusters. This is expected, as reads that are unique to one or two species to a particular level of similarity can be considered common to all three strains, when a lower standard of similarity is used.

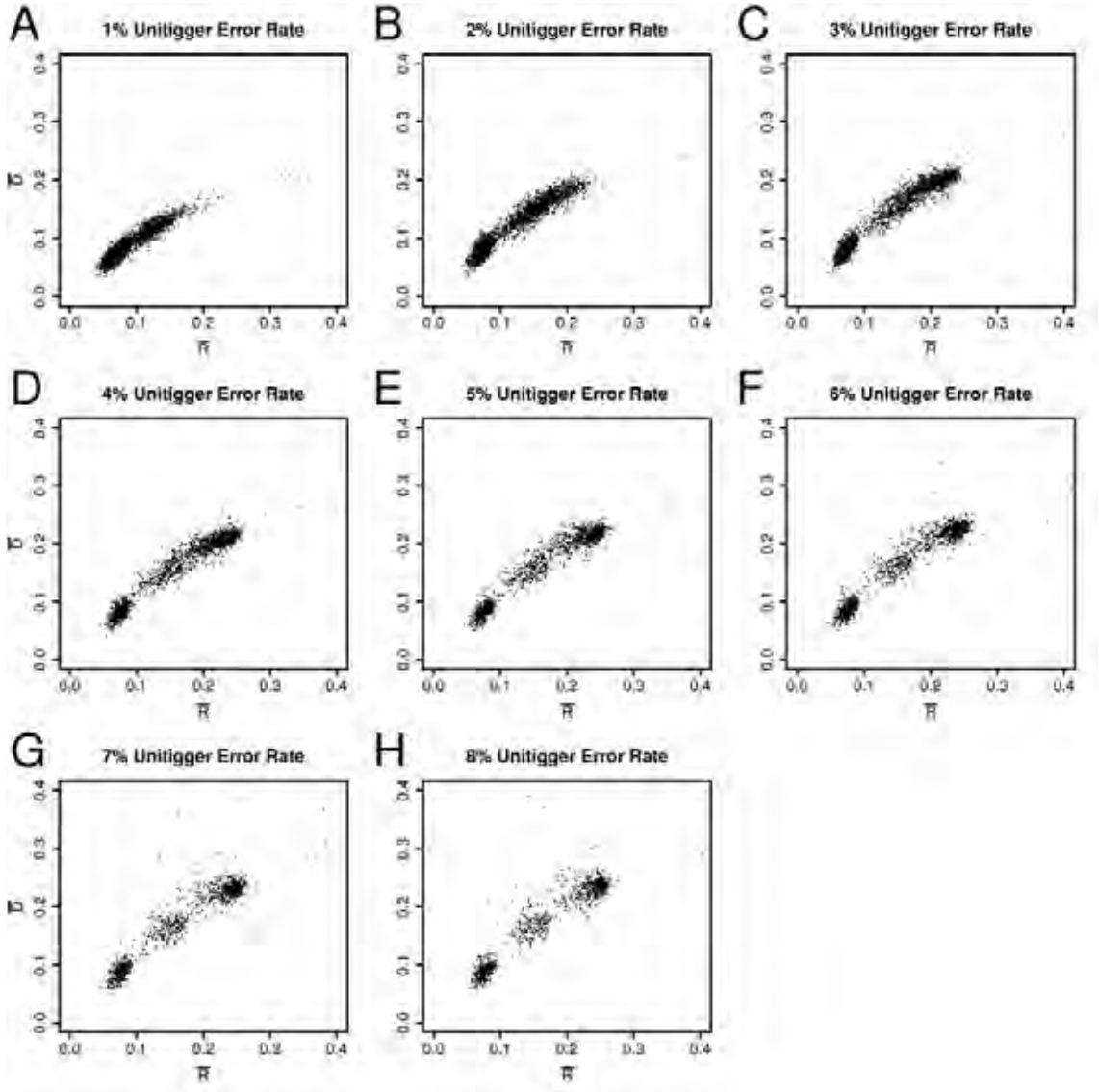

Figure 3.11: Unitigger error rates can be adjusted to improve clustering.
Three-strain *E. coli* assemblies with varying unitigger error rates.
$\bar{R}$: reads per unit of contig length. $\bar{D}$: discrepancies per unit of contig length.

A four-strain *E. coli* 4% assembly was created (Figure 3.12A). It had four clusters, and the fourth cluster's position followed the pattern of assemblies with less strains. However, the boundary between the third and fourth clusters was poorly defined. A 6% (Figure 3.12B) and 8% assembly (Figure 3.12C) were also created. The boundary between these clusters becomes clearer at higher tolerances but the density of the third cluster is severely reduced. The two-strain cluster loses fewer unitigs and the loss of unitigs from the one-strain cluster is barely noticeable. The number of unitigs in the four-strain cluster is also decreasing, but at the slowest rate. The total number of unitigs is decreasing because the mean unitig length in each cluster increases with the unitigger error rate and this is most pronounced in the four-strain assembly. All of the other

clusters are losing reads and the four-strain cluster is gaining them.

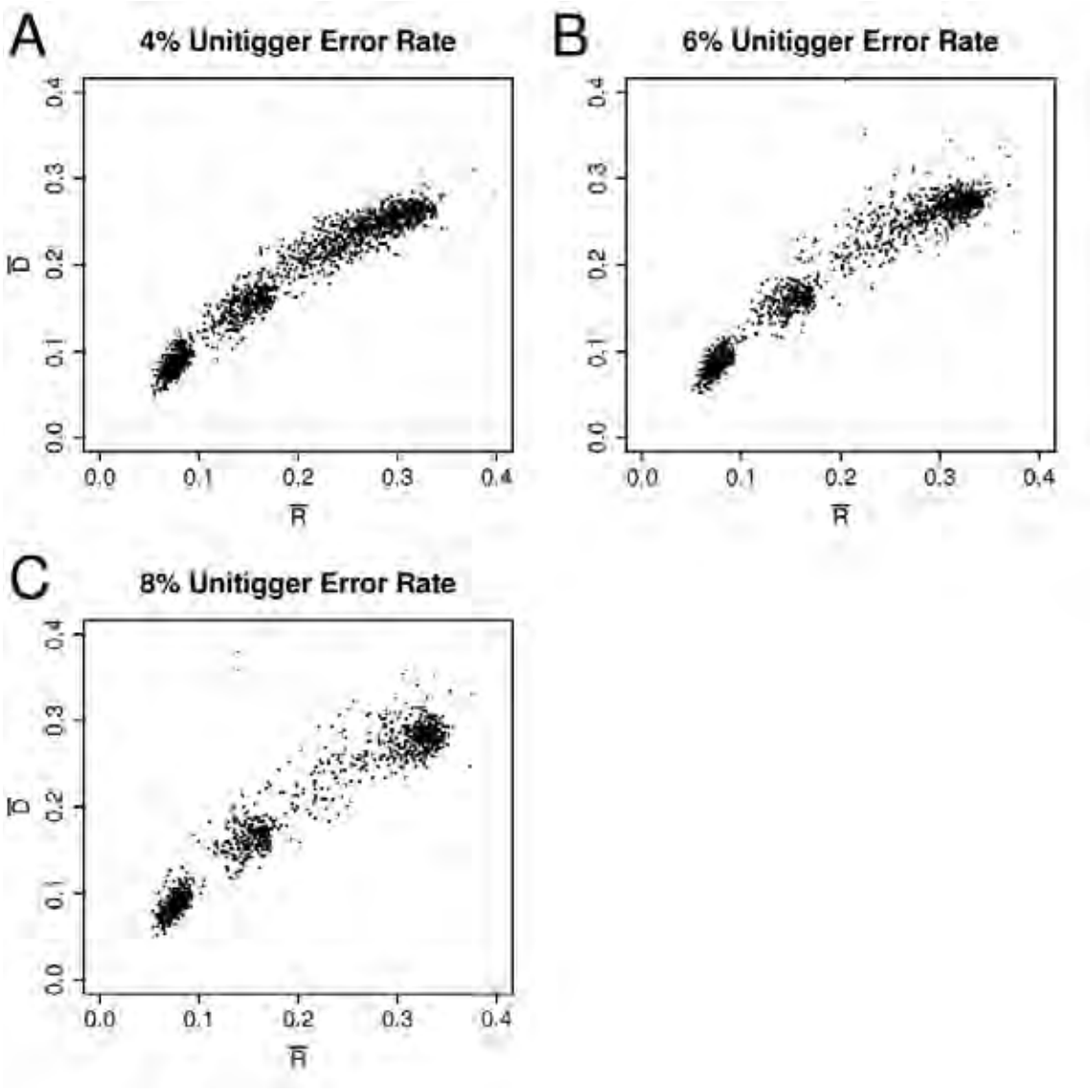

Figure 3.12: Besides the last cluster, clusters with more strains lose unitgs more easily.
Four-strain *E. coli* assemblies with varying unitigger error rates.
$\overline{R}$: reads per unit of contig length. $\overline{D}$: discrepancies per unit of contig length.

If a unitig contains sequence common to three strains, then the likelihood of the fourth strain also having similar sequence is higher than if that unitig was common to only one or two strains. Likewise, if a unitig is common to two strains then its likelihood of having similar sequence in the other two strains is higher than if it was unique to only one strain. Thus, clusters with fewer strains should be more resilient to changes to the assembly tolerances.

Four-strain *N. meningitidis* assemblies produced using the same three unitigger rates also followed the clustering pattern and the same effects on cluster density across

unitigger error rates were observed (Figure 3.13). The decrease in density of the clonal clusters is more apparent in these assemblies, though this is still most obvious in the three-strain clusters. These *N. meningitidis* assemblies used unitigs rather than contigs. These plots were produced with a larger scale than the *E. coli* assemblies because the same number of reads was used with a smaller genome. Thus, these clusters all have larger $\bar{R}'$ values.

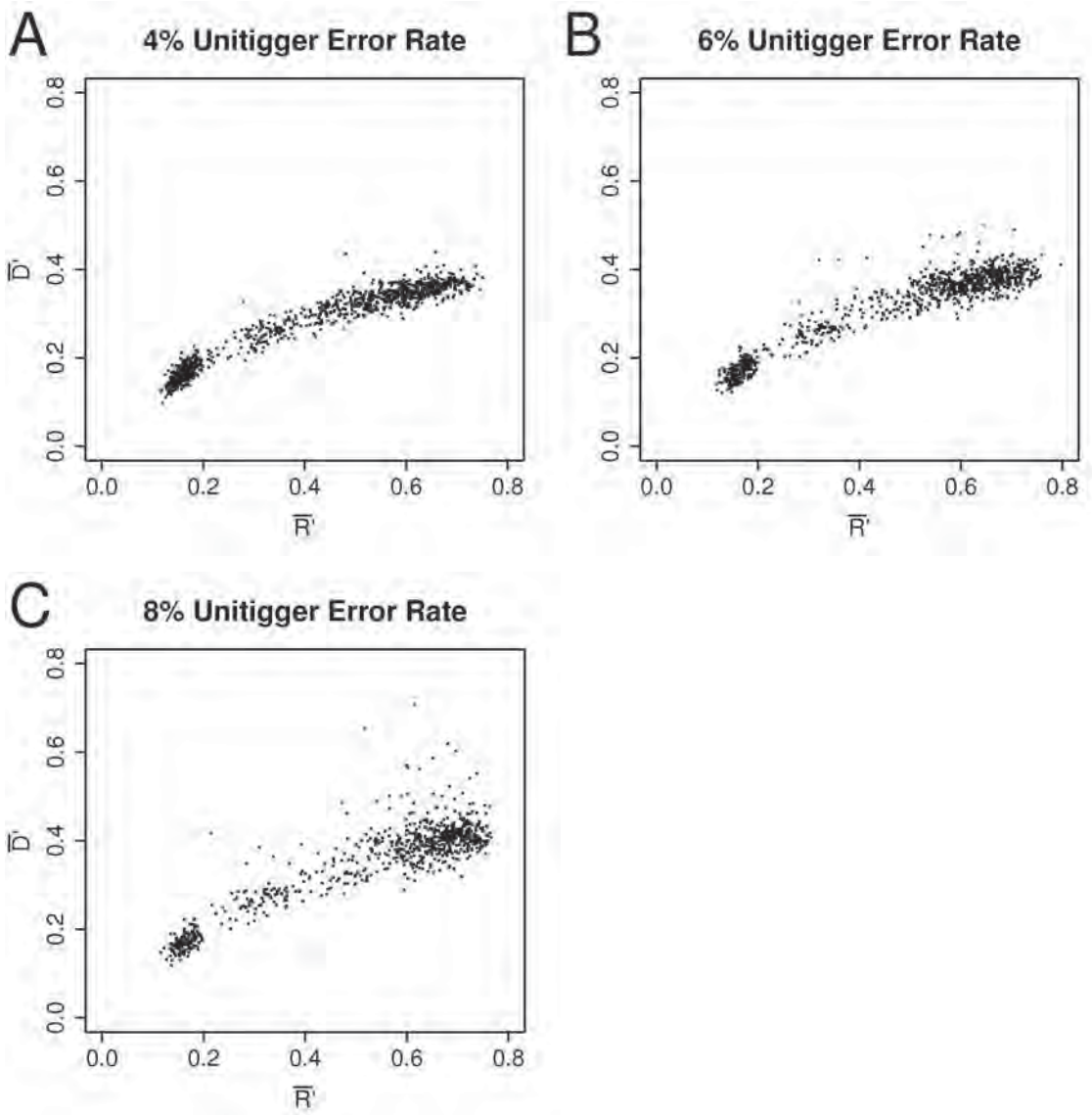

Figure 3.13: Clustering patterns also apply to *N. meningitidis* assemblies.
Four-strain *N. meningitidis* assemblies with varying unitigger error rates.
$\bar{R}'$: reads per unit of unitig length. $\bar{D}'$: discrepancies per unit of unitig length.

3.4.4 Normalisation of Coverage

The original assemblies were produced with exactly the same number of reads regardless of which species and strains were assembled. For the two-strain *S. aureus*

assemblies, lower quantities of reads led to clusters that more closely matched the positions of the *E. coli* clusters (Figure 3.14). The match between the *S. aureus* assembly with 60% reads and the standard *E. coli* assembly makes sense since the mean *S. aureus* genome length is 58% of the mean *E. coli* genome. Similarly, the mean *N. meningitidis* genome is 44% of the length of the mean *E. coli* genome.

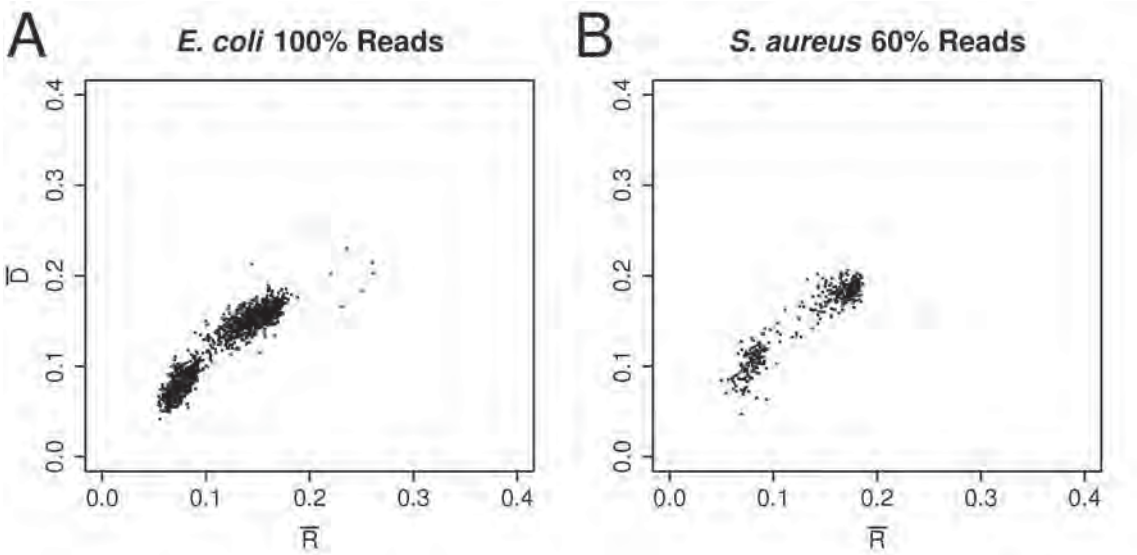

Figure 3.14: Adjusting the coverage of assemblies of distantly related species can make cluster postions more similar.
Similar two-strain assemblies from different species:
A) *E. coli*.
B) *S. aureus* with 60% of the standard number of reads.
$\bar{R}$: reads per unit of contig length. $\bar{D}$: discrepancies per unit of contig length.

When plotted together, the positions of the assembly's clusters are in decreasing order of genome size (Figure 3.15A). These assemblies of *S. aureus* and *N. meningitidis* were reproduced with the same per-strain coverage as *E. coli* leading to more consistently located clusters across species (Figure 3.15B). This normalisation showed that cluster location was highly dependent on per-strain coverage but much less dependent on species. Consistent cluster locations between species could allow accurate predictions of chimerism in one species with another species used as the training data.

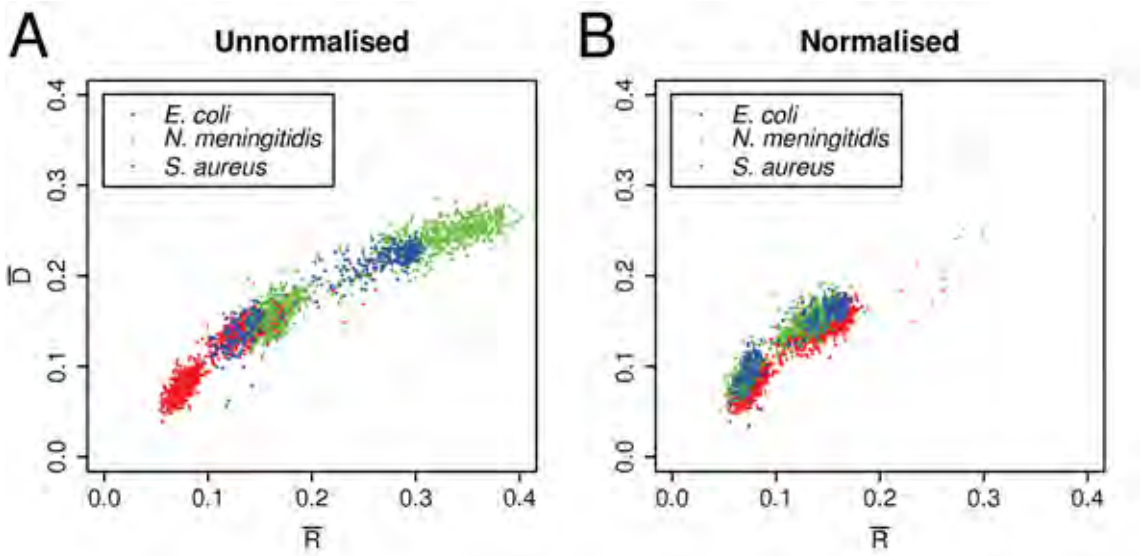

Figure 3.15: Using the same coverage gives clusters with very similar positions across assemblies of different classes and phyla.
Comparison of two-strain assemblies of *E. coli, S. aureus* and *N. meningitidis*:
A) Standard read numbers.
B) Normalised coverage.
$\bar{R}$: reads per unit of contig length. $\bar{D}$: discrepancies per unit of contig length.

Since one of the chosen variables ($\bar{R}$) is equivalent to contig read depth and the other variable is correlated to it, this effect of coverage on cluster position is not surprising. These assemblies were all produced with the same sequencing error rates and the similarities between the strains of each species have been controlled, thus the rate of discrepancy has been largely controlled as well. However, the differences in $\bar{D}$ values between the species do not have the expected correlation to strain similarity.

There is a distinct curve ($y = 0.1121 \ln x + 0.3646$, $R^2 = 0.9562$) to the clusters in Figure 3.15A. The reason why this relationship is not linear is because $\bar{D}$ saturates at higher $\bar{R}$ values. More reads mean more sequencing errors at each consensus base, on average. However, due to the method of calculating discrepancies, additional sequencing errors at a discrepancy have no effect on $\bar{D}$. Thus, the higher $\bar{R}$ becomes, the lower the increase in $\bar{D}$ values.

3.4.5 Dichotomous Prediction using Logit Regression and ROC curves

Contigs from the simulated data shown in Figure 3.7A and B and Figure 3.9B were automatically sorted into clonal and chimeric contigs by logit regression. This regression was used to classify contigs in each assembly into clonal and chimeric by

using a different *S*-binned assembly as training data. ROC plots were used to evaluate the resulting classifiers as well as showing the ideal cut-off for each. The *S*-binning of the assembly to be classified was used to create the ROC plots.

ROCR cannot be used to create a classifier with a one-strain assembly as training data, as there are no negative values. Likewise, if a cut-off for any classifier was calculated using a one-strain assembly, it could be used to perfectly classify that one-strain data by using a cut-off of one. Thus, the simplest control possible was to use a classifier and cut-off produced on a two-strain assembly to make a prediction on a one-strain assembly. To produce this classifier and cut-off, predictions were made on a two-strain *E. coli* assembly using that same assembly as training data. This produced a ROC plot with an AUC of 0.9789 (Figure 3.16A). The boundaries of the contigs classified as clonal closely matched the clonal cluster. Classification using the optimal cut-off of 0.5935 gave 550 true positives, 17 false positives, 793 true negatives and 34 false negatives (Figure 3.16B). When used on a one-strain assembly, this classifier and cut-off gave 80 true positives and 21 false positives (Figure 3.16C). The true positives correlated closely with the distinct clonal cluster. This prediction method correctly labelled the clonal clusters in both one- and two-strain assemblies as being clonal, using a classifier developed on a two-strain assembly.

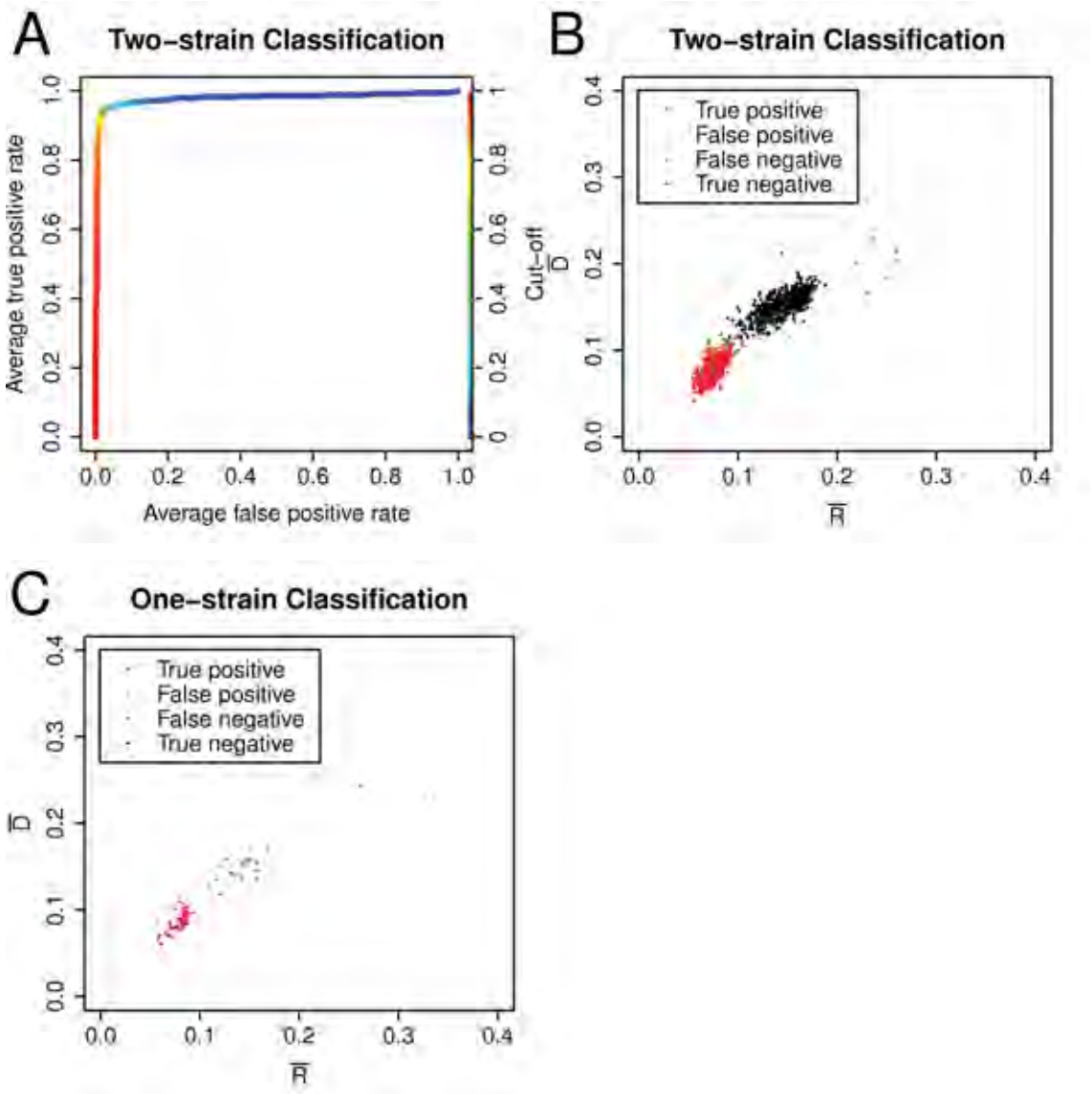

Figure 3.16: Logit classification allows accurate predictions of clonal unitigs for one- and two-strain assemblies.
Logit prediction of chimerism using a two-strain *E. coli* assembly as training data:
A) ROC plot of classifications of the same assembly.
B) Classification of contigs in the same assembly using the optimal cut-off shown in A.
C) Classification of contigs in a one-strain *E. coli* assembly using the same classifier and cut-off.
$\bar{R}$: reads per unit of contig length. $\bar{D}$: discrepancies per unit of contig length.

A four-strain *N. meningitidis* 8% assembly as shown in Figure 3.13C was used to predict which unitigs in an 8% three-strain assembly were clonal. The classifier achieved an AUC of 0.9983 (Figure 3.17A). Classification with a cut-off of 0.3952 gave 143 true positives, 3 false positives, 509 true negatives and 3 false negatives (Figure 3.17B). The unitigs classified as clonal perfectly matched the clonal cluster, discounting the three false positives. This prediction method correctly labelled the clonal cluster in a three-strain assembly as clonal using a four-strain assembly as training data. Accurate

predictions of chimerism were made for assemblies of different species, different numbers of strains and assemblies using different assembly settings.

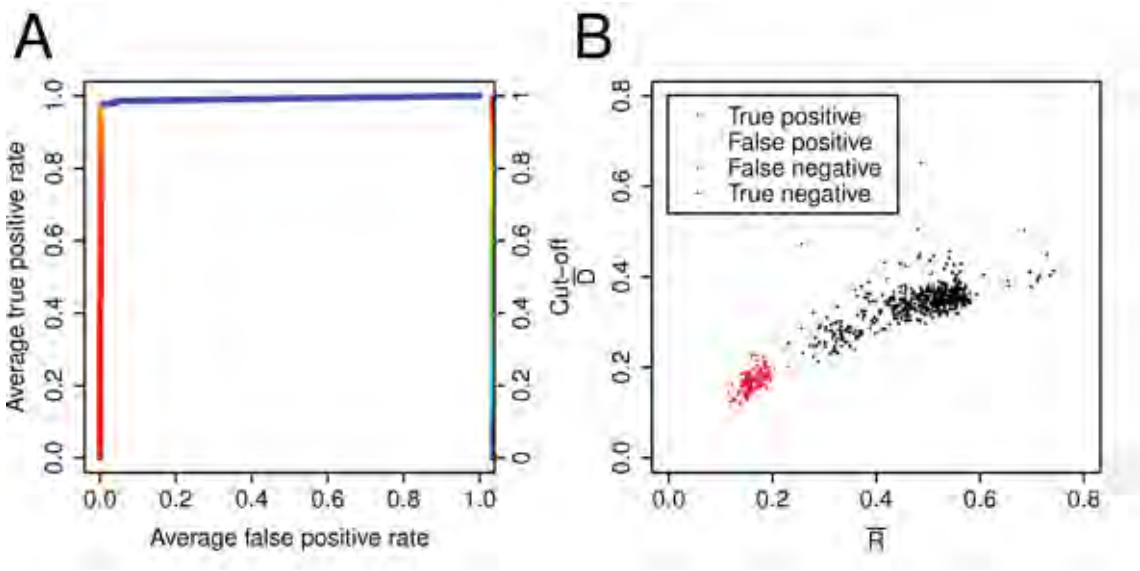

Figure 3.17: Logit predictions are also accurate in an assembly of a different species with more strains.
Logit prediction of chimerism using a four-strain *N. meningitidis* 8% assembly as training data:
A) ROC plot of classifications of 8% three-strain assembly.
B) Classification of contigs in the three-strain assembly using the optimal cut-off shown in A.
$\overline{R}$: reads per unit of contig length. $\overline{D}$: discrepancies per unit of contig length.

A two-strain normalised *S. aureus* assembly, as shown in Figure 3.15B, was used to make a prediction on a two-strain *E. coli* assembly. The classifier achieved an AUC of 0.9799 (Figure 3.18A). Classification with a cut-off of 0.4223 gave 545 true positives, 10 false positives, 798 true negatives and 39 false negatives (Figure 3.18B).

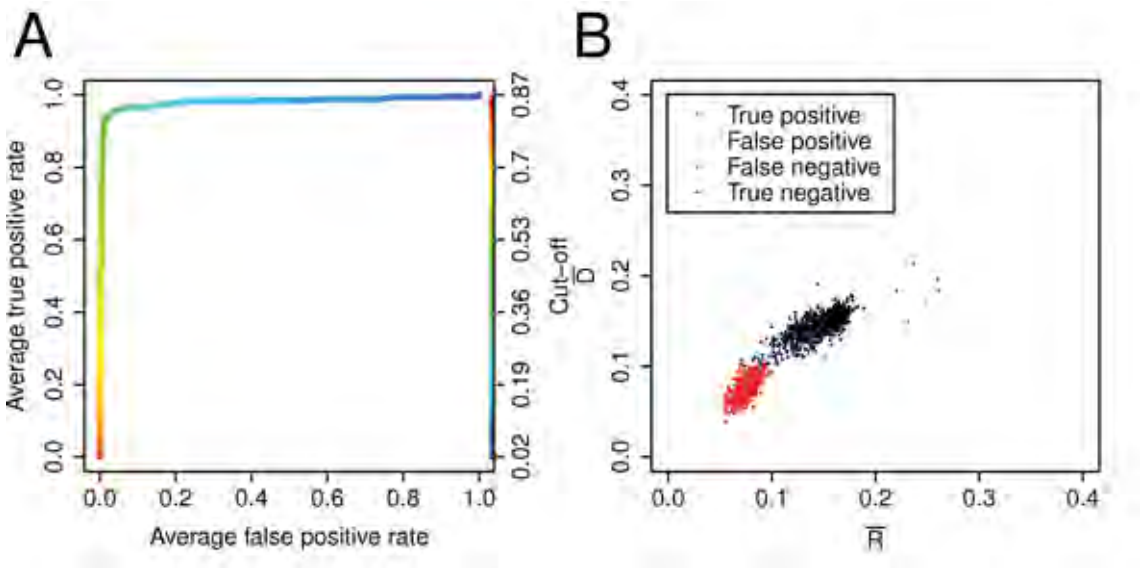

Figure 3.18: Logit predictions can also classify accurately when the training data comes from a distantly related species.
Logit prediction of chimerism using a two-strain *S. aureus* normalised assembly as training data:
A) ROC plot of classifications of an *E. coli* two-strain assembly.
B) Classification of contigs in the *E. coli* assembly using the optimal cut-off shown in A.
$\overline{R}$: reads per unit of contig length. $\overline{D}$: discrepancies per unit of contig length.

This prediction on the two-strain *E. coli* assembly had a higher AUC and less false positives than the prediction that used the same assembly as the training data. The training data was successful even though it came from a different phylum. This shows that accurate predictions of unitig chimerism can be made for an assembly of a species without access to training data from that species. Accurate predictions on the chimerism of experimentally-derived metagenomic assemblies could potentially be made by using a model organism for the training data. This is useful as it would allow reuse of training data and allow predictions on assembly data from species that do not have a genome of a close relative available.

All of the assemblies in this chapter used strains in equal proportions. This will rarely be the case in nature. Despite this, these assemblies allowed the development of a graphical analysis method based on the informative variables $\overline{R}'$ and $\overline{D}'$ (Figure 3.6). Whilst the boundaries between clusters are not always clear in the resulting scatter plots (Figure 3.8), several optimisations can help disambiguate. These include adjusting the unitigger error rate (Figure 3.9), using contour plots (Figure B.1) and, for simulated data, colouring by strain count bin (Figure 3.5 and Figure 3.6). The scatter plots cluster unitigs according to their level of chimerism across species from different classes and phyla (Figure 3.7, Figure 3.8 and Figure 3.10), and different coverage (Figure 3.10).

While the ideal unitigger error rate of 6% to 7% was not used as the standard assembly setting, accurate predictions of chimerism were still produced (Figure 3.16AB).

The unitig binning method described (Equations 3.1 and 3.2) classified clusters effectively using information on the source strains of reads (Figure 3.5 and Figure 3.6). This binning method could be used for predictions of the quantity of strains in each unitig, rather than just predicting which unitigs are clonal. It could be used to create training data needed to make these predictions and to evaluate them. In order to be used for assemblies of strains in unequal proportions, this method will need to be extended, as discussed in Chapter 4.

Both the one- and two-strain *E. coli* assemblies have multiple unitigs that are located outside the clonal clusters. These data points can be considered outliers and they are discussed in Chapter 4. The logit classification identified the clonal clusters almost perfectly but did not classify the outliers correctly. Thus, removing these outliers could greatly improve this analysis. The clonal cluster in the *N. meningitidis* three-strain 8% assembly has only three of these outliers (false negatives) (Figure 3.17). This is not because *N. meningitidis* has fewer regions of self-similarity than *E. coli*, as it in fact has more (Figure B.2). This implies the low number of erroneous classifications is due to the quantity of strains assembled together, or the higher unitigger error rate.

Since these outliers are highly conserved within a strain, they may also be highly conserved between strains. Thus the higher quantity of strains and higher unitig error rate are likely to be the cause of the low number of errors in Figure 3.17. This is because if there are more strains, then the likelihood of a unitig's sequence being common to at least two strains will be higher. Likewise, a higher unitig error rate decreases the threshold of similarity between similar sequences from different strains and increases the likelihood of these sequences being assembled together into one unitig.

3.5 Conclusion

This chapter describes informative variables that were chosen to allow a graphical analysis of simulated metagenomic assemblies. This analysis method was robust across read-depths and, to a lesser extent, assembly tolerances. A robust method of binning unitigs from these assemblies based on the number of strains represented by their reads was also developed. Accurate dichotomous predictions of the chimerism of multiple assemblies were made, including a prediction that used training data from a different species. Thus, progress has been made towards the overall aim of predicting the number of strains in a metagenomic sample.

Chapter 4

Predicting the Number and Relative Abundances of Strains

4.1 Summary

The graphical analysis method described in this Chapter was applied to sequencing data from microbial communities found in Ace Lake, Antarctica. A high fidelity unitig filtering method with minimal data loss was developed to select for species of interest.

Chapter 3 described informative variables that were chosen to allow the binning of each unitig in an assembly to ensure linkage to the strains of origin. Chapter 4 describes a pattern analysis of the clustering of the resulting scatter plots that allowed the number of strains to be predicted in multi-strain assemblies. Using this analysis with assemblies of strains in unequal ratios required rules describing the relationship between the number and proportions of strains and the positioning of clusters. The clustering package *MCLUST* was investigated for automating the analysis. The method of binning unitigs by their source strains was extended to create training data for clustering of strains in unequal ratios. Cluster density differences helped distinguish between ambiguous cluster patterns. Idealised assemblies of simulated reads without sequencing errors were produced, which allow more detail that links clusters to their strains of origin to be observed.

4.2 Introduction

After demonstrating predictions of whether the unitigs in an assembly were clonal or not (Chapter 3), it was important to determine the quantity of strains in these assemblies. Assemblies of simulated reads from multiple strains, each with the same read depth were used for dichotomous predictions (Chapter 3). Distinct clusters in scatter plots of these assemblies indicated the number of strains in each assembly. Thus, Chapter 4 describes a pattern analysis method that was developed to determine the relationship between the quantity and proportions of strains and cluster patterns.

4.2.1 Chapter Aim

The aim of the work described in this chapter was to infer the number of closely related strains in a metagenomic assembly contig and to estimate the relative abundances of those strains. Techniques to achieve this were developed using simulations and then applied to “contaminated” genome, and low complexity metagenome, data. Rules connecting cluster location and strain abundances were developed. Methods involving additional features were explored to enable the analysis of ambiguous cluster patterns.

4.3 Materials and Methods

4.3.1 Strain Choice

The strains and species used for assemblies of simulated reads were as described in Chapter 3 (Subsection 3.3.1).

4.3.2 Unitig Binning

When simulated reads from *E. coli* strains were combined in unequal proportions, the percentages of reads from each strain were scaled to allow use of the unitig binning method described in Chapter 3 (Equation 3.1). For each unitig, the percentage of reads from each strain was adjusted as follows:

$$p_i^* = \left(\frac{\frac{p_i}{q_i N}}{\sum_{j=1}^{N} \frac{p_j}{q_j N}} \right) \quad \textbf{(4.1)}$$

Where p_i^* is p_i scaled; p_i is the percentage of reads in that unitig from strain s_i; q_i is the proportion of reads in the assembly from the same strain; N is the total number of strains in the assembly and $s_i \in \{s_1, s_2, \cdots, s_N\}$.

This scaling was necessary because, in an assembly of strains in unequal proportions, the ratio of reads from each strain in a perfectly mixed unitig is also unequal. For example, if an assembly of two strains is produced from a sample with 90% of its reads from one strain, then 90% of the reads in a perfectly mixed unitig will be from that strain. In this case, the scaling method would convert the strain ratio for a unitig with a 90:10 mixture into a 50:50 mixture to allow it to be binned correctly:

$$\left(\frac{\frac{0.9}{0.9\times2}}{\frac{0.9}{0.9\times2}+\frac{0.1}{0.1\times2}} \right), \left(\frac{\frac{0.1}{0.1\times2}}{\frac{0.9}{0.9\times2}+\frac{0.1}{0.1\times2}} \right) = 0.5, 0.5.$$

Ranking the unitigs in an assembly of simulated reads based on their proximity to the expected mixture ratio clarifies the structure of the resulting scatter plot.

4.3.3 *Grinder*

Grinder (version 0.1.6; http://biogrinder.sourceforge.net) is an alternative program to *MetaSim*. The parameters provided by *Grinder* are very straight forward compared to the 454 error parameters in *MetaSim*. *Grinder* uses a single parameter for indels rather than allowing control of insertions and deletions independently. Unlike *MetaSim*, *Grinder* does not provide separate models for different sequencing technologies. Assemblies of *Grinder* reads specified the coverage of each strain rather than the amount of reads. Equal proportion three-strain assemblies were created with 20× per-strain coverage. Unequal proportions for three-strain assemblies were chosen so that the total coverage also had a sum of 60.

The error parameters in *Grinder* were set to match *MetaSim*. This led to the discovery that *MetaSim* had not been adding any substitution errors to its reads. Thus, assemblies with more realistic substitution settings were produced. However, assemblies that matched *MetaSim* were set as the standard for *Grinder* for consistency. Both kinds of assemblies used a sequencing error frequency for indels of 0.5. The sequencing error frequency for substitutions in the more realistic settings was set to 0.2. To calculate this, the ratio of substitutions to indels in the three real genome sequencing projects used for calibrating *MetaSim* in Chapter 2 and Chapter 3 were averaged. The substitution frequency rate was then chosen so that this ratio would apply, but without changing the frequency of indels.

4.3.4 *MCLUST*: Model-based Clustering

Logit regression is designed for binary systems and thus was inappropriate for predictions involving more than two classes. Since clusters in the scatter plots are strongly correlated with the level of chimerism of their unitigs, automated clustering of the data was performed.

The *R* library *MCLUST* (version 3; Fraley and Raftery 2002) was used with default parameters. The *MCLUST* package provides normal mixture modelling. It does this via expectation-maximisation and model-based clustering. *MclustDA* provides discriminant analysis by performing model-based clustering on each class in the training set. The *Mclust* function uses a Bayesian information criterion to evaluate which models and how many clusters to use (Fraley and Raftery 2006).

4.3.4.6 Outlier Tracking and Filtering

Pattern analysis provided a model of clustering in the scatter plots. However, a small number of observations did not fit this model. To improve the quality of this model, these outliers were removed from the *MclustDA* training sets. These outliers are likely due to paralogous regions within a strain's genome being erroneously assembled together. For example, consider a unitig in a two-strain assembly that contains reads from two regions of each strain. The read depth of this unitig will resemble that of a four-strain unitig. In an experimentally-derived metagenomic assembly, orthologous regions could also have increased read depths. Outliers were defined as those few unitigs on scatter plots that have higher $\overline{R}'$ than all the clearly identifiable clusters. Such outliers were detected by *Mclust* as a large sparse cluster, furthest from the origin and containing only these observations. The distance of these outliers from the final cluster mean was measured as a confirmation. *MclustDA* was originally used with its default settings: unrestricted in the number, orientation, size and shape of the clusters it could use to describe each bin in the training data. It was ultimately restricted to representing each bin as a single ellipsoidal cluster.

Outliers with similar positions were found in the scatter plots across different species and unitigger error rates. Two-strain assemblies were produced for *E. coli*, *N. meningitidis* and *S. aureus* at both 4% and 8% unitigger error rates. For these assemblies, outliers were defined as those unitigs with an $\overline{R}'$ value greater than 0.2. The 4% assemblies were repeated with reads simulated with *Grinder*. For the *Grinder* assemblies, an outlier cut-off of an $\overline{R}'$ value greater than 0.17 was used. The outliers in the different assemblies for each species were compared using *cross_match* (www.phrap.org/phredphrapconsed.html; version 1.080801). This program was used to create a mapping from outliers in each 4% assembly to unitigs in the corresponding 8% assembly and vice versa. Mappings were also calculated between the 4% assembly and the *Grinder* assembly for each species.

The genes present in the outliers from 4% assemblies were examined to determine if the outliers could be systematically filtered out. Likewise, outliers from the 8% assemblies that were not matches to any 4% outliers were examined. The consensus sequence was used to obtain *BLASTX* (Basic Local Alignment Search Tool using a

translated nucleotide query) hits against NR (Non-Redundant protein sequences database). Unique hits from the top five hits for each contig were examined.

4.3.5 Cluster Locations: Contig Binning

Assemblies of *S. aureus* COL and *S. aureus* JH1 in varying ratios were produced. The total number of reads was kept to the standard number for a two-strain assembly, as described in Chapter 3. The proportion of reads from *S. aureus* COL was varied from 10% to 90% in 10% increments. The contigs were binned into two clonal clusters and one chimeric cluster according to their proportion of strains. For the equal proportions assembly, the standard *S* bin cut-offs were used. That is, contigs were divided into those with greater than 75% *S. aureus* COL, those with greater than 75% *S. aureus* JH1 and those with 25% to 75% of each. In the other assemblies, this originally chosen threshold was too stringent for the expected inter-strain-variation signal strength. Thus, a revised strategy for thresholding with adjusted cut-offs was developed. (Table 4.1). These adjustments were made because the number of reads from each strain in a perfectly mixed chimeric contig is proportional to abundance in the sample. The cut-off values were chosen to be halfway between the values for a purely clonal and a perfectly mixed contig. For example, in the 90% *S. aureus* COL, 10% *S. aureus* JH1 assembly, the *S. aureus* COL cluster has a cut-off of 95%, which is halfway between 90% for the perfectly mixed cluster and 100% for a completely clonal cluster. Likewise, the *S. aureus* JH1 cluster has a cut-off of 55% which is halfway between 10% and 100%.

Table 4.1: Cluster cut-offs for variable proportion two-strain *S. aureus* assemblies.

% *S. aureus* Reads in Assembly		Clonal *S. aureus* Cluster Cut-offs	
JH1	**COL**	**JH1**	**COL**
10	90	JH1 > 55%	COL > 95%
20	80	JH1 > 60%	COL > 90%
30	70	JH1 > 65%	COL > 85%
40	60	JH1 > 70%	COL > 80%
50	50	JH1 > 75%	COL > 75%

The cut-offs for the clonal clusters are specified. All other contigs are assigned to the chimeric cluster.

Outlier rejection of contigs with an $\bar{R}$ value over 0.4 was performed on these assemblies. These contigs were over four standard deviations from the mean $\bar{R}$ value of

the chimeric cluster. Standard deviations for the chimeric cluster were then recalculated.

4.3.6 Peak Height Prediction

The observed heights of peaks (i.e. maximal estimated cluster kernel density) were estimated directly off the contour plots in *R* by rounding down to the nearest level. These heights were also calculated by the custom script *peak_picker.py*.

The input for *peak_picker.py* was produced using two-dimensional kernel density estimation with the *kde2d* function in R. Default bandwidths, or a bandwidth of 0.05, were used in this function. The default bandwidth was used as the standard for both the script and the contour plots. *kde2d* adjusts its default bandwidth depending on the input data. It was set to a 1000 by 1000 matrix of smoothed density values, for both the script and the plots. The points which were higher than all eight adjacent points were reported as peaks. It was assumed that the edge cases would not contain any peaks. Heights of peaks that weren't reported by the script were not examined.

When the contour plots were used, the centre of an obscured cluster was estimated. The location of the cluster was calculated using Equation 4.4. The peak centre for this cluster was then estimated using the distance from known peak centres and plot features. The shoulders of larger peaks made estimating the heights and centres of smaller nearby peaks more error prone. The height of the smaller peak tended to be overestimated. The contour plot of overlapping clusters like these is distinctive (Figure 4.9B).

The expected heights of peaks in assemblies of strains in unequal abundances were estimated. This required the peak heights in a plot with the same strains in equal abundances. The peak heights in the equal abundances assembly were divided by the number of contributing clusters. For n strains, the number of clusters that have k strains is equal to the number of distinct subsets with k items that can be chosen from a set of n items, i.e. $\binom{n}{k}$ clusters. Thus, for a three-strain equal ratios assembly, the height of the first cluster would be divided by $\binom{3}{1} = 3$, the second by $\binom{3}{2} = 3$ and the last left as is, as $\binom{3}{3} = 1$. The heights of these contributing clusters were then summed when the unequal ratio clusters overlapped sufficiently. If a cluster had no significant overlap, its expected height was set to the height of the corresponding equal ratio contributing cluster.

4.3.7 *M. frigidum* Genome Data

The *M. frigidum* genome project contained small amounts of bacterial contamination with a low GC content (Webster 2010). To determine which unitigs were from the contaminant, the mean and standard deviation of GC content for the assembly were calculated from the GC content of all the unitigs in the assembly larger than 1 kb.

The coverage of the genome was calculated from the contigs over 5 kb in length. The mean read length in these contigs was estimated by using the mean read length of all untrimmed reads. Since the reads were trimmed before assembly (Webster 2010), this means that the calculated coverage is an over estimate.

4.3.8 Filtering Metagenomic Datasets

To investigate intra-species variation in a microbial community, sequences from species other than the species of interest needed to be filtered out. For the Ace Lake samples, a filtered assembly of Sanger reads had already been produced for the GSB (Ng *et al.* 2010) and *Pelagibacter* samples (DeMaere, unpublished work). This filtering was performed post-assembly on contigs using GC content, read depth, consensus sequence length and normalised di- and tri-nucleotide frequencies. Clustering was performed using a self organising map, after which contigs pertaining to the dominant species were selected.

These dominant Sanger contigs were decomposed into reads. The unitigs in the hybrid 454 and Sanger assembly that contained these reads were recorded. These unitigs were used as a testing dataset for a ROC plot analysis. The ROC plots were used to compare di-, tri- and tetra-nucleotide frequencies. These frequencies were evaluated based on their value for filtering the hybrid assembly. The oligomer frequency that created a ROC plot with the highest AUC was chosen. A cut-off value was calculated from this ROC plot according to the method described in Chapter 3. The unitigs with a value above this cut-off were then analysed.

As a corroboration of the filtering method, the program *MEGAN* (MEtaGenome ANalyzer) (version 3.7; http://ab.inf.uni-tuebingen.de/software/megan; Mitra *et al.* 2009), was used to assign unitigs to taxa. Results were reported at the family level and

above.

4.4 Results and Discussion

4.4.1 *MCLUST*

In Chapter 3, a relationship was demonstrated between the number of clusters in a scatter plot and the number of strains in the corresponding assembly. Thus, automated clustering of the chosen variables could enable automated predictions on the number of strains present in the assembly, and, to a lesser extent, predictions on the number in each unitig to be performed. Model based clustering using the *MCLUST* package was chosen for this task. *MCLUST*'s *Mclust* function produced good results for a four-strain assembly with the default settings (Figure 4.1). However, the spacing between the cluster centres was uneven.

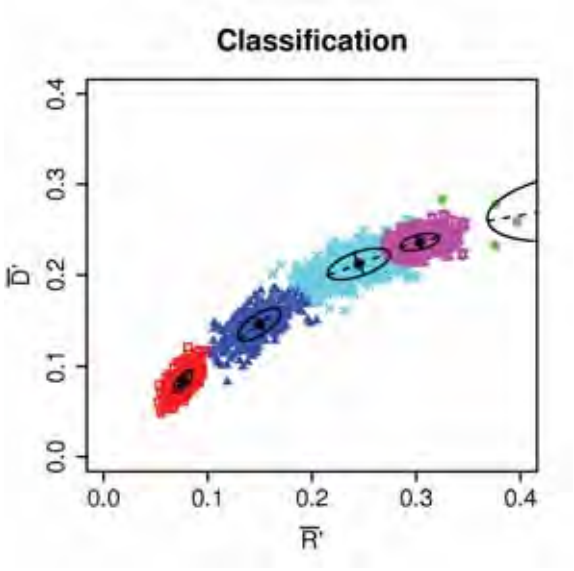

Figure 4.1: *Mclust* clustered a four-strain assembly into four unevenly spaced clusters plus an outlier cluster.
Mclust clustering of an *E. coli* four-strain assembly using default parameters.
$\overline{\boldsymbol{R}}'$: reads per unit of unitig length. $\overline{\boldsymbol{D}}'$: discrepancies per unit of unitig length.

The usefulness of *Mclust* was also limited due to its tendency to over-estimate the number of clusters at low strain counts. For example, it assigned five clusters including outliers to an *E. coli* two-strain assembly where two clusters plus outliers would have been appropriate (Figure 4.2). While the number of clusters can be specified, this is not helpful if *Mclust* is being used to determine how many clusters there are.

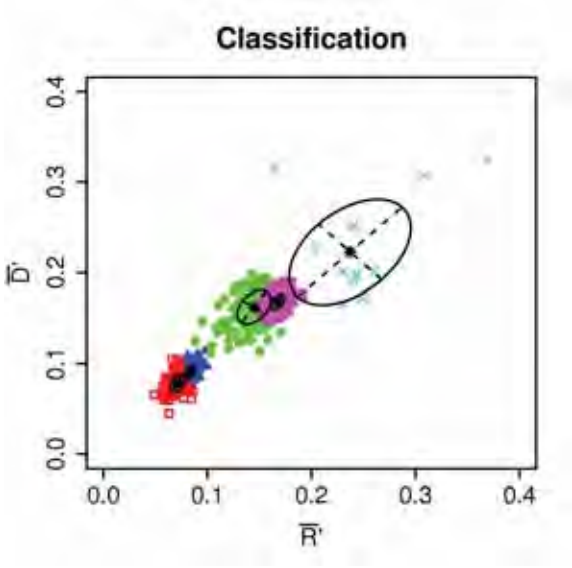

Figure 4.2: *Mclust* clustered a two-strain assembly into four clusters plus an outlier cluster.
Mclust clustering of an *E. coli* two-strain assembly using default parameters.
$\overline{\boldsymbol{R}}'$: reads per unit of unitig length. $\overline{\boldsymbol{D}}'$: discrepancies per unit of unitig length.

By default, *Mclust* compares 10 models (covariance structures), with different distributions, volumes, shapes, and orientations, each with up to nine clusters (components) (Table 4.2). The set of models to be compared and the limits on the number of clusters to be assigned can be specified. For Figure 4.1 and Figure 4.2, *Mclust* used an ellipsoidal model with varying volume and orientation but with a fixed shape (VEV) as the best fit.

Table 4.2: *MclustDA* multivariate mixture models

Multivariate Mixture Models	Explanation
EII	Spherical, equal volume
VII	Spherical, unequal volume
EEI	Diagonal, equal volume and shape
VEI	Diagonal, varying volume, equal shape
EVI	Diagonal, equal volume, varying shape
VVI	Diagonal, varying volume and shape
EEE	Ellipsoidal, equal volume, shape, and orientation
EEV	Ellipsoidal, equal volume and equal shape
VEV	Ellipsoidal, equal shape
VVV	Ellipsoidal, varying volume, shape, and orientation

The model selected by *Mclust* was not fitting well despite specifying the volume, orientation and shape appropriately. Thus, the plots were examined for further relationships. The clusters in the plots lie along the curve $y = a \ln x + c$ (Figure

3.15A). The spacing of clusters along this curve is uniform (Figure 3.3 and Figure 3.12). Controlling these aspects of the clustering could have improved the fit of the model. However, the spacing between clusters in *Mclust* cannot be controlled. The source code of *MCLUST* was examined, but the time required to modify the package was too great. Since it was not feasible to change the models in *Mclust*, a different method of utilising the observed cluster spacing pattern in *MCLUST* was sought.

4.4.2 *MclustDA*

MclustDA is a *MCLUST* function that uses training data to predict the locations of clusters. The use of *MclustDA* was investigated as the training data could provide an implicit guide to the spacing of clusters.

4.4.2.1 Investigation, Filtering and Comparison of Outliers

For all three species used in simulated assemblies (*E. coli*, *S. aureus* and *N. meningitidis*), there were unitigs that had a much higher read depth than expected. Outlier rejection was used to improve the *MclustDA* training data.

To determine whether outliers in assemblies of different species and with different assembly settings had similar sequences, 4% and 8% two-strain assemblies of each of the three species were produced (Figure C.1). For each of the three species, outliers in 4% *Grinder* assemblies were compared with those in the *MetaSim* 4% assemblies. For each of these species, *cross_match* was used to map outliers in each assembly to the outliers in the other assemblies of the same species.

Of the 80 outliers in the nine assemblies, 63 mapped to outliers in assemblies with different unitigger error rates and to those in assemblies using a different read simulator (Table C.1 and Table C.3). A high proportion of repetitive genes such as phage proteins and transposases were reported in these outliers by *BLASTX* using the NR database (Table C.1 and Table C.3). However, the set of genes detected was not sufficiently predictable to allow sequence based filtering of the outliers. Filtering sequences by gene type could require a closely related species with an annotated genome. This restriction would be excessive for metagenomic studies because the species of interest and the close relatives of that species are often poorly studied.

Analogous outliers were found for clusters with less than the maximum quantity of strains. These often overlap with the adjacent cluster furthest from the origin. Filtering of these kinds of unitigs could also improve the analysis.

An *N. meningitidis* four-strain *Grinder* assembly was used as training data for clustering of a three-strain assembly. *Mclust* was run on the training data and any observations that were beyond the last obvious cluster were classed as outliers and removed (Figure 4.3). This last outlier cluster is easy to distinguish visually as it is a much larger ellipse containing far fewer observations. The outliers identified in this assembly are 1.9 to 12.9 standard deviations from the four-strain cluster mean.

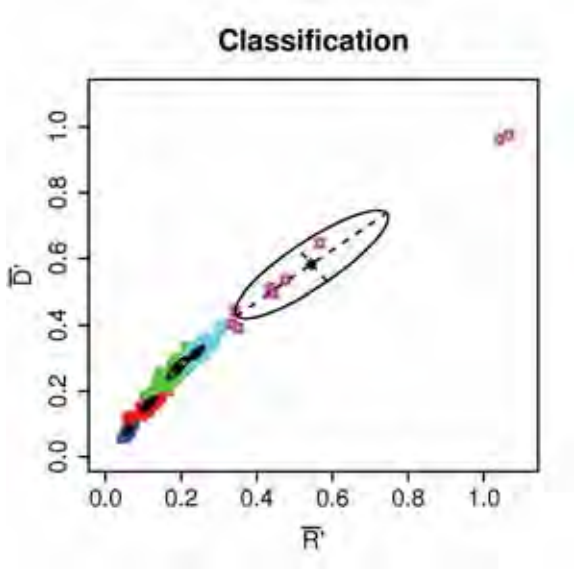

Figure 4.3: *Mclust* can be used to detect outliers.
Mclust clustering of an *N. meningitidis* four-strain *Grinder* assembly using default settings. $\bar{R}'$: reads per unit of unitig length. $\bar{D}'$: discrepancies per unit of unitig length.

4.4.2.2 Training Data

In Chapter 3, a binning method was described that assigned a real number, the strain number estimator S, to each unitig (Equation 3.1). This continuum was then transformed back into discrete bins by specifying a width range of this number. Both restrictions in the width of the discrete S bins and the number of standard deviations from the mean (μ) $\bar{R}'$ and $\bar{D}'$ values for each bin were specified to finetune this binning (Figure C.2 and Figure C.3). A compromise was required between the number of unitigs in a bin and the accuracy of the cluster centres, i.e. between signal strength and quality. To test this parameter space, an *N. meningitidis* four-strain assembly was used to make predictions on an equivalent three-strain assembly. S bin widths of ± 0.05, 0.1, 0.2, 0.3,

0.4 and 0.5 were used. Each of these *S* widths was used twice – once with σ widths of 1 (Table C.3) and once with 1.5 (Table C.4). Table 4.2 and Table C.5 describe the models used to cluster each bin. A setting of ± 1 σ ± 0.2 *S* achieved the lowest test error.

4.4.2.3 Results

MclustDA achieved similar but inferior results to *Mclust* when using an *N. meningitidis* four-strain *Grinder* assembly to make predictions on a three-strain *Grinder* assembly of the same species. *MclustDA* achieved AUCs of 0.973, 0.888 and 0.978 for the three clusters and a mean AUC of 0.946; *Mclust* gave AUCs of 0.979, 0.923, 0.970 and a mean of 0.957. This experiment was selected as it should have produced optimal results for *MclustDA*. The training and test data in this case are very similar except that the training data has one more cluster. *MclustDA* should outperform *Mclust* in some situations such as two-strain assemblies where *Mclust* detected too many clusters.

The even spacing in clusters in Chapter 3 only applied because the read depth for each strain was the same. When the proportion of reads from each strain is varied, the clusters move in relation to each other as described in the next section. Thus, *MclustDA* would not be practical for strains in unequal ratios as the training data would probably not match the test data. Also, the single strain read depth in training data needs to match the assembly being analysed. Estimating this read depth in real samples would be difficult but possible, as the clonal cluster is generally easily identified visually. However, it may not be possible to estimate this to the required accuracy. Lastly, *MclustDA* requires training data with at least as many clusters as the test data, which may also prove restrictive.

Because of the variable spacing between clusters, constraining model spacing in *Mclust* would not be of much benefit. Without a different method of constraining the clustering performed by *Mclust*, this function is of limited use. Likewise, without realistic training data, *MclustDA* is of limited use. Thus a different method of predicting strain quantities and abundances was necessary.

4.4.3 Rules for Structure of Plots

4.4.3.1 Location of Clusters

When there are equal numbers of reads per strain in an assembly, the scatter plot of this data should have one visible cluster for each strain. All but one of the visible clusters are made of multiple overlapping clusters with coincident centres, i.e. $\binom{n}{k} \geq 2$, where $n, k \in \mathbb{Z}^+$, $n > 1$ and $k < n$. The exception is the cluster furthest from the origin which denotes a mixture of all the strains in the assembly, i.e. $\binom{n}{n} = 1$. Figure 4.4A shows two *S. aureus* strains that were assembled together. Both strains produce a clonal cluster which has a coincident centre with the other (red and blue). There is also a two-strain cluster (purple).

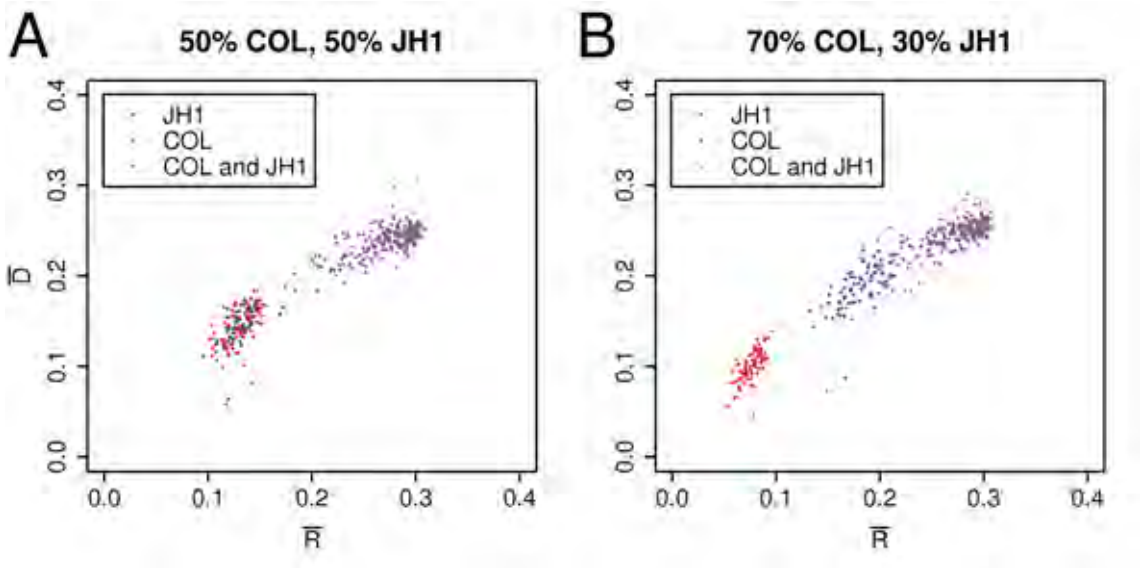

Figure 4.4: The rule of one visible cluster per strain only applies to strains in equal proportions.
Two-strain *S. aureus* assemblies: Red contigs: *S. aureus* JH1. Blue contigs: *S. aureus* COL. Purple contigs: *S. aureus* JH1 and *S. aureus* COL.
A) Strains in equal proportions.
B) 70% of reads from *S. aureus* COL, 30% from *S. aureus* JH1.
$\overline{\boldsymbol{R}}$: reads per unit of contig length. $\overline{\boldsymbol{D}}$: discrepancies per unit of contig length.

In an assembly with three strains in equal proportions, both the clonal and two-strain visible clusters are each composed of three clusters with coincident centres (Figure 4.5). When the strains are not in equal proportions, the centres of clusters with the same level of chimerism will no longer be coincident (Figure 4.4B). The degree of coincidence is dependent on the relative proportions of the strains.

When the strains assembled are in equal proportions, the clusters are evenly spaced.

When the proportions of reads from each strain are varied, more clusters are visible and are no longer evenly spaced.

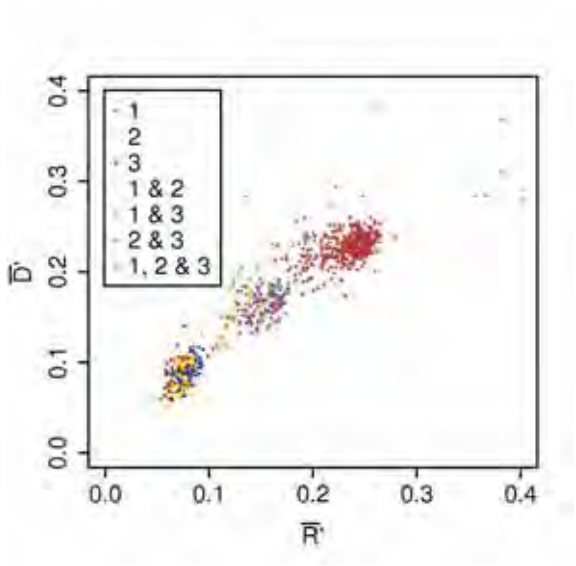

Figure 4.5: Three-strain assemblies contain seven clusters.
Three-strain *E. coli* assembly.
Red contigs: $\boldsymbol{S} < 1.5$ and *E. coli* 55989 > 50%.
Yellow contigs: $\boldsymbol{S} < 1.5$ and *E. coli* APEC O1 > 50%.
Blue contigs: $\boldsymbol{S} < 1.5$ and *E. coli* ATCC 8739 > 50%.
Orange contigs: $1.5 \leq \boldsymbol{S} \leq 2.5$ and *E. coli* 55989 > *E. coli* ATCC 8739 and *E. coli* APEC O1 > *E. coli* ATCC 8739.
Purple contigs: $1.5 \leq \boldsymbol{S} \leq 2.5$ and *E. coli* 55989 > *E. coli* APEC O1 and *E. coli* ATCC 8739 > *E. coli* APEC O1.
Green contigs: $1.5 \leq \boldsymbol{S} \leq 2.5$ and *E. coli* APEC O1 > *E. coli* 55989 and *E. coli* ATCC 8739 > *E. coli* 55989.
Brown contigs: $\boldsymbol{S} > 2.5$.
Strain 1: *E. coli* 55989. Strain 2: *E. coli* APEC O1. Strain 3: *E. coli* ATCC 8739.
$\bar{\boldsymbol{R}}'$: reads per unit of unitig length. $\bar{\boldsymbol{D}}'$: discrepancies per unit of unitig length.

Two-strain *S. aureus* assemblies were produced with varying proportions of reads but the same total number of reads. The proportion of *S. aureus* JH1 was varied from 10% to 50% by 10% increments. This means that the proportion of *S. aureus* COL varied from 90% down to 50%. For the assembly with 90% of the reads from *S. aureus* COL (Figure 4.6), there are three clusters, two of which overlap. There is a large gap between the bottom red clonal cluster, and the top blue clonal and purple two-strain clusters. The red clonal cluster contains only reads from the less abundant *S. aureus* JH1 strain. The blue clonal cluster has reads from the more abundant *S. aureus* COL strain and the tightly clustered two-strain purple cluster contains contigs with reads from both strains. The two clonal cluster centres approach coincidence as the proportions of strains

approaches equality. The *S. aureus* JH1 cluster rises and the *S. aureus* COL cluster drops and separates from the mixed cluster (Figure 4.6B to E).

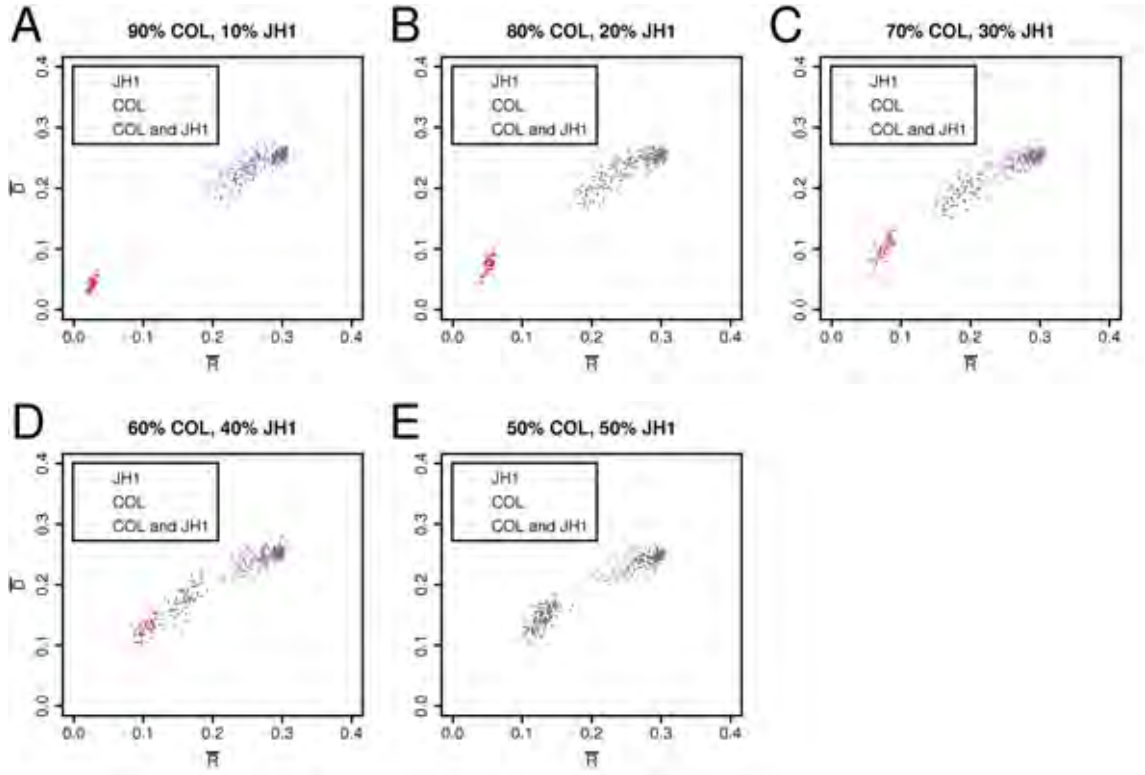

Figure 4.6: Cluster positions move predictably due to changes in strain proportions.
Two-strain *S. aureus* assemblies with variable strain proportions. The percentage of the strains *S. aureus* COL and *S. aureus* JH1 are as indicated.
$\bar{\boldsymbol{R}}$: reads per unit of contig length. $\bar{\boldsymbol{D}}$: discrepancies per unit of contig length.

The locations of the cluster centres on the $\bar{R}$ axis were plotted against strain read depth (Figure 4.7). The two-strain clusters used the sum of the read depths for the two strains. For the clonal JH1 clusters, the $\bar{R}$ values have a linear relationship (R^2 = 0.9995). The *S. aureus* COL clonal clusters also have a linear relationship (R^2 = 0.9992). The pattern from the *S. aureus* JH1 clusters continues with the *S. aureus* COL clusters (R^2 = 0.9996) and two-strain clusters (0.9965). The mixed *S. aureus* COL and *S. aureus* JH1 clusters all have similar $\bar{R}$ values (μ = 0.2781, σ = 0.003927).

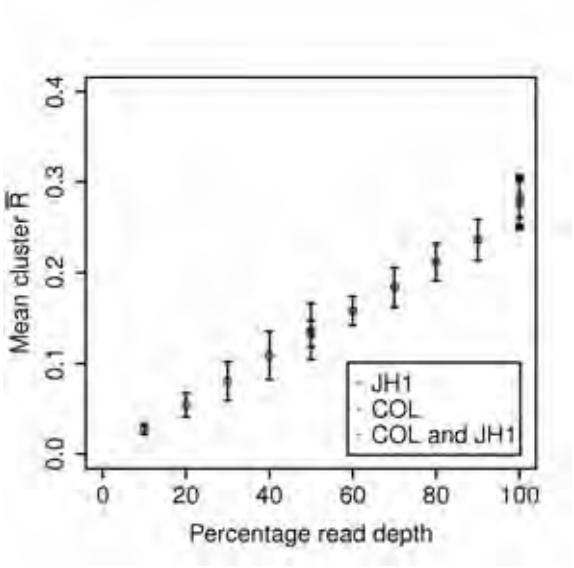

Figure 4.7: There is a strong linear relationship between cluster location and strain proportions.
Two-strain *S. aureus* assemblies with variable strain proportions: mean cluster $\bar{R}$ values versus one- and two-strain cluster read depths.
Red and blue data points denote single-strain cluster values of *S. aureus* JH1 and *S. aureus* COL, respectively. Purple data points denote clusters of two-strain contigs. Error bars are ± 1 σ.
$\bar{R}$: reads per unit of contig length.

Three-strain *E. coli Grinder* assemblies were produced with their strains in varying proportions and with substitution errors (Figure C.4). The locations of the clusters in these assemblies followed the same pattern as in the *S. aureus* scatter plots. The linear relationship applies across the combined dataset of clonal, two-strain and three-strain clusters (R^2 = 0.9863) (Figure C.5). The main deviation from this pattern is the *E. coli* APEC O1 and *E. coli* ATCC 8739 clusters (green) (Figure C.5). These clusters contain a low number of unitigs which decrease the accuracy of the cluster centres and the linearity of their relationship (R^2 = 0.7042). The low number of unitigs is due to these strains having a lower similarity than the other combinations of strains (Table 3.1).

Formulae to predict the location of clusters given the read depths of the strains was developed. Example formulae are given for assemblies with two and three strains (Equations 4.2 and 4.3) and then a general formula is given (Equation 4.4). If there are two strains with read depths of d_1 and d_2 times, then there will be clusters at:

$$d_1 c, d_2 c, d_1 c + d_2 c\, \bar{R}' \quad \textbf{(4.2)}$$

Where c is a constant; d_1, d_2 and $c \in \mathbb{R}$ and d_1, d_2 and $c > 0$.

For three strains, with read depths of d_1, d_2 and d_3 times, there will be clusters at:

$$d_1c, d_2c, d_3c, d_1c + d_2c\ , d_1c + d_3c, d_2c + d_3c, d_1c + d_2c + d_3c\ \bar{R'} \quad \textbf{(4.3)}$$

Where $d_3 \in \mathbb{R}$ and $d_3 > 0$.

For n strains, with read depths of d_1, d_2, d_3, d_4, ... , d_{n-2}, d_{n-1} and d_n times, there will be clusters at:

Combinations of one strain: $d_1c, d_2c, \cdots, d_nc;$

Combinations of two strains:
$$d_1c + d_2c, d_1c + d_3c, \cdots, d_1c + d_nc,$$
$$d_2c + d_3c, \cdots$$
$$\vdots \qquad \ddots$$
$$d_{n-1}c\ +\ d_nc;$$

Combinations of three strains:
$$d_1c + d_2c + d_3c, d_1c + d_2c + d_4c, \cdots$$
$$\vdots \qquad \ddots$$
$$d_{n-2}c\ +\ d_{n-1}c\ +\ d_nc;$$

$$\vdots$$

Combination of n strains: $d_1c + d_2c + \cdots +\ d_nc\ \bar{R'}$ **(4.4)**

Where d_4, ... , $d_n \in \mathbb{R}$; $n \in \mathbb{Z}^+$ and d_4, ... , $d_n > 0$.

For n strains, there are $\binom{n}{k}$ clusters that have k strains, giving a total of:

$$\sum_{k=1}^{n} \binom{n}{k} = 2^n - 1 \text{ clusters.} \quad \textbf{(4.5)}$$

By using Equation 4.4, the positions of clusters can be used to determine the proportions and quantities of strains for most assemblies with a low numbers of strains.

4.4.3.2 Peak Picking

Given the limitations of *Mclust* and *MclustDA*, an alternative method of locating cluster

centres for real data was sought. Peak picking tools were sought but were too specific in their design, e.g. the *R* library *rNMR* (http://rnmr.nmrfam.wisc.edu; Lewis *et al.* 2009). Since contour plots clustered equal proportion assemblies well (Figure 4.8), a custom peak picking tool based on contour plots was developed.

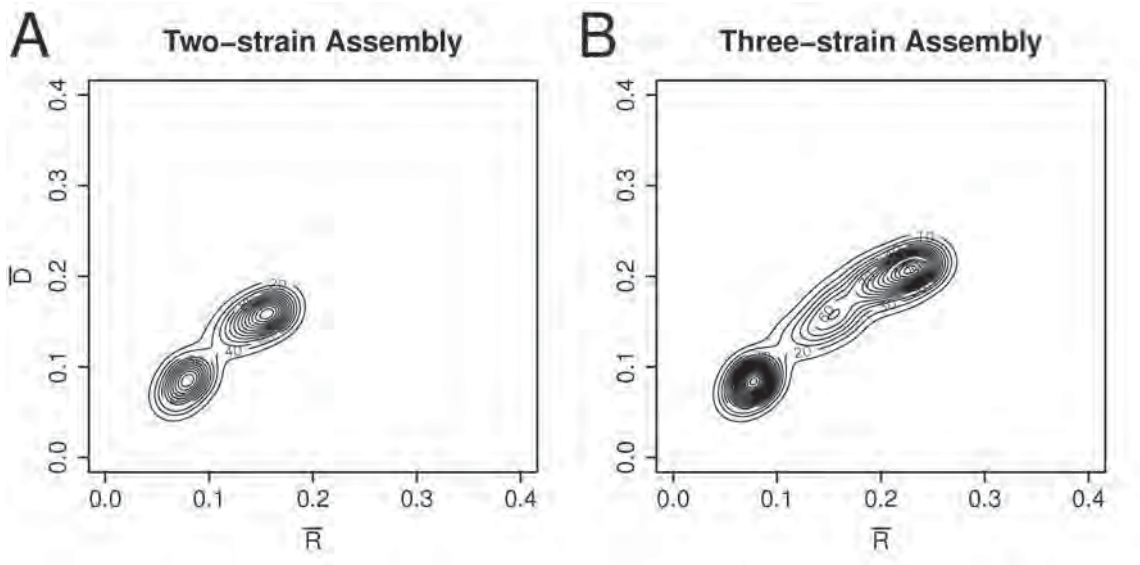

Figure 4.8: Contour plots show two and three clear clusters for two- and three-strain equal proportion assemblies, respectively.
E. coli assembly contour plots. Bandwidth = 0.05.
$\bar{\boldsymbol{R}}$: reads per unit of contig length. $\bar{\boldsymbol{D}}$: discrepancies per unit of contig length.

4.4.3.3 Densities of Clusters

The density of a cluster provides information about the unitigs that form it. For example, a cluster formed by two completely overlapping clonal clusters would be denser than a single clonal cluster. If the underlying clonal and chimeric clusters in a scatter plot have distinct densities then this can decrease the possible interpretations of each visible cluster. For example, the combination of the two kinds of clusters should be noticeably different from either kind of single cluster or two overlapping clusters of the same kind. The density of a three-strain cluster may also be noticeably denser than a two-strain cluster.

Considering only the positions of cluster centres, it would be possible to construct assemblies with different numbers of strains that have ambiguous patterns. For example, consider an assembly of three strains, each with a read depth of 10×. This assembly would be largely indistinguishable from a two-strain assembly with 10 and 20× per-strain read depths. Both would have clusters centred around 10, 20 and 30× per-strain

read depths. However, assemblies of this nature should be distinguishable when cluster density is taken into account.

Successful predictions of density patterns were made using contour plots and the associated underlying kernel density estimations. For example, an *E. coli Grinder* assembly with all three strains at 20× read depth had peak heights of 135, 55 and 140 (Figure 4.9A). This was used to predict the peak heights in an equivalent assembly with per-strain read depths of 12, 24 and 24× (Figure 4.9B and Table 4.3). The calculated densities of overlapping clusters in Figure 4.9A were used to estimate the peak heights at the five distinct cluster centres in Figure 4.9B. The second last cluster in this assembly was close to the large final peak and thus appears larger.

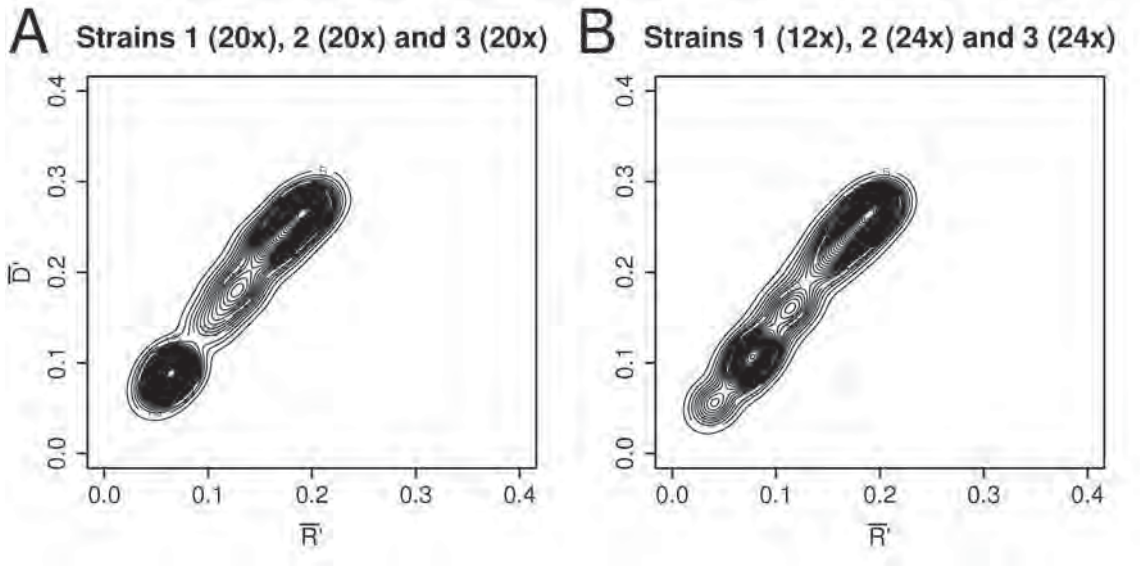

Figure 4.9: An equal proportion assembly can be used to estimate the densities of clusters in an assembly with unequal proportions.
Contour plots of *E. coli* three-strain assemblies with a bandwidth of 0.05.
Strain 1: *E. coli* 55989. Strain 2: *E. coli* APEC O1. Strain 3: *E. coli* ATCC 8739.
$\overline{\boldsymbol{R}}'$: reads per unit of unitig length. $\overline{\boldsymbol{D}}'$: discrepancies per unit of unitig length.

Table 4.3: Peak heights in 12, 24, 24× *E. coli* assembly.

Clusters	Formula	Estimated	Observed
12× clonal cluster	$\frac{1}{3} \times 135$	45	35
Two overlapping 24× clonal clusters	$\frac{2}{3} \times 135$	90	110
Two overlapping 12× with 24× two-strain clusters	$\frac{2}{3} \times 55$	36.67	50
Two-strain cluster of both 24×	$\frac{1}{3} \times 55$	18.33	40
Three-strain cluster	140	140	140

The contour plots did not always distinguish between clusters sufficiently, even after adjusting the bandwidth settings. Thus, the locations of peak centres had to be estimated using the read depths of the contributing strains. The density predictions also

required knowledge of the density values for an equal proportion assembly of the same strains. There was a general pattern in the equal-proportion assembly heights across species. The three-strain equal 20× ratio *E. coli* assembly had peaks of 185, 70 and 180, at a bandwidth of 0.05. The equivalent *N. meningitidis* assembly had peaks of 175, 40 and 170. These two species have ratios of peaks of 1:0.378:0.973 and 1:0.229:0.971, respectively. The average of these two peak-ratios could be sufficient for predictions of densities in an experimentally-derived metagenomic assembly. For two-strain assemblies of *Grinder* reads, *E. coli* had peaks of 255.61 and 279.47 and a ratio of 1:1.094 (*peak_picker.py*, bandwidth = 0.05). For *N. meningitidis*, the peaks were at 212.94 and 301.65 and the ratio was 1:1.417. *S. aureus* peaks were at 237.66 and 395.31 with a ratio of 1:1.663. While these ratios are a little different, these assemblies all have a denser chimeric cluster.

Various factors can affect how distinct the heights of different kinds of clusters are. Higher unitigger error rates lead to lower densities for the clonal clusters as they increase assembly (Figure 3.11). Likewise, a lower sequencing error rate will also increase assembly (Figure 4.10). Furthermore, more similar strains will have less distinct sequence and thus less dense clonal clusters. In Figure 3.15A, the order of decreasing clonal cluster density matches the order of alignment and identity scores, i.e. *E. coli*, *N. meningitidis*, *S. aureus* (Subsection 3.3.1).

The degree to which different combinations of clusters can be distinguished will depend on the strength of the signal and the signal to noise ratio. That is, if there are more observations and less noise due to contaminating reads then this technique will be more useful. Even if the exact densities cannot be accurately predicted, a qualitative manual approach could still aid in interpretation. This could be done by utilising densities and other visual cues such as cluster size and $\overline{D}'$.

4.4.4 Idealised Assemblies with Zero Sequencing Errors

Three-strain variable proportion *E. coli* assemblies of simulated reads without any sequencing errors were analysed. The boundaries in the scatter plots for these assemblies were not clear (Figure C.6). Thus, the filtering on these assemblies was increased to 2.5 kb (Figure 4.10). In these assemblies, the clonal clusters were distinct from the two-strain clusters in their size and location. The clonal clusters have lower $\overline{D}'$

values and a smaller range in these values as seen in a 20, 20, 20× assembly (Figure 4.10A). However, the two-strain clusters could still obscure the clonal ones. This is seen in a 12, 18, 30× assembly where the blue 30× clonal unitigs are directly below and very close to the 12, 18× chimeric cluster (Figure 4.10B). This ambiguity is accentuated because the lack of errors increased assembly and therefore there are only a few observations for the clonal assemblies. The stricter length cut-off of 2.5 kb further decreased the number of observations. More stringent assembly parameters (i.e. lower unitigger error rates) may allow the detection of the obscured clusters. Filtering of sequencing errors could enable some of the extra information in zero error assemblies to be available to non-simulated reads. Analysis of experimentally-derived metagenomic samples by filtering of sequencing errors is discussed further in Chapter 5.

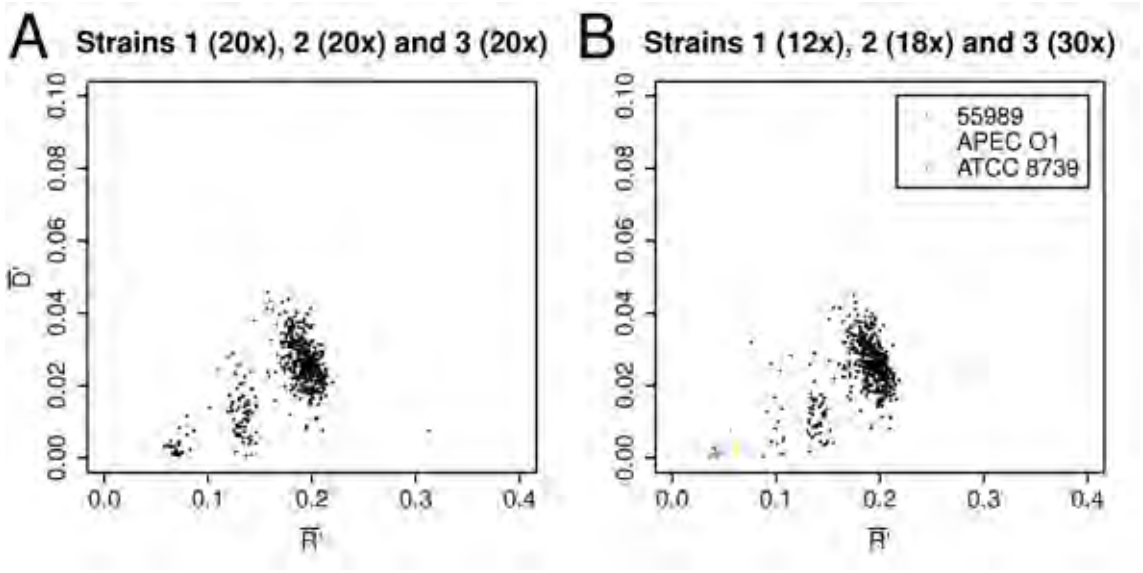

Figure 4.10: The clonal clusters in idealised assemblies have lower $\overline{D}'$ values and a smaller range in these values.
Three-strain *E. coli* assemblies with zero sequencing errors and a 2.5 kb cut-off.
Strain 1: *E. coli* 55989. Strain 2: *E. coli* APEC O1. Strain 3: *E. coli* ATCC 8739.
$\overline{R}'$: reads per unit of unitig length. $\overline{D}'$: discrepancies per unit of unitig length.

4.4.5 Analysis of Single Genome and Environmental Sequence Data

Ace Lake is located in the Vestfold Hills, Antarctica. Metagenomic samples were taken from six depths in the lake. These samples were fractionated on successive 3.0, 0.8 and 0.1 μm filters. The dominant phylum (excluding viruses) in the 0.1 to 0.8 μm size fraction of the 11.5 m sample is Alphaproteobacteria, predominantly from the SAR11

clade. The dominant species from this phylum is closely related to *Pelagibacter ubique* HTCC1062 and HTCC1002 strains (Lauro *et al.* 2010). The dominant species in the 12.7 m sample is a member of Green Sulfur Bacteria (GSB) (*Chlorobiaceae*) (Ng *et al.* 2010). The GSB was also isolated on a 0.1 μm filter after passing through a 0.8 μm filter. These metagenomic samples were sequenced using both Sanger (3730*xl* capillary sequencers) and 454 (GS20 FLX Titanium) technologies (Lauro *et al.* 2010). Hybrid assemblies of 454 and Sanger data were produced for each sample using the *Celera assembler* with a unitigger error rate of 3% (DeMaere, unpublished work, Ng *et al.* 2010). The archaeon *Methanogenium frigidum* was isolated from Ace Lake (Franzmann *et al.* 1997) and its genome was sequenced using the GS20 FLX Titanium. A 454 assembly of the *M. frigidum* sequence was produced with a 4% unitigger error rate (Webster 2010).

4.4.5.1 M. frigidum

An assembly of *M. frigidum* was analysed. The sample that was sequenced had bacterial contamination which was mostly removed before sequencing. The plot of this assembly has a single cluster near the origin (Figure 4.11A). This is as expected for a clonal population. The cluster has a smooth density profile indicating that it is not multiple overlapping clusters (Figure 4.11B). Reads of definite bacterial origin were identified as having a low GC mean content of 33% (Webster 2010). The mean GC content of *M. frigidum* is 49 ± 0.045% (Figure 4.11C). A unitig with a GC content of 35% would be more than three standard deviations from the mean and thus unlikely to be from *M. frigidum*. Suspect low GC unitigs were few in number and spread across the long diagonal axis of the cluster (Figure 4.11D). Removing these did not affect the analysis, indicating the robustness of the $\bar{R}'$ and $\bar{D}'$ method. The *M. frigidum* cluster is closer to the origin than in simulations. This is due to the lower coverage; *M. frigidum* was sequenced to ∼12.4× coverage. The coverage calculation used untrimmed reads and thus could be an overestimation. The length of the cluster, along its longest axis, is greater than that found in clonal clusters in the standard simulations (Figure 3.7). This is due to differences in the substitution error rates (Figure C.4F). This assembly shows that the simulations are applicable to experimentally-derived datasets, at least for a very simple case.

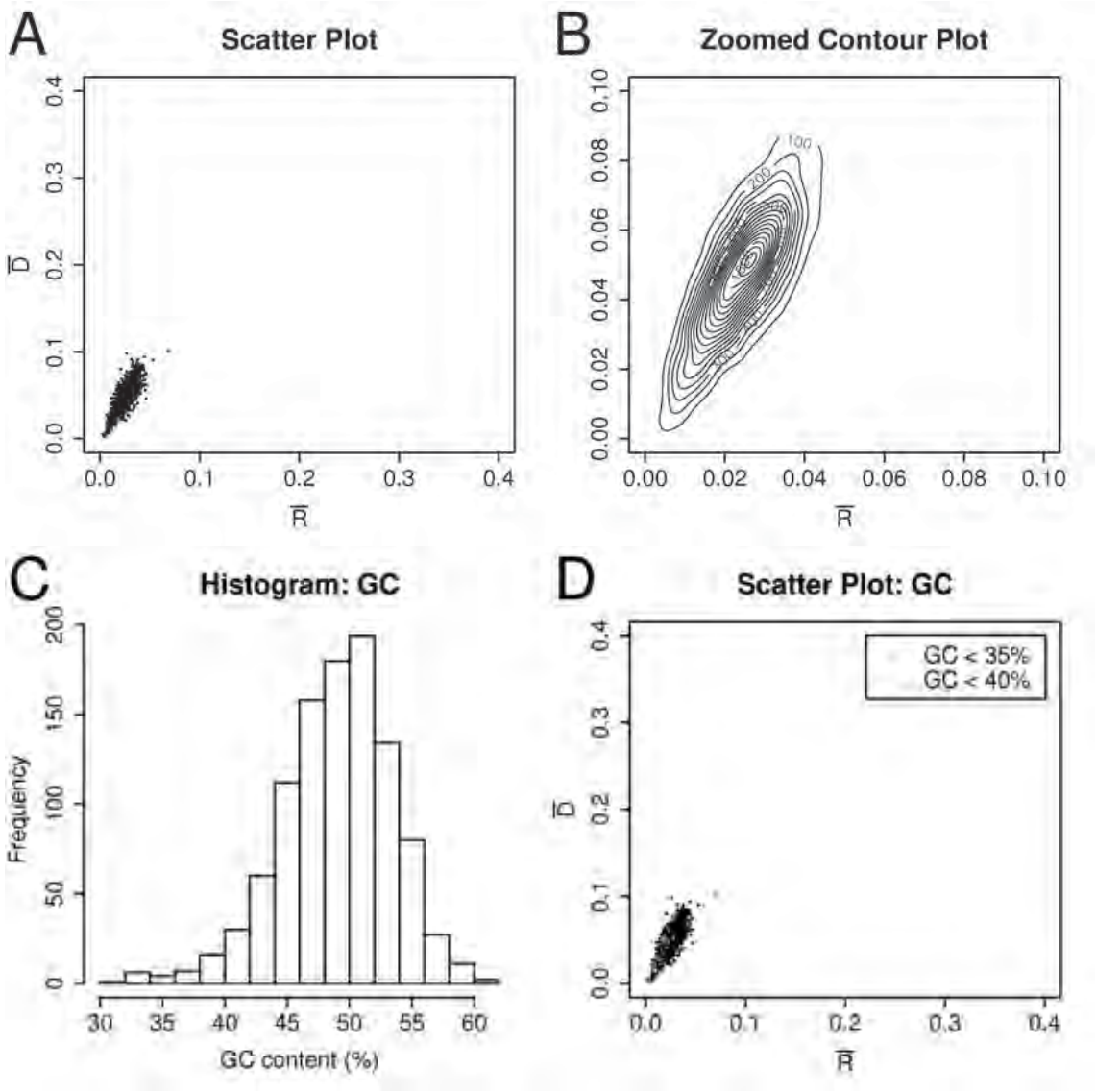

Figure 4.11: The *M. frigidum* assembly appears clonal despite a small amount of contamination.
A) Scatter plot of *M. frigidum* contigs.
B) Zoomed contour plot of *M. frigidum* contigs.
C) Histogram of *M. frigidum* GC content.
D) *M. frigidum* contigs with a GC content less than 35% are marked in red. Those with a GC content greater than or equal to 35% but less than 40% are marked in green.
$\overline{\boldsymbol{R}}$: reads per unit of contig length. $\overline{\boldsymbol{D}}$: discrepancies per unit of contig length.

4.4.5.2 Green Sulfur Bacteria

The Ace Lake, 12.7 m, 0.1 μm filter sample (ANTRC230_0.1) had been identified as having a large population of GSB. Histograms of the GC content of this sample showed at least two main peaks denoting at least two different species (Figure 4.12). Thus, the sample required filtering to isolate the GSB sequences. However, a preliminary analysis

of the unfiltered data can allow confirmation of results from the filtered data.

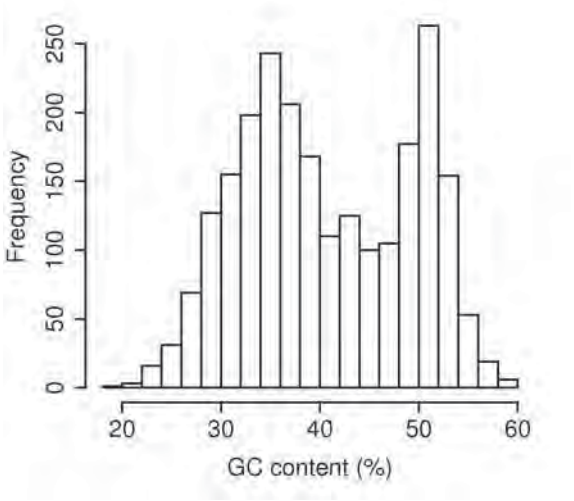

Figure 4.12: ANTRC230_0.1 contains at least two species.
Unfiltered ANTRC230_0.1 GC content histogram.

The scatter plot of the unfiltered ANTRC230_0.1 assembly has a cluster near the origin that resembles the clonal cluster in simulations (Figure 4.13A). It also has additional sparsely-clustered observations that correspond to the location of chimeric contigs. The structure of the plot does not match simulations as it is not divided into a line of distinct or overlapping clusters. This is because this data is an unfiltered environmental sample with sequence from multiple species. To determine if the scattered observations in the top right of the ANTRC230_0.1 plot were outliers, the lengths of contigs were investigated. Outliers in simulated assemblies were generally short, for example, all outliers in the 4% and 8% normalised two-strain assemblies (Figure C.1) were less than 7 kb. Therefore, given the distance of these contigs from the only distinct cluster it would be appropriate to remove these contigs if they were short. However, all 10 of the contigs with $\bar{R}$ values over 0.175 have lengths over 25 kb and four of these are over 100 kb. Therefore, these contigs are not outliers and they should not be rejected (Figure 4.13B). The likelihood of entire contigs over 100 kb in length being highly conserved sequence is low. There are additional long contigs between the putative clonal cluster and the putative final cluster. Additional information is required to determine the boundaries of the clusters that these points belong to. Evidence is also required to determine which of the two identified clusters and other long contigs are GSB.

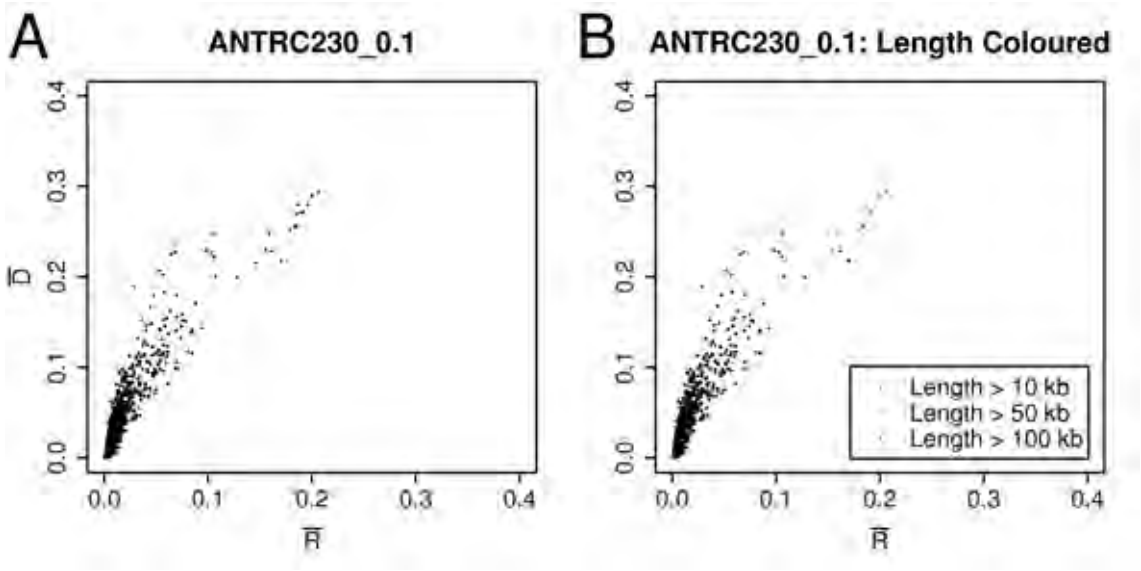

Figure 4.13: The contigs in ANTRC230_0.1 with the highest $\bar{R}$ values are too long to be outliers.
Unfiltered ANTRC230_0.1 assembly
A) Uncoloured
B) Long contigs coloured by contig length.
$\bar{R}$: reads per unit of contig length. $\bar{D}$: discrepancies per unit of contig length.

The unfiltered ANTRC230_0.1 scatter plot was coloured for GC content (Figure 4.14). The histogram of GC for the assembly shows two peaks at approximately 35% and 51% (Figure 4.12). All except four of the low GC unitigs (< 35%) appeared in the bottom cluster. All of these unitigs have an $\bar{R}'$ value less than 0.1. The high GC (> 50%) unitigs are located across the plot. Of 12 GSB species compared, *Prosthecochloris vibrioformis* DSM 265 was found to be the genome with the highest level of similarity to the Ace Lake GSB (Ng *et al.* 2010). *P. vibrioformis* strains have a GC content of 52.0% to 53.5% (Imhoff 2003), which provides evidence that the potential outliers are genuine and informative GSB unitigs that denote multiple strains assembled together (Figure 4.14). The GC content of the ANTRC230_0.1 GSB was also measured from the filtered Sanger assembly to be 52.2% (Ng *et al.* 2010). A high proportion of the contigs between the two identified clusters have a high GC content, including eight contigs over 10 kb.

Whilst this preliminary analysis of the unfiltered assembly identified two clusters with a GC content consistent with GSB, the plot is limited in its interpretability. GC content has not provided sufficient information to allow the boundaries of all clusters to be determined or to give sufficient evidence of which contigs are GSB. Filtering is

required to determine if there are any other clusters and to confirm that the identified clusters are GSB. This need is heightened because the structure of the plot does not match simulations.

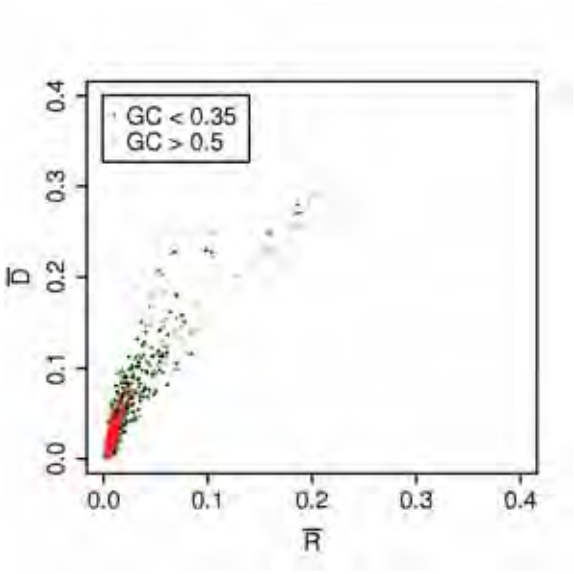

Figure 4.14: GC content suggests the presence of at least one clonal and one chimeric GSB cluster in ANTRC230_0.1.
Unfiltered ANTRC230_0.1 assembly coloured by GC content.
$\overline{\boldsymbol{R}}$: reads per unit of contig length. $\overline{\boldsymbol{D}}$: discrepancies per unit of contig length.

4.4.5.1 Oligonucleotide Frequency Filtering

The program *Tetra* (www.megx.net/tetra; Teeling *et al.* 2004a) was investigated in conjunction with GC content and *BLAST* (Basic Local Alignment Search Tool) to filter the ANTRC230_0.1 data. *Tetra* calculates di-, tri-, tetra- and penta-nucleotide frequencies in order to compare DNA sequences. The time taken to run *Tetra* on large datasets and limited access to Macintosh computers (Apple, Cupertino, CA, USA) meant that only unitigs greater than 2.5 kb were used. *P. vibrioformis* was used as a comparison.

To bin the hybrid assembly of 454 and Sanger reads, ROC plots of the di-, tri- and tetra-nucleotide frequencies; GC content and *BLAST* E-values against *P. vibrioformis* (Figure C.7) were produced. These were used to determine which measurement correlated most effectively with the previously filtered assembly of Sanger reads. Other ROC plots were made of the di-, tri- and tetra-nucleotide frequencies and GC content with *BLAST* hits used as the test data (Figure C.8).

From these plots, it was decided that di-nucleotide frequency (dimers) was the most

effective method for filtering the ANTRC230_0.1 dataset. This is because dimers produced the classifier with the highest AUC value against either test data set (Table C.6).

Due to the limited availability of dimer binning with *Tetra*, a custom script *tetra.py* was written to implement the core routine of this program, as described in Teeling *et al.* (2004a). However, since *Tetra* uses an undescribed method for dimer binning, the dimer scores for *tetra.py* used normalisation. The classifiers produced with these dimers achieved even higher AUCs of 0.9807 against *P. vibrioformis* and 0.9457 against *BLAST*. This gives extra support for the choice of dimers.

The filtered set of GSB unitigs suggests chimerism since there are unitigs of varying $\bar{R}'$ values including a strong cluster at higher values (Figure 4.15).

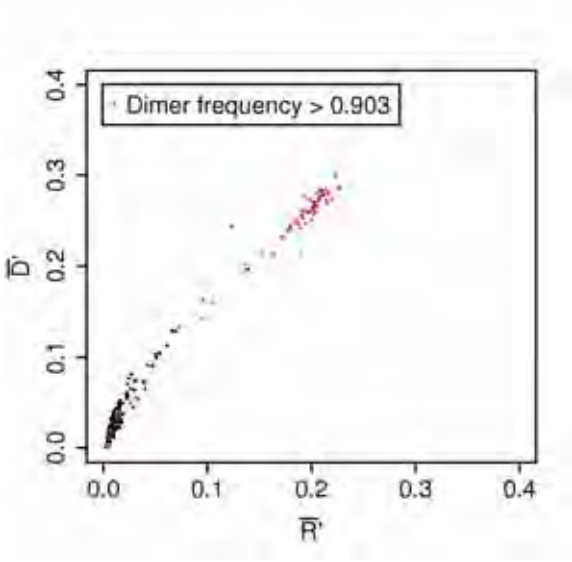

Figure 4.15: The filtered ANTRC230_0.1 assembly contains evidence of chimerism.
ANTRC230_0.1 assembly with filtered unitigs coloured, length > 2.5 kb.
$\bar{\boldsymbol{R}}'$: reads per unit of unitig length. $\bar{\boldsymbol{D}}'$: discrepancies per unit of unitig length.

As a confirmation of the filtering, the ANTRC230_0.1 unitigs were assigned to taxa using *MEGAN* (Figure 4.16). The filtered assembly contained 92% of the unitigs that *MEGAN* classified as Chlorobiaceae (GSB). The percentage of unitigs that *MEGAN* identified as GSB was enriched from 26% to 80% by the filtering. Only two unitigs that *MEGAN* had assigned to other phyla were retained by the filtering. Thus these *MEGAN* classifications provide corroboration that the dimer filtering has kept a very high percentage of the GSB unitigs and discarded the vast majority of everything else. Judged against *MEGAN*, the filtering achieved a TP of 91.8% and a FP of 7.95%. However, if false positives are restricted to those confidently classified as taxa outside

the lineage of GSB (i.e. Proteobacteria) then FP equals 1.14%. The classifications of "Not assigned" and "Bacteria" that *MEGAN* assigned to unitigs from the filtered assembly are not inconsistent with a classification of GSB.

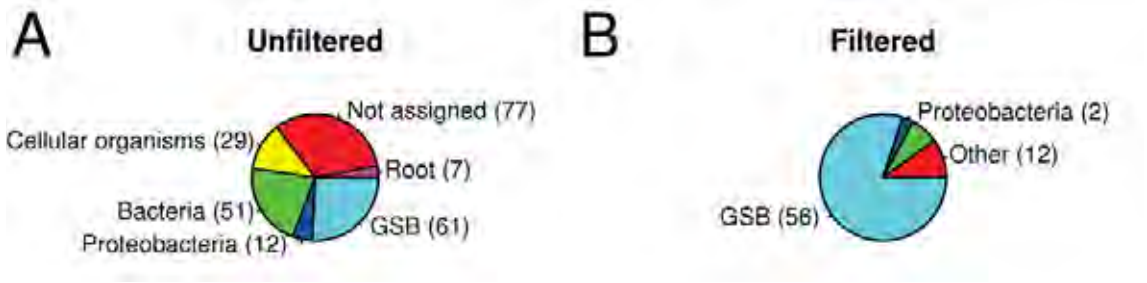

Figure 4.16: The dimer filtering process is corroborated by *MEGAN*.
ANTRC230_0.1 assembly *MEGAN* pie charts. Numbers in brackets indicate the number of unitigs assigned to that category by *MEGAN*.

MEGAN provides strong corroboration for the top cluster, assigning all but one unitig to GSB (Figure 4.17A). Four out of nine of the unitigs between the two distinct clusters were not corroborated by *MEGAN*. The other five of these unitigs are only loosely clustered. While only six of the unitigs in the bottom cluster were corroborated by *MEGAN*, four of the unitigs that *MEGAN* classified as GSB but which were not kept by the dimer filtering also fall in this cluster. This gives extra evidence for the bottom cluster representing at least one real GSB strain (Figure 4.17B).

Even if a unitig can be confidently identified as GSB, it could still be an outlier. Thus, unitigs were compared by length (Figure 4.17C). Only two of the unitigs between the two clusters is over 10 kb and only one of these is corroborated by *MEGAN*. This supports the assumption that these intervening unitigs are noise. Whilst only two of the unitigs in the bottom cluster are over 10 kb, these are both unitigs that were corroborated by *MEGAN*. The top cluster is even more confident since it has a 98% corroboration by *MEGAN* and it contains 46 unitigs, 12 of which are over 50 kb in length. Excluding the intervening unitigs, the simplest explanation is a single strain at low abundance and a single strain at high abundance which combine to extend the top cluster parallel to y = x. The bottom cluster has a similar length along its longest axis. This could be due to two strains in low abundance combining to extend this cluster. However, the top cluster is not long enough to support this theory. Also, if the unitig with the lowest $\bar{R}'$ value, which was assigned to Proteobacteria by *MEGAN*, was removed then this would slightly reduce this length.

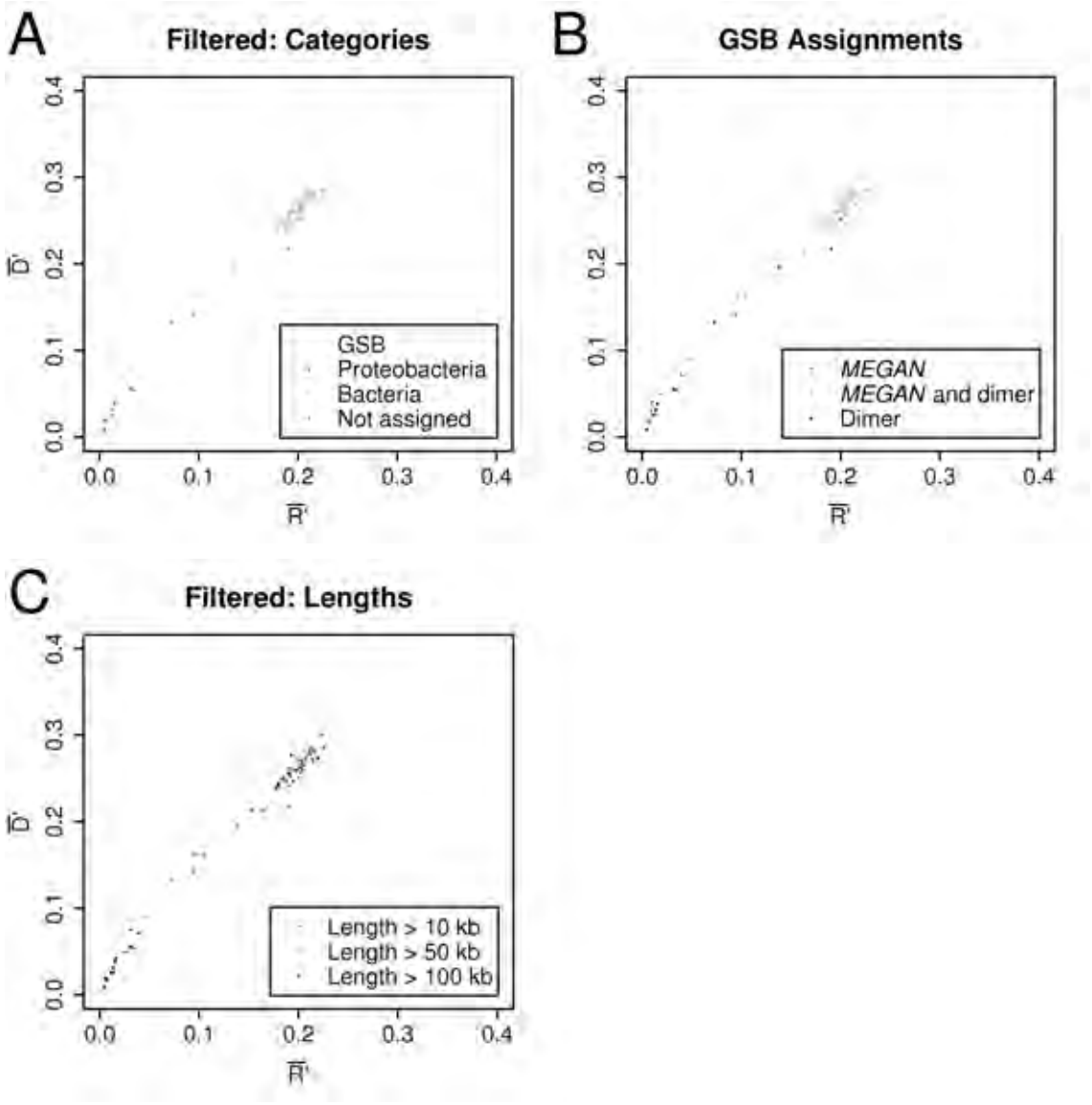

Figure 4.17: The first and last filtered ANTRC230_0.1 clusters are well supported, but the sparse unitigs in between are not.
A) ANTRC230_0.1 filtered assembly coloured by *MEGAN* classification.
B) ANTRC230_0.1 filtered assembly coloured by *MEGAN* GSB classification with additional data points from the unfiltered assembly that were classified as GSB by *MEGAN*.
C) ANTRC230_0.1 filtered assembly coloured by length.
$\overline{\boldsymbol{R}}'$: reads per unit of unitig length. $\overline{\boldsymbol{D}}'$: discrepancies per unit of unitig length.

Furthermore, the extended clusters may be due to the higher, more realistic levels of substitution errors as seen in Figure C.4. While the gradient of the cluster curve is not as steep as in these figures, it is still steeper than the standard plots (Figure 4.18). This suggests a slightly lower level of substitutions and thus a lower level of cluster extension than the assembly in Figure 4.18B. The peak near the origin in the bottom cluster and the plateau at the opposite end could suggest two clonal clusters and their two-strain cluster. However, the lack of corroboration for most of the unitigs in this

cluster makes this evidence unconvincing. The low cluster density in the bottom cluster is extra evidence against multiple strains in the bottom cluster (Figure 4.18A). In fact, the cluster density is not only lower than the two overlapping clonal clusters in Figure 4.18B, it is also lower than the single clonal cluster in Figure 4.18C. This low number of unitigs may be due to the very low read depths for this putative strain. The peak of this cluster has an $\bar{R}'$ value 14% of the value from the clonal cluster in the *S. aureus* assembly and 54% of the value from the *E. coli* assembly. At very low read depths, the number of unitigs assembled should be much lower. The top cluster is much more compact and uniform in density than the top clusters in the other assemblies. Given the lower read depth of the putative low abundance strain, the distance between the peaks of the overlapping clonal and two-strain clusters in the top cluster should be smaller. The high density of the top cluster and low density of the bottom cluster could also be due to a higher similarity between the two putative strains than between the strains in the other assemblies. A higher similarity would decrease the amount of reads in the clonal clusters and increase the amount in the two strain cluster. The top half of the top cluster has all the long unitigs which, combined with the low read depth of the putative low abundance assembly, would explain the uniformity of density in the top cluster if the two strains are highly similar.

If either of the weak clusters at ~0.1 or ~0.15 $\bar{R}'$ do depict a GSB strain then there should be at least one cluster at 0.3 $\bar{R}'$ or more (Figure 4.18A). The cluster at 0.2 $\bar{R}'$ cannot be explained as purely chimeric and according to Equation 4.4 there needs to be a two-strain cluster to accompany any two clonal clusters. According to this rule, no combination of the potential intervening clusters at 0.1 and 0.15 $\bar{R}'$ and the bottom cluster (~0.01 $\bar{R}'$) could explain the cluster at 0.2 $\bar{R}'$ as purely chimeric, i.e. 0.01 + 0.1 $\neq$ 0.2, 0.01 + 0.15 $\neq$ 0.2, 0.1 + 0.15 $\neq$ 0.2 and 0.01 + 0.1 + 0.15 $\neq$ 0.2. Since any intervening cluster or clusters should be accompanied by a cluster at 0.3 $\bar{R}'$ or more, which does not exist, the intervening unitigs can be safely excluded and the conclusions based on just the top and bottom clusters are well-supported by the available evidence.

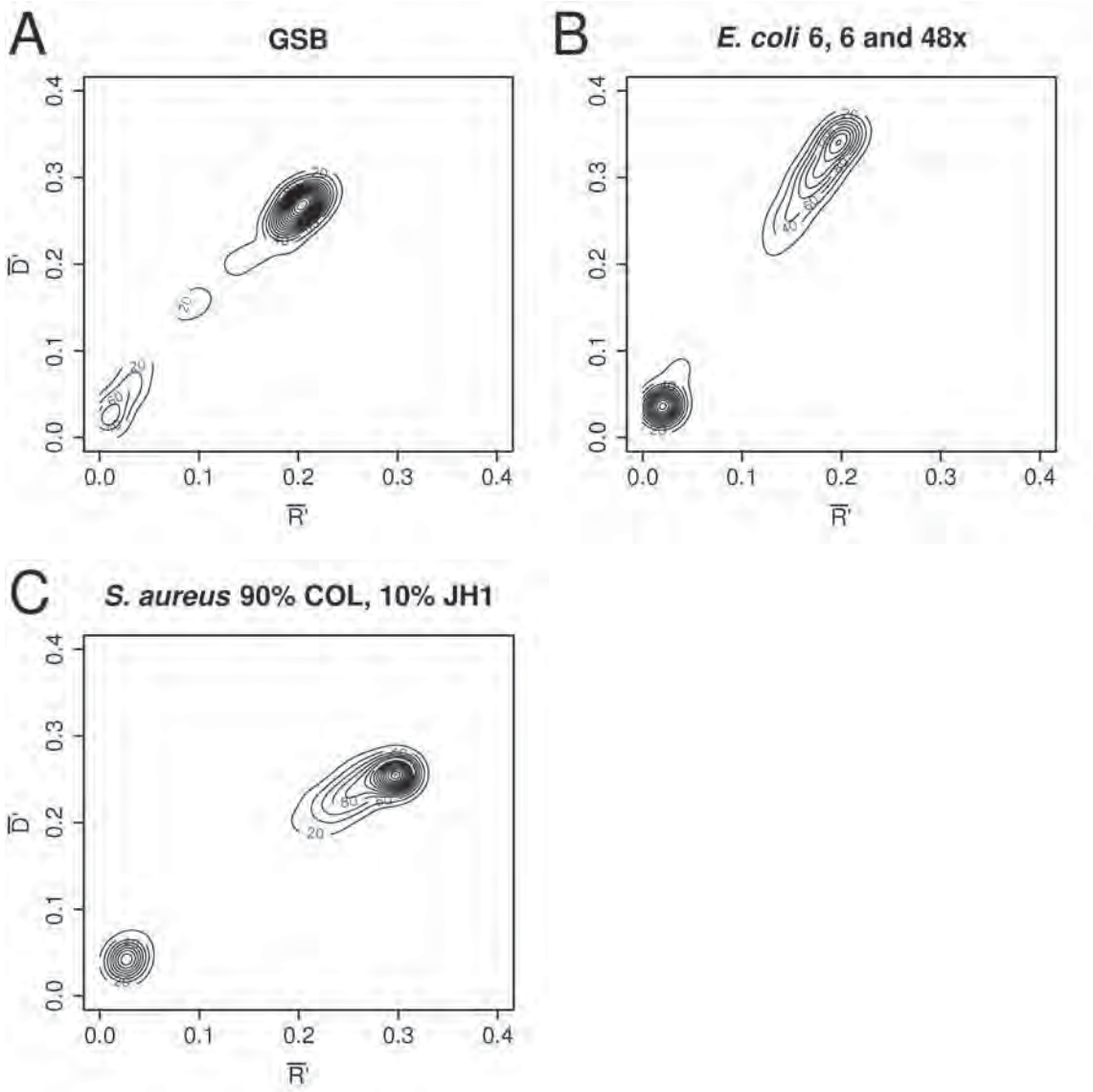

Figure 4.18: The well-supported filtered ANTRC230_0.1 clusters suggest one strain in low abundance and one strain with approximately nine times that abundance.
A) Contour plot of ANTRC230_0.1 filtered assembly.
B) Contour plot of 6× *E. coli* 55989, 6× *E. coli* APEC O1 and 48× *E. coli* ATCC 8739.
C) Contour plot of 90% *S. aureus* COL, 10% *S. aureus* JH1.
$\overline{\boldsymbol{R}}'$: reads per unit of unitig length. $\overline{\boldsymbol{D}}'$: discrepancies per unit of unitig length.

The low number of unitigs in the filtered ANTRC230_0.1 assembly and uncertainties of the filtering process both decrease the certainty with which this assembly could be analysed. However, *MEGAN* corroboration, unitig lengths, cluster positions and cluster density all provided evidence for the quantity and proportions of strains in this assembly. This interpretation is that there are two strains: one in high abundance (> 90%); and one in low abundance (< 10%). The high similarity of these strains (greater than the *E. coli* or *S. aureus* strains) was also inferred.

4.4.5.2 Pelagibacter

The Ace Lake, 11.5 m, 0.1 μm filter (ANTRC231_0.1) sample was filtered using dimers. *Pelagibacter ubique* HTCC1062 was used as the reference genome. The quantity of *Pelagibacter* strains is not apparent from the scatter plot, it may or may not be clonal (Figure 4.19).

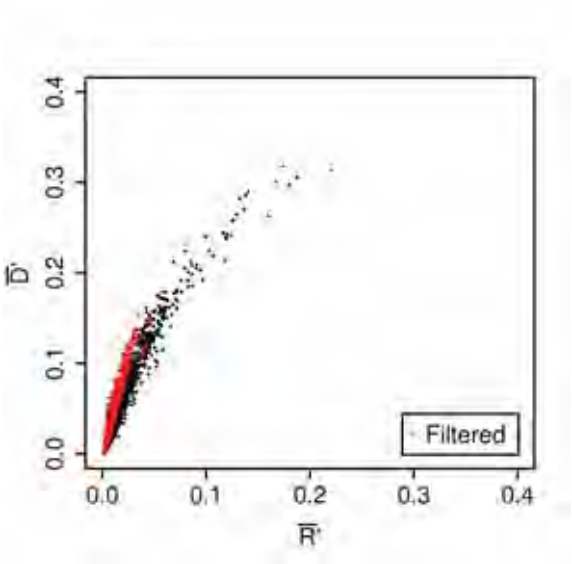

Figure 4.19: The structure of the strains is not apparent from the filtered ANTRC231_0.1 scatter plot.
ANTRC231_0.1 assembly with filtered unitigs coloured.
$\bar{R}'$: reads per unit of unitig length. $\bar{D}'$: discrepancies per unit of unitig length.

To investigate this structure, the filtered assembly was clustered with *Mclust* but the clustering was not reliable (Figure 4.20). The spacing and positioning of the clusters did not match the simulations.

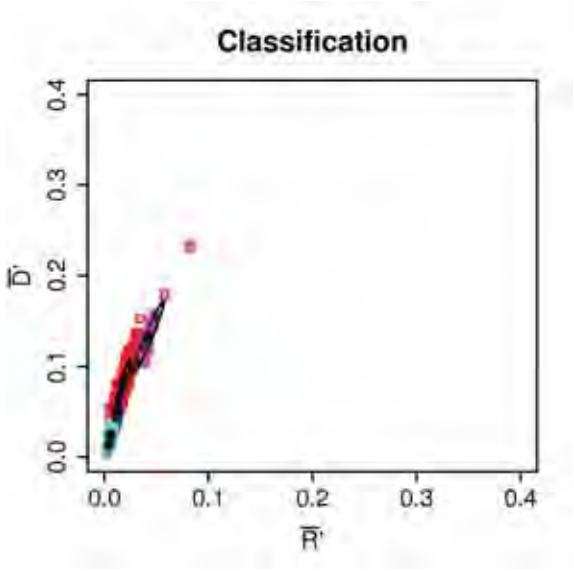

Figure 4.20: *Mclust* clustering of ANTRC231_0.1 did not match simulations.
Mclust clustering of filtered ANTRC231_0.1 assembly with default settings.
$\bar{R}'$: reads per unit of unitig length. $\bar{D}'$: discrepancies per unit of unitig length.

Thus, a contour plot of this data was produced at the standard 2.5 kb cut-off (Figure 4.21). The plot showed three clusters, though the third was quite faint. In simulations the final cluster was always strong, so this was investigated further by using a variety of length cut-offs to see whether this peak would disappear.

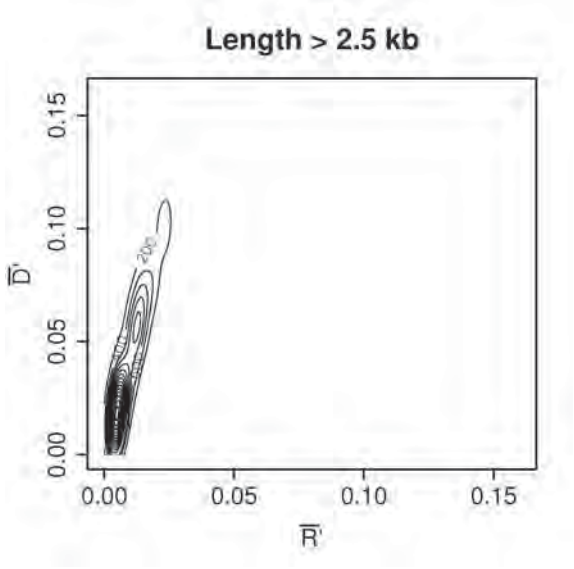

Figure 4.21: The third ANTRC231_0.1 cluster is weaker than in simulations.
Contour plot of filtered ANTRC231_0.1 assembly with only the unitigs longer than 2.5 kb.
$\bar{R}'$: reads per unit of unitig length. $\bar{D}'$: discrepancies per unit of unitig length.

Additional cut-offs of four, five, six, eight and 10 kb were used (Figure 4.22). This helped identify additional well-defined clusters consistently across different cut-offs, which had approximately even spacing.

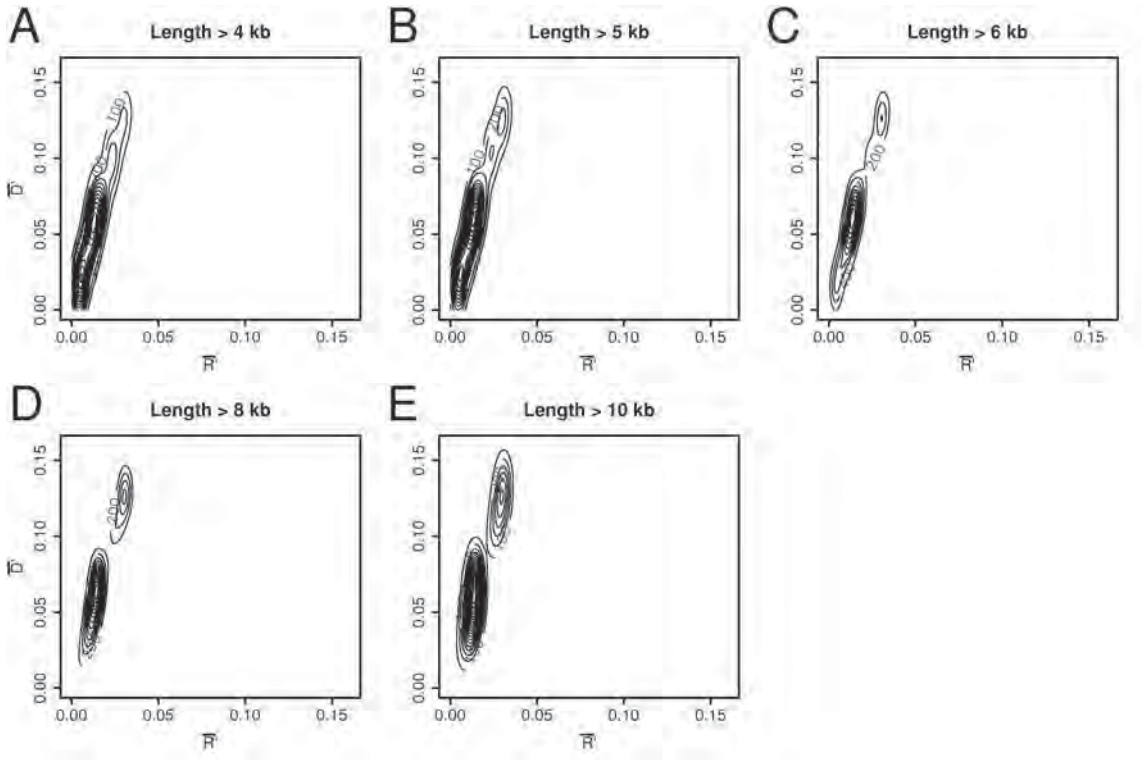

Figure 4.22: Additional ANTRC231_0.1 clusters are consistently placed across length cut-offs.
Contour plots of filtered ANTRC231_0.1 assemblies with varying length cut-offs.
$\bar{R}'$: reads per unit of unitig length. $\bar{D}'$: discrepancies per unit of unitig length.

The three confident clusters in this assembly's scatter plot require at least two strains for their explanation. Four clusters of approximately equal spacing could be explained by four strains in equal proportions. This interpretation is supported by the low density three-strain cluster which was observed in multiple simulations. However, this is not the most parsimonious explanation. The same clustering pattern could be observed with fewer strains. The data do not support more than four strains as this would require at least five clusters with non-coincident centres. Given the density of unitigs involved, confirmation of these unitigs by *MEGAN* is probably unnecessary but would be beneficial. The low density of the top cluster may be due to low similarity of the strains combined with the low unitigger error rate. The particular density pattern in this assembly decreases the certainty with which this assembly can be analysed.

4.5 Conclusion

Consistent rules relating the number of strains, and their proportions in an assembly, to the locations of clusters in a scatter plot have been developed. Whilst some cluster patterns are ambiguous, multiple methods for disambiguating these patterns have also been developed. Density prediction and error filtering could both be expanded on and this is discussed in Chapter 5. It has been possible to apply these findings to analyse multiple experimentally-derived data sets.

An automated program to apply this analysis to experimentally-derived data sets with quantitative certainty is beyond the scope of this project. However, a manual approach can tie together multiple strands of evidence to allow a well-supported and parsimonious interpretation of the number of strains in an assembly, their proportions, and even their similarity.

Chapter 5

Research Findings and Future Directions

5.1 Research Findings

In this thesis, new methods to study DNA microheterogeneity in environmental samples from microbial communities have been described. Specifically, methods to infer the quantity and proportions of strains in unitigs have been described.

For Sanger read simulation using *MetaSim*, all parameters, except one, could be calibrated to within 1% of the values in a real sequencing project (Table 2.1). This included parameters for controlling the types and rates of sequencing error rates. However, when the number of projects was increased to nine this was not possible because of the large variability in sequencing error rates in the available genome projects (Table 2.2). Neither the inconsistent project ages (Table 2.4 and Figure 2.4) or GC content (Table A.1) could explain this variability. Differences in coverage provided only a partial explanation (Table A.2). *MetaSim*'s Sanger error parameters had very linear response curves suggesting that calibration would have been simple given a consistent dataset to calibrate to (Figure 2.3). The available 454 data was more consistent than the Sanger data. However, the 454 error parameters in *MetaSim* had complex non-linear relationships with the quantity of errors added (Figure 2.5). There are also strong complex dependencies between these parameters. Thus, only the total quantity of errors was able to be calibrated (Figure 2.6).

A standard set of strains for each of the three species used in simulations was chosen, which kept variability in strain alignments and percent identities low (Table 3.1 to Table 3.4). Informative variables, $\bar{R}$ and $\bar{D}$, were chosen to allow microheterogeneity in metagenomic samples to be graphically analysed. Length filtering and normalising these variables by length resulted in more informative scatter plots (Figure 3.6). This choice of variables was supported by a moderately high correlation (Figure 3.4). The amount of information contained in different potential variables could be compared due to the tracking of the strain of origin of each read, in these simulations (Figure 3.5 and

Figure 3.6C). The quantity of distinct clusters in one-, two- and three-strain equal-proportion assemblies was equal to the number of strains (Figure 3.7). The positioning of one- and two-strain clusters also matched across these assemblies. This graphical analysis method was independent of read depth over an order of magnitude range (Figure 3.10). It was also robust over changes to assembly tolerances (Figure 3.11D to H and Figure 3.12) and species (Figure 3.7B, Figure 3.8A and Figure 3.10J). Using the same coverage in different species made the corresponding clusters align more effectively (Figure 3.15). Methods were developed for robustly binning unitigs according to the strains of origin of their reads (Equation 3.1). This allowed accurate predictions of the clonality of unitigs in one-, two- and three-strain assemblies to be made (Figure 3.16 and Figure 3.17). An accurate prediction was also made using assemblies from different species for the training and test data (Figure 3.18).

Model based clustering of scatter plots, using *Mclust*, gave mixed results. A four-strain assembly was clustered well except with unevenly spaced cluster centres (Figure 4.1). A two-strain assembly was assigned two clusters too many (Figure 4.2). To improve the clustering, *MclustDA* used training data which implicitly contained information about cluster spacing. To create this training data, the unitig binning method was expanded for use with assemblies with strains in unequal proportions (Equation 4.1). However, even when very similar training data with finetuned binning was used, *MclustDA* did not improve on *Mclust*'s binning (Subsection 4.4.2.3). *Mclust* could be used for detecting outliers (Figure 4.3). Outliers in the plots were similar across species and assembly settings (Table C.1 and Table C.2). This information was used to improve the training data.

Strong linear relationships were found between the locations of clusters in scatter plots and strain proportions (Figure 4.7, Figure C.5). To predict cluster locations given strain read depths, rules were developed that allow prediction of strain quantities and proportions in assemblies with low strain complexity (Equation 4.4). Contour plots clustered equal proportion assemblies well (Figure 4.8). Thus, a custom peak picking tool that used the same kernel density estimates as the contour plots was developed. Patterns in the densities of clusters allowed predictions of these densities to be made (Figure 4.9 and Table 4.3), which helped to improve visual analysis. Scatter plots of idealised assemblies with zero sequencing errors showed a stratification of clusters with different strain counts (Figure 4.10).

The *M. frigidum* genome was tested and correctly evaluated as clonal (Figure 4.11). This provided basic verification of the methodology. Removing the known bacterial contamination from *M. frigidum* data set did not change the analysis, indicating that the method was robust. The Antarctic metagenome sample, ANTRC230_0.1, which contains a high level of coverage for GSB was found to have at least two different species (Figure 4.12). Both contigs in the presumably clonal cluster and the potentially chimeric cluster had a GC content close to that of GSB (Figure 4.14). All of the contigs with high $\bar{R}$ values in the second cluster had reads over 25 kb and thus are unlikely to be outliers (Figure 4.13B). This gives strong evidence that the GSB is not entirely clonal.

Different oligonucleotide frequencies were compared as methods of filtering out the GSB sequences from the hybrid assembly. This was done using two different test sets: one based on a filtered Sanger assembly and the other using *BLAST* e-values (Figure C.7 and Figure C.8). Dimers used with the Sanger-assembly-based test data were found to classify the data most effectively (Table C.6). Due to the limited platform independence of *Tetra* and lack of command line interface, *tetra.py* was written to implement the core routine in *Tetra*. However, *Tetra* used an undescribed method for dimer frequency calculations rather than the z-score method described in the accompanying paper.

The dimer filtering was corroborated by *MEGAN* (Figure 4.16). *MEGAN* assignments combined with unitig lengths gave strong support that the top and bottom clusters in the filtered ANTRC230_0.1 scatter plot represented GSB (Figure 4.17). The intervening points lacked support. These clusters suggest that there is one GSB strain in low abundance and one with approximately nine times that abundance (Figure 4.18).

For the ANTRC231_0.1 dimer filtered data set, *Mclust* clustering was not reliable because the spacing and positioning of the clusters did not match the simulations (Figure 4.20). A contour plot showed a weak final cluster compared to the strong final clusters in simulations (Figure 4.21). Thus contour plots were produced with a variety of length cut-offs. This helped identify four well-defined evenly-spaced clusters consistently across the different cut-offs (Figure 4.22). The clustering pattern observed can be explained with two to four strains. The low density of the third cluster is supportive of four strains.

Analysis of the ANTRC230_0.1 and ANTRC231_0.1 environmental samples found evidence that the complexity of these samples was within the usable range of the $\bar{R}'$ and

$\overline{D}'$ graphical analysis method. This method was able to make well supported conclusions about the number and proportions of strains in these samples.

5.2 Future Directions

The $\bar{R}'$ and $\overline{D}'$ graphical analysis techniques presented in this thesis are currently limited by the strain complexity of the metagenomic samples that can be analysed. Whilst rules for predicting cluster location have been formulated, additional techniques such as cluster density prediction are not sufficiently developed for automation. Ideally, this work should be extended to allow automated analysis of metagenomic samples of any microbial species regardless of the number or proportions of strains, given sufficient read depth. There appear to be good avenues for improving these techniques to allow for both the analysis of samples with higher strain counts and the automation of the process, some of which are discussed below.

5.2.1 Outlier filtering

In the scatter plots of simulated metagenomic assemblies, unitigs with excessive read depth were found for clusters in assemblies of different species and across the range of strain counts (Figure 3.16, Figure 4.3 and Figure C.1). It may be possible to filter out these outliers in a more sophisticated manner than used here. Many outliers for the completely chimeric clusters were found to map to outliers in assemblies with different settings (Table C.1 and Table C.2). There were also similarities between the genes found in these outliers across species (Table C.1 and Table C.2). This suggests that a filter based on parts of the genome that are known to be repetitive may be able to accurately remove many of the outliers from each cluster. This could allow a more accurate analysis as the cluster boundaries would be more sharply defined. However, sequence based filtering may not be feasible for genomes from poorly-studied taxa. This is because an annotated genome from a close relative may be necessary. Alternatively, it may also be possible to use some of the repeat detection from within the assembly process to help identify repeats. However, this would not be appropriate if this relied too heavily on read depth as a detection method.

5.2.2 Assembler Dependence

All of the assemblies analysed in this thesis were produced using the *Celera assembler*. However, the work could be extended to utilise other assemblers. If the $\bar{R}'$ and $\bar{D}'$ graphical analysis was automated, it would ideally be made assembler-independent. The ability to use contigs instead of unitigs would help facilitate this. The unitig features $\bar{R}'$ and $\bar{D}'$ were used instead of their contig counterparts due to reasons that would not apply to all assemblers. When a unitig is labelled as a surrogate, the *Celera assembler* includes the consensus sequence of the unitig in a contig but not the associated reads. This led to very low $\bar{R}$ values being reported for numerous long informative contigs. These reasons are likely to be absent in other assemblers that do not use unitigs, and thus contigs could be used in these cases. Since contigs are often longer than unitigs and $\bar{R}$ and $\bar{D}$ are normalised by length, they make more accurate data points. However, unitigs are more numerous than contigs. A lower quantity of more accurate points would work most effectively with weighted values so that the longer more accurate points have a greater influence on clustering than shorter less accurate points.

5.2.3 Resolving Cluster Overlaps

The method used for estimating cluster peak heights was not precise and would not be suitable for automation. However, other methods may allow automated quantitative results. For example, the shapes of the density curves for individual clusters could be used to predict these shapes for overlapping clusters. One method of doing this would be Gaussian deconvolution. Such a technique may allow the $\bar{R}'$ and $\bar{D}'$ graphical analysis developed in this work to make predictions on the quantity and abundance of strains in significantly more complex metagenomic samples. If discrepancy filtering was used to at least partially separate the clusters, then this technique would be even more informative.

5.2.4 Validation with a well-studied, low-complexity metagenome

To validate the $\bar{R}'$ and $\bar{D}'$ graphical analysis method, a low complexity metagenome from an acid mine drainage (Tyson *et al.* 2004b) was investigated. This dataset has a very dominant species (75%) with low polymorphism, and has been rigorously studied.

However, a preliminary characterisation of this dataset found that whilst the data was available in an appropriate format, the descriptions of the microheterogeneity were not sufficient to allow a meaningful comparison. Similarly, the description of the overall structure of the dataset was not adequately detailed, and the provided data on variation at specific loci was not generalisable. Thus, whilst this may have been a useful dataset for validation, this was unfortunately not possible. If a dataset of similar complexity became available, with a global analysis of the number of strains present, their proportions, and their percent identities, then this would provide a much better validation set for the $\bar{R}'$ and $\bar{D}'$ graphical analysis.

5.2.5 Variability of Genomic Divergence

The degree of variability of genomic divergence between related strains is not always constant throughout the genome. For example, Didelot *et al.* (2007) found a bimodal pattern of divergence between two *Salmonella enterica* strains which appears to be due to many recombination events between distantly related strains. Whilst this is an extreme example, recombination even on a smaller scale could decrease the accuracy of the $\bar{R}'$ and $\bar{D}'$ graphical analysis. It is likely that some level of recombination has occurred in the datasets analysed and has made a significant contribution to the inaccuracies in the results. If the effect of this variable divergence could be compensated for, then the accuracy of the analysis could be significantly improved. Tools exist that help detect recombination events in assemblies [e.g. (Eppley *et al.* 2007)] and these may help achieve this improvement.

The genomic divergence between related strains is further complicated by differences in the rate at which these mutations occur and the rate at which they are fixed within the population. Recombination plays an important role in how these mutations are accumulated. Some work has been done to estimate some of these rates (Johnson and Slatkin 2009), so it may be possible to utilise this information to improve the $\bar{R}'$ and $\bar{D}'$ graphical analysis.

5.2.6 Idealised Assemblies and Discrepancy Filtering

Discrepancy filtering is one area that could lead to improvement in these techniques. In zero-sequencing-error simulated assemblies there is extra information about cluster

chimerism on the $\overline{D}'$ axis (Figure 4.10). In these assemblies, clusters are stratified according to their strain counts, and clusters with more strains have higher $\overline{D}'$ values. Thus, clusters with different strain counts that would otherwise overlap should only partially overlap if at all. This in turn means that decreasing error rates could give increased cluster resolution.

These assemblies had low quantities of data points for clonal clusters due to excessive assembly, which decreased the definition of the clusters. This could be rectified by choosing appropriate assembly parameters, i.e. lower unitigger error rates.

Discrepancy filtering could allow increased cluster resolution for assemblies of samples from real microbial communities. The sequencing errors could be filtered by changing how the $\overline{D}'$ values are calculated. This would not require the removal of any reads from the assemblies and thus the $\overline{R}'$ values would not be affected. For example, the program *analyzeSNPs* can restrict which discrepancies are reported. Minimum values can be set for the number of consistent disagreeing reads, conflicting cumulative quality values and conflicting quality values. This could be combined with appropriate assembly parameters. Such an approach could separate clusters with different numbers of strains and thus allow easier interpretation of the scatter plots. This in turn could allow accurate predictions of the quantity and proportion of strains in assemblies of data from more complex microbial communities.

5.2.7 Clustering with Improved Models

Model based clustering of scatter plots, using *Mclust*, gave mixed results (Figure 4.1 and Figure 4.2). Relationships in the spacing of clusters were identified that could not be specified by the provided normal mixture models. If these relationships were incorporated into a model, the clustering may have been improved. Such changes would have required modification of the *MCLUST* codebase, which was not possible due to time constraints.

The relationships that were originally identified were based on the uniform spacing of clusters that only applies to equal proportion assemblies. However, there are additional relationships that could be incorporated into a clustering model for unequal proportion assemblies, such as the rules developed for cluster locations.

These rules could be incorporated by performing a search through different possible

cluster combinations to see which most effectively fits the data. This approach should be more feasible when combined with discrepancy filtering. With this filtering, candidate clonal clusters could be located first by searching a band of the lowest $\bar{D}'$ values (e.g. those less than 0.01 $\bar{D}'$). This would be followed by candidate two-strain clusters being located in the next band (e.g. 0.01 – 0.02 $\bar{D}'$), three-strain clusters in a third band (e.g. 0.02 – 0.03 $\bar{D}'$) and so on. Different interpretations of the evidence for clonal clusters could be evaluated based on the evidence for two-strain clusters. The potential two-strain clusters could in turn be evaluated according to the evidence for three-strain clusters and so on. Thus, the best supported interpretation of the data would be derived during the clustering process.

For example, consider an analysis of a three-strain *E. coli* 12, 18, 30× assembly with zero sequencing errors (Figure 5.1). In this scatter plot, there are obvious clonal clusters at approximately 0.04 and 0.06 $\bar{R}'$. At 0.1 $\bar{R}'$, the clonal cluster blends into the two-strain cluster but still has enough points in line with the other clonal clusters to be well supported. The low $\bar{D}'$ ends of two-strain clusters at 0.14 and 0.16 $\bar{R}'$ would be less probable clonal clusters. Given these potential clonal clusters, the potential two-strain clusters can be used for validation. The clear two-strain cluster at 0.1 validates the clonal clusters at 0.04 and 0.06 $\bar{R}'$. The two-strain clusters at 0.14 and 0.16 $\bar{R}'$ can be easily identified as at least one two-strain cluster. These two-strain clusters and the 0.1 $\bar{R}'$ clonal cluster can be validated because they fit with each other and with the previously validated clusters. The three-strain cluster, at 0.2 $\bar{R}'$, provides further confirmation for all six of these clusters. The three-strain cluster could appear as evidence for the further two-strain clusters needed to validate a clonal cluster at 0.14 or 0.16 $\bar{R}'$. However, since there are no clusters after the three-strain cluster, additional two-strain clusters cannot be validated and therefore a fourth clonal cluster cannot be validated.

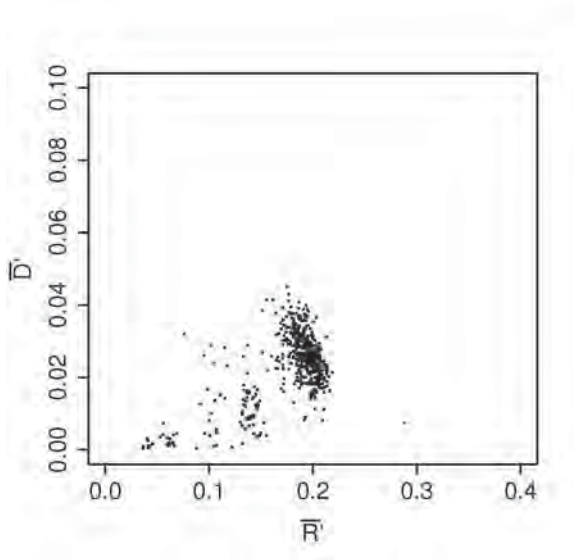

Figure 5.1: Removing sequencing errors could provide sufficient cluster structure for automated analysis.
Three-strain *E. coli* 12, 18, 30× assembly with zero sequencing errors.
$\bar{R}'$: reads per unit of unitig length. $\bar{D}'$: discrepancies per unit of unitig length.

Sequencing error rate affects the angle of the clusters on the plots, i.e. more errors gives a steeper gradient (Figure C.4). Filtering of discrepancies would not remove all errors, meaning that the clusters would be on a slight incline. However, the bands used in the analysis could be rotated about the origin. This would position the bands correctly to allow location of the clusters of each level of chimerism.

It may also be possible to weight unitigs by length, so that longer, more informative unitigs have a greater influence on the clustering. This would also remove the need for strict length filtering, allowing more data points without loss of clarity. This weighting would have a similar effect to the method used to analyse the *Pelagibacter* data (Figure 4.21 and Figure 4.22).

By adding this information, it may be possible to create a model that greatly improves *Mclust*'s ability to fit it to real data. This might then make it possible to eliminate the need for a graphical interpretation, which would substantially increase the quantity of samples that could be analysed. Accurate assignment of clusters to their chimerism-bins would allow the number and proportion of strains in a metagenomic sample to be known. The contigs from each cluster could be analysed separately to determine which genes are unique to each strain and which are shared by the different combinations of strains. This could allow the ecological roles of the different strains to be understood. The understanding of these roles could be further developed by analysing time series data or comparing samples of the same species over a longitudinal

or latitudinal range. This would allow conclusions about how the species is evolving or how it has adapted to different environments.

References

Allen EE, Tyson GW, Whitaker RJ, Detter JC, Richardson PM, Banfield JF: **Genome dynamics in a natural archaeal population**. *Proceedings of the National Academy of Sciences* 2007, **104**(6):1883.

Allison PD: **Logistic regression using the SAS system: theory and application**: SAS Publishing; 1999.

Aparicio S, Chapman J, Stupka E *et al.*: **Whole-genome shotgun assembly and analysis of the genome of *Fugu rubripes***. *Science* 2002, **297**(5585):1301-1310.

Balzer S, Malde K, Lanzén A, Sharma A, Jonassen I: **Characteristics of 454 pyrosequencing data—enabling realistic simulation with flowsim**. *Bioinformatics* 2010, **26**(18):i420-i425.

Balzer S, Malde K, Lanzén A, Sharma A, Jonassen I: **Characteristics of 454 pyrosequencing data—enabling realistic simulation with flowsim**. *Bioinformatics* 2011, **27**(15):2171.

Béjà O: **To BAC or not to BAC: marine ecogenomics**. *Curr Opin Biotechnol* 2004, **15**(3):187-190.

Bentley SD, Vernikos GS, Snyder LAS *et al.*: **Meningococcal genetic variation mechanisms viewed through comparative analysis of serogroup C strain FAM18**. *PLoS Genet* 2007, **3**(2):e23.

Blattner FR, Plunkett G, Bloch CA *et al.*: **The complete genome sequence of *Escherichia coli* K-12**. *Science* 1997, **277**(5331):1453-1462.

Brockman W, Alvarez P, Young S *et al.*: **Quality scores and SNP detection in sequencing-by-synthesis systems**. *Genome research* 2008, **18**(5):763-770.

Brzuszkiewicz E, Brüggemann H, Liesegang H *et al.*: **How to become a uropathogen: comparative genomic analysis of extraintestinal pathogenic *Escherichia coli* strains**. *Proceedings of the National Academy of Sciences* 2006, **103**(34):12879-12884.

Chou H-H, Holmes MH: **DNA sequence quality trimming and vector removal**. *Bioinformatics* 2001, **17**(12):1093-1104.

Cock PJ, Antao T, Chang JT *et al.*: **Biopython: freely available Python tools for computational molecular biology and bioinformatics**. *Bioinformatics* 2009, **25**(11):1422-1423.

Coleman ML, Sullivan MB, Martiny AC *et al.*: **Genomic islands and the ecology and evolution of Prochlorococcus**. *Science* 2006, **311**(5768):1768.

Copeland A, Lucas S, Lapidus *et al.*: **Complete sequence of *Thermotoga* sp. RQ2**. Unpublished data; 2008a.

Copeland A, Lucas S, Lapidus A *et al.*: **Complete sequence of chromosome of *geothermalis* DSM 11300**. Unpublished Data; 2006.

Copeland A, Lucas S, Lapidus A *et al.*: **Complete sequence of chromosome of *Staphylococcus aureus* subsp. *aureus* JH1**. Unpublished data; 2007.

Copeland A, Lucas S, Lapidus A *et al.*: **Complete sequence of *Anabaena variabilis* ATCC 29413**. Unpublished Data; 2005.

Copeland A, Lucas S, Lapidus A *et al.*: **Complete sequence of *Escherichia coli* C str. ATCC 8739**. Unpublished data; 2008b.

Deloger M, El Karoui M, Petit M-A: **A Genomic Distance Based on MUM Indicates Discontinuity between Most Bacterial Species and Genera**. *J Bacteriol* 2009, **191**(1):91-99.

Denisov G, Walenz B, Halpern AL *et al.*: **Consensus generation and variant detection by Celera Assembler**. *Bioinformatics* 2008, **24**(8):1035-1040.

Didelot X, Achtman M, Parkhill J, Thomson NR, Falush D: **A bimodal pattern of relatedness between the Salmonella Paratyphi A and Typhi genomes: Convergence or divergence by homologous recombination?** *Genome research* 2007, **17**(1):61-68.

Dijkshoorn L, Ursing BM, Ursing JB: **Strain, clone and species: comments on three basic concepts of bacteriology**. *Journal of Medical Microbiology* 2000, **49**(5):397-401.

Dybvig K, Zuhua C, Lao P *et al.*: **Genome of *Mycoplasma arthritidis***. *Infect Immun* 2008, **76**(9):4000-4008.

Eppley JM, Tyson GW, Getz WM, Banfield JF: **Strainer: software for analysis of population variation in community genomic datasets**. *BMC Bioinformatics* 2007, **8**:398.

Ewing B, Green P: **Base-calling of automated sequencer traces using Phred. II. Error probabilities**. *Genome Res* 1998, **8**(3):186-194.

Fawcett T: **ROC graphs: notes and practical considerations for researchers** [home.comcast.net/~tom.fawcett/public_html/papers/ROC101.pdf] 2004

Fraley C, Raftery AE: **Model-based clustering, discriminant analysis, and density estimation**. *J Am Stat Assoc* 2002, **97**(458):611-631.

Fraley C, Raftery AE: **MCLUST version 3 for R: normal mixture modeling and model-based clustering**. Department of Statistics, University of Washington; 2006.

Franzmann PD, Liu Y, Balkwill DL, Aldrich HC, Conway De Macario E, Boone DR: ***Methanogenium frigidum* sp. nov., a psychrophilic, H2-using methanogen from Ace Lake, Antarctica**. *Int J Syst Bacteriol* 1997, **47**(4):1068-1072.

Fraser-Liggett CM, Mongodin EF, Casjens B *et al.* Unpublished data; 2008.

Gill SR, Fouts DE, Archer GL *et al.*: **Insights on evolution of virulence and resistance from the complete genome analysis of an early methicillin-resistant *Staphylococcus aureus* strain and a biofilm-producing methicillin-resistant *Staphylococcus epidermidis* strain**. *J Bacteriol* 2005, **187**(7):2426-2438.

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**(1):81-91.

Handelsman J: **Metagenomics: application of genomics to uncultured microorganisms**. *Microbiol Mol Biol Rev* 2004, **68**(4):669-685.

Hirao I, Nishimura Y, Tagawa Y-i, Watanabe K, Miura K-i: **Extraordinarily stable mini-hairpins: electrophoretical and thermal properties of the various sequence variants of d(GCFAAAGC) and their effect on DNA sequencing**. *Nucleic Acids Res* 1992, **20**(15):3891-3896.

Imhoff JF: **Phylogenetic taxonomy of the family Chlorobiaceae on the basis of 16S rRNA and fmo (Fenna-Matthews-Olson protein) gene sequences**. *Int J Syst Evol Microbiol* 2003, **53**(4):941-951.

Johnson PLF, Slatkin M: **Inference of Microbial Recombination Rates from Metagenomic Data**. *Plos Genetics* 2009, **5**(10).

Johnson TJ, Kariyawasam S, Wannemuehler Y *et al.*: **The genome sequence of avian pathogenic *Escherichia coli* Strain O1:K1:H7 shares strong similarities with human extraintestinal pathogenic *E. coli* genomes**. *J Bacteriol* 2007, **189**(8):3228-3236.

Kalyuzhnaya M, Lapidus A, Ivanova N *et al.*: **High-resolution metagenomics targets specific functional types in complex microbial communities**. *Nature biotechnology* 2008, **26**(9):1029-1034.

Kellis M, Patterson N, Endrizzi M, Birren B, Lander ES: **Sequencing and comparison of yeast species to identify genes and regulatory elements**. *Nature* 2003, **423**(6937):241-254.

Kettler GC, Martiny AC, Huang K *et al.*: **Patterns and implications of gene gain and loss in the evolution of *Prochlorococcus***. *PLoS Genet* 2007, **3**(12):e231.

Konstantinidis KT, Ramette A, Tiedje JM: **The bacterial species definition in the genomic era**. *Philosophical Transactions of the Royal Society B: Biological Sciences* 2006, **361**(1475):1929.

Kunin V, Copeland A, Lapidus A, Mavromatis K, Hugenholtz P: **A bioinformatician's guide to metagenomics**. *Microbiol Mol Biol Rev* 2008, **72**(4):557-578.

Kurtz S, Phillippy A, Delcher AL *et al.*: **Versatile and open software for comparing large genomes**. *Genome Biol* 2004, **5**(2):R12.

Lauro FM, DeMaere MZ, Yau S *et al.*: **An integrative study of a meromictic lake ecosystem in Antarctica**. *ISME J* 2010, **5**:879–895.

Lewis I, Schommer S, Markley J: **rNMR: open source software for identifying and quantifying metabolites in NMR spectra**. *Magn Reson Chem* 2009, **47**(Suppl S1):S123-S126.

Lucas S, Copeland A, Lapidus A *et al.*: **Complete sequence of *Hydrogenobaculum* sp. Y04AAS1**. Unpublished Data; 2008a.

Lucas S, Copeland A, Lapidus A *et al.*: **Complete sequence of *Dictyoglomus turgidum* DSM 6724**. Unpublished data; 2008b.

Mardis ER: **The impact of next-generation sequencing technology on genetics**. *Trends Genet* 2008, **24**(3):133-141.

Margulies M, Egholm M, Altman WE *et al.*: **Genome sequencing in microfabricated high-density picolitre reactors**. *Nature* 2005, **437**(7057):376-380.

Mavromatis K, Ivanova N, Barry K *et al.*: **Use of simulated data sets to evaluate the fidelity of metagenomic processing methods**. *Nat Methods* 2007, **4**(6):495-500.

Miller TR, Delcher AL, Salzberg SL, Saunders E, Detter JC, Halden RU: **Genome sequence of the dioxin-mineralizing bacterium *Sphingomonas wittichii* RW1**. *J Bacteriol* 2010, **192**(22):6101-6102.

Mitra S, Klar B, Huson DH: **Visual and statistical comparison of metagenomes**. *Bioinformatics* 2009, **25**(15):1849-1855.

Myers EW, Sutton GG, Delcher AL *et al.*: **A whole-genome assembly of *Drosophila***. *Science* 2000, **287**(5461):2196-2204.

Ng C, DeMaere MZ, Williams TJ *et al.*: **Metaproteogenomic analysis of a dominant green sulfur bacterium from Ace Lake, Antarctica**. *ISME J* 2010, **4**:1002–1019.

Parkhill J, Achtman M, James KD *et al.*: **Complete DNA sequence of a serogroup A strain of Neisseria meningitidis Z2491**. *Nature* 2000, **404**(6777):502-506.

Peng J, Yang L, Yang F *et al.*: **Characterization of ST-4821 complex, a unique *Neisseria meningitidis* clone**. *Genomics* 2008, **91**(1):78-87.

Petrosino JF, Highlander S, Luna RA, Gibbs RA, Versalovic J: **Metagenomic**

Pyrosequencing and Microbial Identification. *Clin Chem* 2009, **55**(5):856-866.

Phillippy AM, Schatz MC, Pop M: **Genome assembly forensics: finding the elusive mis-assembly**. *Genome Biol* 2008, **9**(3):R55.

Pride DT, Meinersmann RJ, Wassenaar TM, Blaser MJ: **Evolutionary implications of microbial genome tetranucleotide frequency biases**. *Genome Res* 2003, **13**(2):145-158.

Provost F, Fawcett T: **Robust classification for imprecise environments**. *Mach Learn* 2001, **42**(3):203-231.

Raes J, Foerstner K, Bork P: **Get the most out of your metagenome: computational analysis of environmental sequence data**. *Curr Opin Microbiol* 2007, **10**(5):490-498.

Rasko DA, Rosovitz MJ, Myers GSA *et al.*: **The pangenome structure of *Escherichia coli*: comparative genomic analysis of *E. coli* commensal and pathogenic isolates**. *J Bacteriol* 2008, **190**(20):6881-6893.

Read TD, Brunham RC, Shen C *et al.*: **Genome sequences of *Chlamydia trachomatis* MoPn and *Chlamydia pneumoniae* AR39**. *Nucleic Acids Res* 2000, **28**(6):1397-1406.

Read TD, Myers GSA, Brunham RC *et al.*: **Genome sequence of *Chlamydophila caviae* (*Chlamydia psittaci* GPIC): examining the role of niche-specific genes in the evolution of the Chlamydiaceae**. *Nucleic Acids Res* 2003a, **31**(8):2134-2147.

Read TD, Peterson SN, Tourasse N *et al.*: **The genome sequence of *Bacillus anthracis* Ames and comparison to closely related bacteria**. *Nature* 2003b, **423**(6935):81-86.

Reysenbach A-L, Hamamura N, Podar M *et al.*: **Complete and draft genome sequences of six members of the Aquificales**. *J Bacteriol* 2009, **191**(6):1992-1993.

Richter D, Ott F, Auch A, Schmid R, Huson D: **Metasim—a sequencing simulator for genomics and metagenomics**. *PLoS One* 2008, **3**(10):e3373.

Richter DC, Ott F, Auch AF, Schmid R, Huson DH: **User Manual for MetaSim V0.9.5**. 2009.

Richter M, Rosselló-Móra R: **Shifting the genomic gold standard for the prokaryotic species definition**. *Proceedings of the National Academy of Sciences* 2009, **106**(45):19126.

Rodríguez-Valera F: **Environmental genomics, the big picture?** *FEMS Microbiol Lett* 2004, **231**(2):153-158.

Schatz MC, Phillippy AM, Shneiderman B, Salzberg SL: **Hawkeye: an interactive visual analytics tool for genome assemblies**. *Genome Biol* 2007, **8**(3):R34.

Schleper C, DeLong EF, Preston CM, Feldman RA, Wu KY, Swanson RV: **Genomic analysis reveals chromosomal variation in natural populations of the uncultured psychrophilic archaeon *Cenarchaeum symbiosum***. *J Bacteriol* 1998, **180**(19):5003-5009.

Schloss PD, Westcott SL: **Assessing and Improving Methods Used in Operational Taxonomic Unit-Based Approaches for 16S rRNA Gene Sequence Analysis**. *Applied and environmental microbiology* 2011, **77**(10):3219-3226.

Seshadri R, Myers GSA, Tettelin H *et al.*: **Comparison of the genome of the oral pathogen *Treponema denticola* with other spirochete genomes**. *Proc Natl Acad Sci U S A* 2004, **101**(15):5646-5651.

Shen Y, Wan Z, Coarfa C *et al.*: **A SNP discovery method to assess variant allele**

probability from next-generation resequencing data. *Genome research* 2010, **20**(2):273-280.

Shendure J, Ji H: **Next-generation DNA sequencing**. *Nat Biotechnol* 2008, **26**(10):1135-1145.

Stein JL, Marsh TL, Wu KY, Shizuya H, DeLong EF: **Characterization of uncultivated prokaryotes: isolation and analysis of a 40-kilobase-pair genome fragment from a planktonic marine archaeon**. *J Bacteriol* 1996, **178**(3):591-599.

Teeling H, Meyerdierks A, Bauer M, Amann R, Glöckner FO: **Application of tetranucleotide frequencies for the assignment of genomic fragments**. *Environ Microbiol* 2004a, **6**(9):938-947.

Teeling H, Waldmann J, Lombardot T, Bauer M, Glöckner FO: **TETRA: a web-service and a stand-alone program for the analysis and comparison of tetranucleotide usage patterns in DNA sequences**. *BMC Bioinformatics* 2004b, **5**(1):163.

Tettelin H, Saunders NJ, Heidelberg J *et al.*: **Complete genome sequence of *Neisseria meningitidis* serogroup B strain MC58**. *Science* 2000, **287**(5459):1809-1815.

Touchon M, Hoede C, Tenaillon O *et al.*: **Organised genome dynamics in the *Escherichia coli* species results in highly diverse adaptive paths**. *PLoS Genet* 2009, **5**(1):e1000344.

Tringe S, Von Mering C, Kobayashi A *et al.*: **Comparative metagenomics of microbial communities**. *Science* 2005, **308**(5721):554-557.

Tyson G, Chapman J, Hugenholtz P *et al.*: **Community structure and metabolism through reconstruction of microbial genomes from the environment**. *Nature* 2004a, **428**(6978):37-43.

Tyson GW, Chapman J, Hugenholtz P *et al.*: **Community structure and metabolism through reconstruction of microbial genomes from the environment**. *Nature* 2004b, **428**(6978):37-43.

Venter J, Remington K, Heidelberg J *et al.*: **Environmental genome shotgun sequencing of the Sargasso Sea**. *Science* 2004, **304**(5667):66-74

Wagner D, Lipski A, Embacher A, Gattinger A: **Methane fluxes in permafrost habitats of the Lena Delta: effects of microbial community structure and organic matter quality**. *Environ Microbiol* 2005, **7**(10):1582-1592.

Ward D, Xu Q, Earl A *et al.*: **The genome sequence of *Lactobacillus crispatus* strain CTV-05**. Unpublished data; 2010a.

Ward D, Young SK, Kodira CD *et al.*: **The genome sequence of *Brucella abortus* NCTC 8038**. Unpublished data; 2009.

Ward D, Young SK, Zeng Q *et al.*: **The genome sequence of *Neisseria gonorrhoeae* strain F62**. Unpublished data; 2010b.

Wayne L, Brenner D, Colwell R *et al.*: **Report of the ad hoc committee on reconciliation of approaches to bacterial systematics**. *International Journal of Systematic Bacteriology* 1987, **37**(4):463-464.

Webster JD: **Analysing the genome of *Methanogenium frigidum***. *Honours.* Sydney: University of New South Wales; 2010.

Welch R, Burland V, Plunkett G *et al.*: **Extensive mosaic structure revealed by the complete genome sequence of uropathogenic *Escherichia coli***. *Proc Natl Acad Sci U S A* 2002, **99**(26):17020-17024.

Wheeler D, Barrett T, Benson D *et al.*: **Database resources of the national center for biotechnology information**. *Nucleic Acids Res* 2006, **35**(Suppl 1):D5-D12.

Whitaker RJ, Banfield JF: **Population genomics in natural microbial communities**.

Trends Ecol Evol 2006, **21**(9):508-516.

Zerbino DR, Birney E: **Velvet: algorithms for de novo short read assembly using de Bruijn graphs**. *Genome Res* 2008, **18**(5):821-829.

Zhaxybayeva O, Swithers KS, Lapierre P *et al.*: **On the chimeric nature, thermophilic origin, and phylogenetic placement of the Thermotogales**. *Proc Natl Acad Sci U S A* 2009, **106**(14):5865-5870.

Appendix A

Supplementary Results for Chapter 2

Table A.1: GC Content does not explain variation in error rates between species.

Species	GC Content	Discrepancies per kb
B. burgdorferi	0.2894	32.71
M. arthritidis	0.3148	186.71
Hydrogenobaculum sp. Y04AAS1	0.3513	28.5
D. turgidum	0.3541	95.71
P. marinus	0.3746	16.29
C. caviae	0.4552	41.76
N. meningitidis	0.5136	7.61
D. geothermalis	0.6630	10.41

Table A.2: Coverage only partially explains variation in error rates between species.

Species	Discrepancies per kb	Coverage	Normalised Discrepancies per kb
Hydrogenobaculum sp. Y04AAS1	28.497	5.709	4.991
B. burgdorferi	32.706	10.234	3.196
C. caviae	41.755	6.440	6.484
D. geothermalis	10.414	5.530	1.883
D. turgidum	95.711	8.950	10.694
M. arthritidis	186.706	16.662	11.206
N. meningitidis	7.612	3.768	2.020
P. marinus	16.293	6.894	2.363

Appendix B

Supplementary Results for Chapter 3

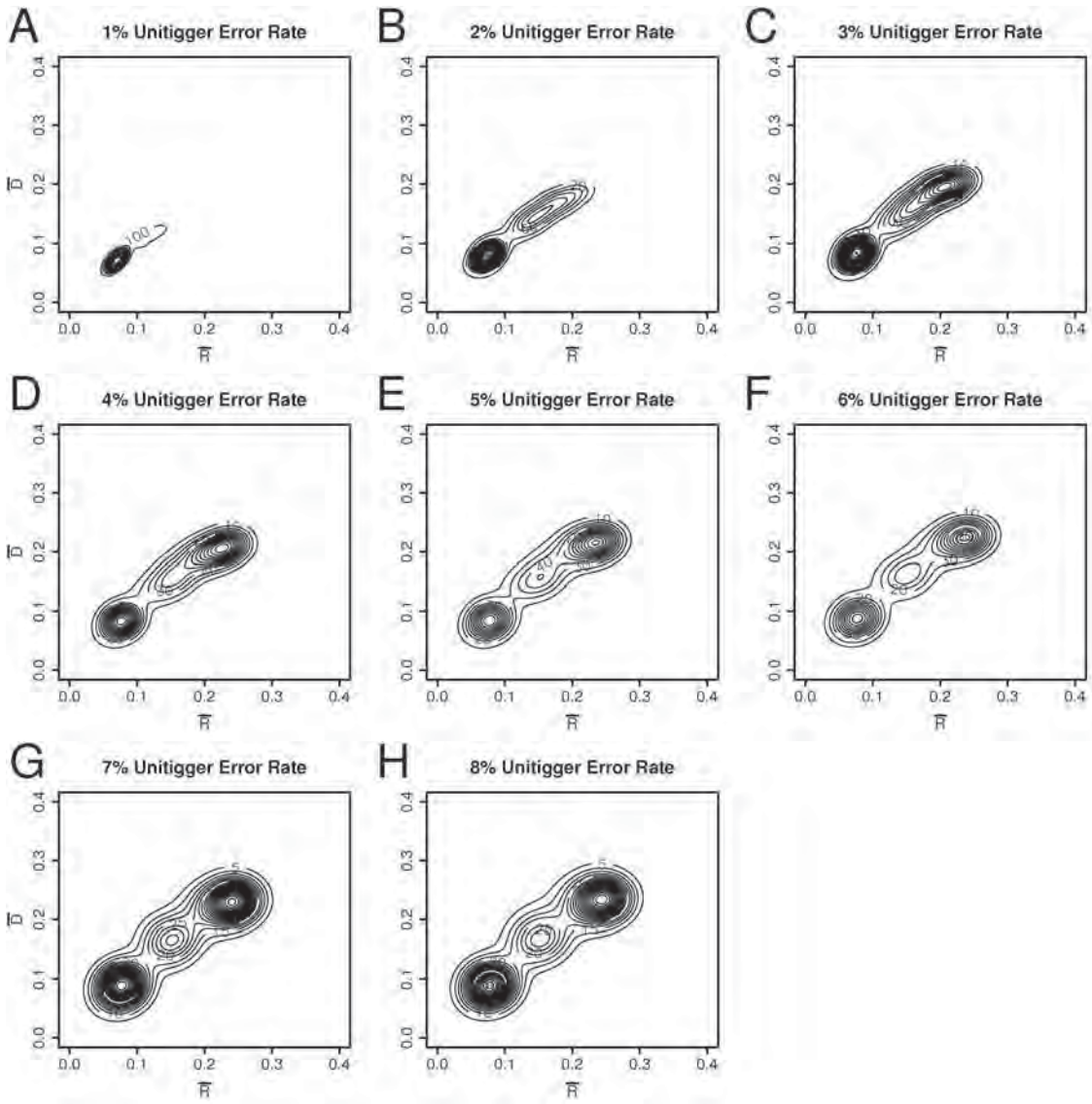

Figure B.1: Contour plots show a lower ideal unitigger error rate.
Three-strain *E. coli* assemblies with varying unitigger error rates.
$\bar{R}$: reads per unit of unitig length. $\bar{D}$: discrepancies per unit of unitig length.

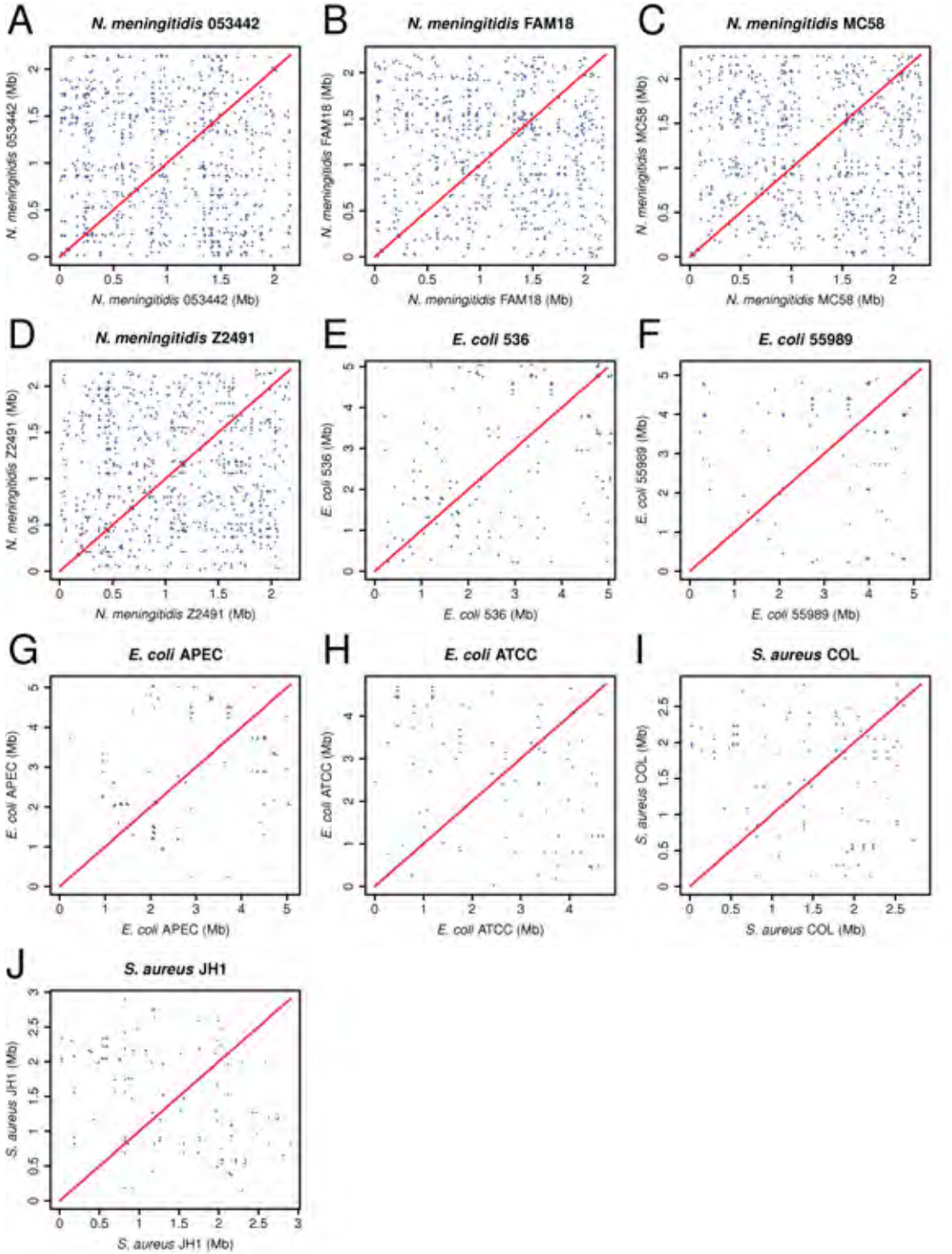

Figure B.2: *N. meningitidis* has more regions of self-similarity than *E. coli* or *S. aureus*.

Appendix C
Supplementary Results and Discussion for Chapter 4

C.1 Investigation, Filtering and Comparison of Outliers

For all three species used in simulated assemblies (*E. coli*, *S. aureus* and *N. meningitidis*), there were unitigs that had a much higher read depth than expected. Outlier rejection was used to improve the *MclustDA* training data.

To determine whether outliers in assemblies of different species and with different assembly settings had similar sequences, 4% and 8% two-strain assemblies of each of the three species were produced. In this comparison, the unitigs with an $\bar{R}'$ value of at least 0.2 were classed as outliers (Figure C.1). For four of these assemblies, the designated outliers were all at least 2.5 standard deviations from the mean $\bar{R}'$ value of the two-strain cluster (Figure C.1A to D). For the 8% *N. meningitidis* and *S. aureus* assemblies, except for one outlier, all outliers were at least two standard deviations from the mean (Figure C.1E and F).

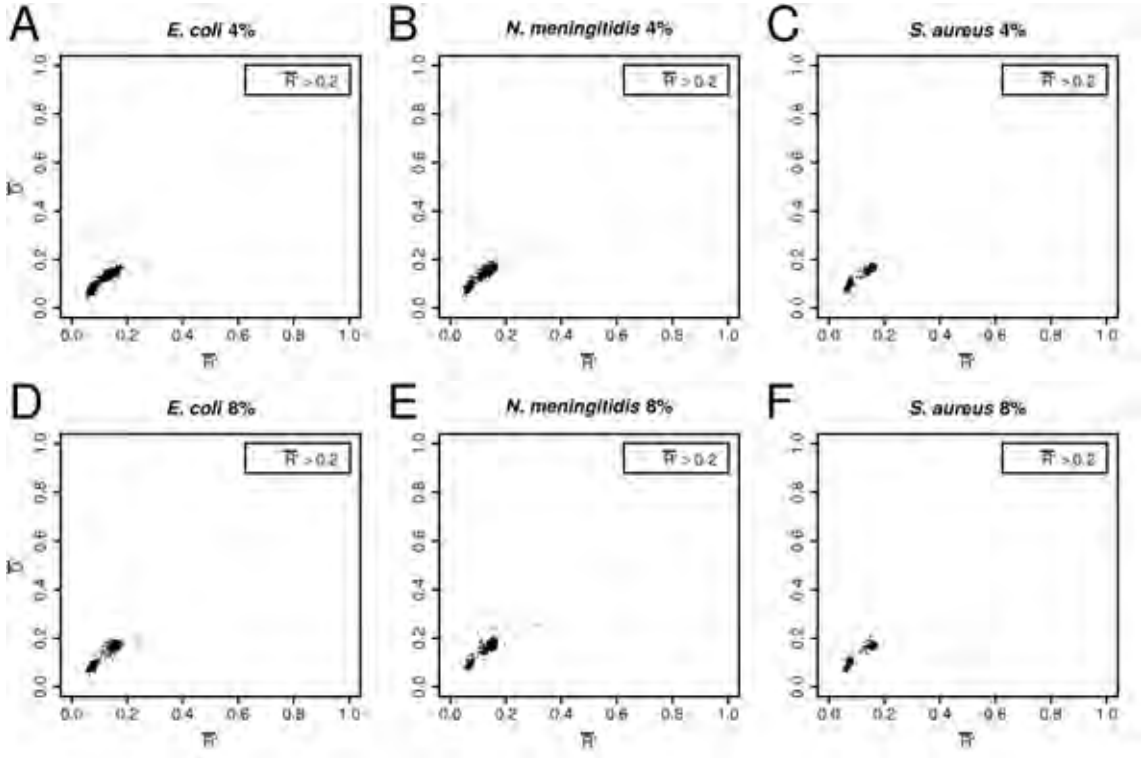

Figure C.1: Similar outliers were found in assemblies of different strains and different unitigger error rates.
Two-strain normalised assemblies.
$\bar{R}'$: reads per unit of unitig length. $\bar{D}'$: discrepancies per unit of unitig length.

Outliers in *Grinder* assemblies for each of the three species were compared with those in the *MetaSim* 4% and 8% assemblies. *Cross_match* was used to map outliers in each assembly for each of the three species to the outliers in the other assemblies of the same species. Outliers were mostly consistent between error rates and simulators. All outliers from the 4% assemblies mapped to 8% assembly outliers (Table C.1 and Table C.2). There were more outliers at 8% and nine out of 30 of these did not map to outliers in the 4% assemblies. All of the *MetaSim* 4% outliers mapped to at least one outlier in the equivalent *Grinder* assembly. For the *E. coli* assemblies, six out of ten *Grinder* outliers mapped to *MetaSim* ones. For *N. meningitidis*, 11 out of 13 did. For *S. aureus*, all four mapped back to a *MetaSim* outlier.

A high proportion of repetitive genes such as phage proteins and transposases were reported in these outliers by *BLASTX* using the NR database. Such genes may stack up to produce the unexpectedly high read depth as these genes are likely to be present more than once per strain. For the *E. coli* 8% assembly, seven of the outliers mapped back to the 4% ones (Table C.1). Three out of the 10 distinct outliers (i.e. the 4% outliers plus the unique 8% ones) received hits to a phage gene. Three received hits to transposases

and two to a repeat-containing protein. By expanding the number of hits reported for each unitig from five to 10, an additional two unitigs received hits to phage (6504 and 2155).

Table C.1 *E. coli* two-strain assembly outlier mapping.

Contig IID					
4% Outlier		**8% Outlier**		**4% Other**	***BLASTX* results**
6504	↔	2060	→	6502	Glutamate decarboxylase (beta subunit/isozyme)/ chain A, crystal structure of an N-terminal deletion mutant of *E. coli* Gadb in an autoinhibited state
6570	↔	2023	→	6571	Host specificity protein J/ putative tail component of prophage/ fibronectin type III domain protein
	→	6565			
6640	↔	2563			Translation elongation factor Tu/ chain A, structure of viral polymerase form I/Qb replicase
7455	↔	1203			YD repeat containing/ RHS repeat containing/ rhsA
7457	↔	3255			YD repeat containing/ Type I RHS/ core protein
	↔	2972			
9184	↔	3311			Integrase catalytic region/ transposase InsF for insertion sequence IS3/ SEc14 transposase B
		2020	→	7533	Tail fibre component K/ (phage) minor tail protein (L)
			→	6582	
			→	7535	
			→	6577	
			→	6576	
			→	6573	
		2155	→	6609	(Conserved) hypothetical
		2424	→	7858	Integrase (family protein)/ prophage P4 integrase
			→	2748	
			→	7861	
		2549	→	6966	Putative transposase insL for insertion sequence element IS186/ unnamed/ transposase DDE domain-containing/IS4 family
			→	7582	

Double-headed arrows denote a bidirectional mapping from outliers in the 4% assembly to outliers in the 8% assembly. Single arrows denote a unidirectional mapping from outliers in one of these assemblies. The IID (Internal IDentifier) numbers of the contigs are shown. Text in brackets shows optional text. Contig identifiers have been given to these unitigs to allow their inclusion in the conversion of the assembly to *AMOS* bank format.

The *N. meningitidis* two-strain assembly also contained phage and transposon related outliers (Table C.2). In this assembly, one out of 13 distinct outliers mapped to phage genes and two to transposons. Both phage genes and transposable elements have been found in high copy number within bacterial genomes (Blattner *et al.* 1997, Parkhill *et al.* 2000). Repetitive regions of the genome may not always contain genes. However, all outliers examined received hits to genes with an E-value of at most 6×10^{-49}.

Table C.2 *N. meningitidis* two-strain assembly outlier mapping.

Contig IID						***BLASTX* results**
4% Outlier		**8% Outlier**		**4% Other**		
2515	↔	2519	→	2517		(Adhesion) MafA2/3/ putative lipoprotein/ adhesin, MafA (family)
	→	2518				
2876	↔	1533	→	4445		Replication initiation factor/ putative phage protein
			→	2875		
	↔	1534	→	2877		
3051	↔	586	→	3050		FrpC operon protein
3062	↔	1768	→	3063		Iron-regulated protein frpA/C
	→	1771				
3146	↔	1485	→	3147		(Conserved) hypothetical protein/ pG1 protein
3149	↔	1485				Cell wall associated hydrolase/ conserved hypothetical/ CrcB protein domain protein
3157	↔	1824	→	3158		Glucose-1-phosphate thymidylyltransferase
3159	↔	1824				dTDP(-D)-glucose 4,6-dehydratase
4163	↔	1490	→	4164		(Putative) invertase/transposase/ transposase, IS116/IS110/IS902 family/ pilin gene inverting protein PivNM-1A
			→	4166		
			→	4167		
4180	↔	1502				Hypothetical/ replication initiation factor/ unnamed
	→	1501				
		1493	→	4168		TspB (family) protein
			→	4169		
			→	4171		
		2482	→	4016		MafB(-related) protein/ adhesion/ (conserved) hypothetical protein
			→	4015		
			→	4013		
		2677	→	4556		(Translation) elongation factor Tu
			→	4558		
			→	4559		

Double-headed arrows denote a bidirectional mapping from outliers in the 4% assembly to outliers in the 8% assembly. Single arrows denote a unidirectional mapping from outliers in one of these assemblies. The IIDs of the contigs are shown. Text in brackets shows optional text. Contig identifiers have been given to these unitigs to allow their inclusion in the conversion of the assembly to *AMOS* bank format.

The majority of outliers mapped to outliers in assemblies with different unitigger error rates and to those in assemblies using a different read simulator. The genes detected in these outliers are similar across species. However, the set of genes detected was not sufficiently predictable to allow sequence based filtering of the outliers.

Filtering sequences by gene type would require a closely related species with an annotated genome. This restriction would be excessive for metagenomic studies because the species of interest and the close relatives of that species are often poorly studied.

Analogous outliers were found for clusters with less than the maximum quantity of strains. These often overlap with the adjacent cluster furthest from the origin. Filtering of these kinds of unitigs could also improve the analysis.

C.2 Training Data

In Chapter 3, a binning method was described that assigned a real number, the strain number estimator S, to each unitig (Equation 3.1). This continuum was then transformed back into discrete bins by specifying a width range of this number.

When the discrete S bins used a width of zero, there were no unitigs classified to the three- and four-strain bins (Figure C.2A). Moreover, the one-strain cluster had points spread across the two-strain cluster and beyond. With a width of ± 0.05 there were unitigs in every bin but none of the bins were well separated (Figure C.2B).

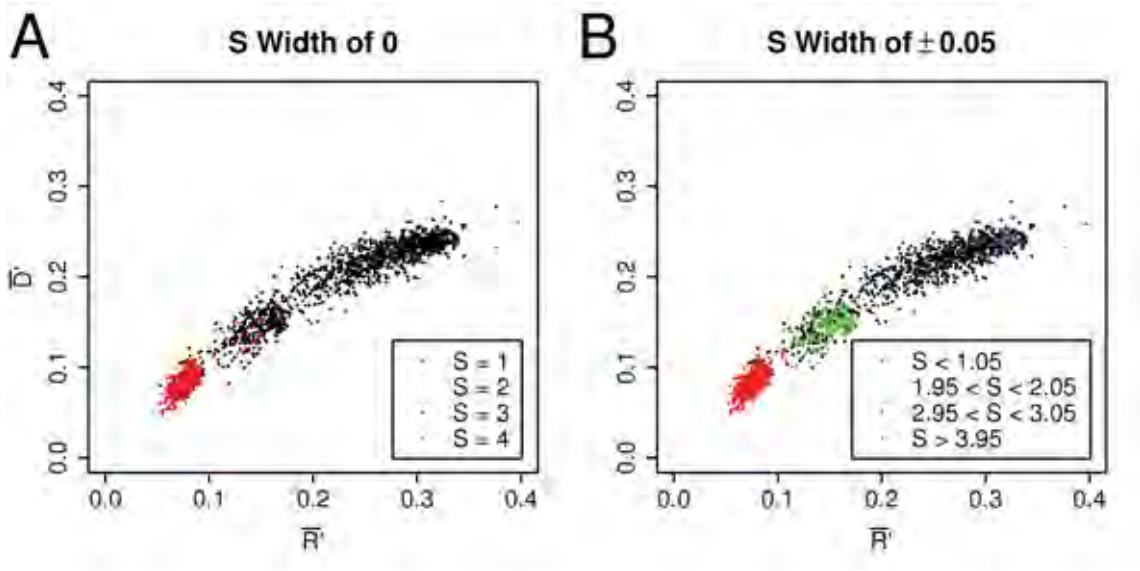

Figure C.2: Neither a width of zero or ± 0.05 creates strong well-separated clusters for all bins.
E. coli four-strain assembly binning using S:
A) Bins with an S width of zero.
B) Bins with an S width of ± 0.05.
$\bar{R}'$: reads per unit of unitig length. $\bar{D}'$: discrepancies per unit of unitig length.

In order to cluster the bins more tightly, the bins were further restricted by removing those unitigs in each bin that were not within a given number of standard deviations from the mean (μ) $\bar{R}'$ and $\bar{D}'$ values. Width restrictions of ± 0.05 for S and ± 1 σ gave well separated clusters but with only a few observations especially for the third cluster (Figure C.3A). Width restrictions of ± 0.5 for S and ± 1 σ increased the minimum observations per cluster from 6 to 69, though the resultant clusters were more rectangular with means shifted to lower $\bar{R}'$ values (Figure C.3B). Thus, a compromise must be made between signal strength and quality.

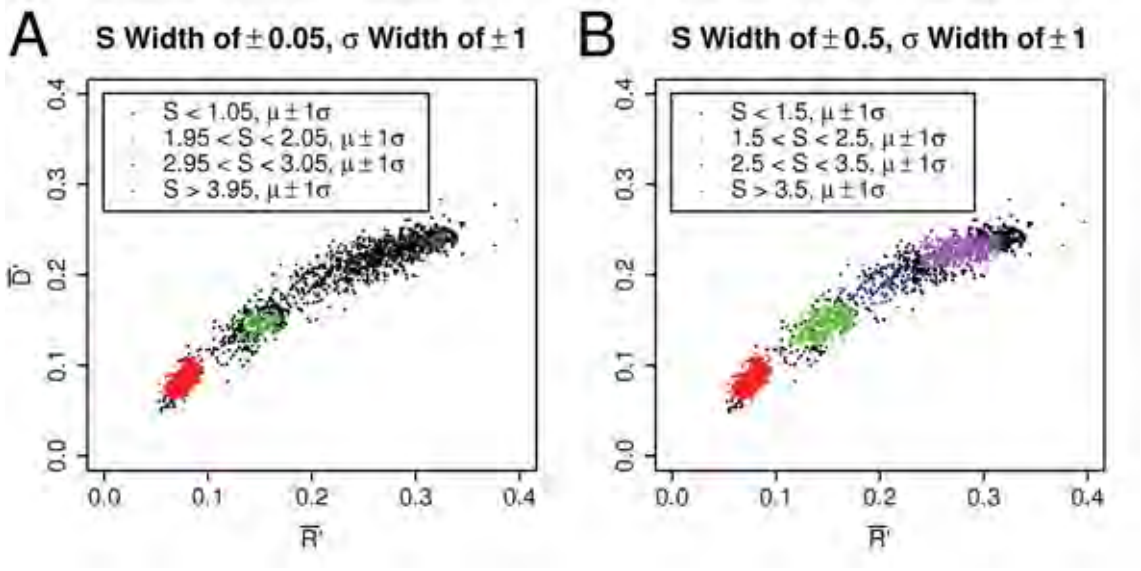

Figure C.3: Width selection involved a trade off between number of unitigs and positioning of cluster means.
E. coli four-strain assembly binning using ***S*** and σ:
A) Bins with width restrictions of ± 0.05 for S and ± 1 σ.
B) Bins with width restrictions of ± 0.5 for S and ± 1 σ.
$\bar{R}'$: reads per unit of unitig length. $\bar{D}'$: discrepancies per unit of unitig length.

To test this parameter space, an *N. meningitidis* four-strain assembly was used to make predictions on an equivalent three-strain assembly. In real microbial communities, the number of strains present is normally not known. The ability to make predictions using training data that does not exactly match the sample to be studied would be advantageous. S bin widths of ± 0.05, 0.1, 0.2, 0.3, 0.4 and 0.5 were used. Each of these widths was used twice – once with σ widths of one (Table C.3) and once with 1.5 (Table C.4). The test error reported by *MclustDA* was used to compare the different training data. AUCs are provided in Table C.3 as a comparison. Since test errors had a strong inverse correlation with AUCs for that training data, the AUCs for Table C.4 were not calculated. Table 4.2 and Table C.5 describe the models used to cluster each

bin. Each of the bin widths within the 1.5 σ set gave higher test errors than the corresponding bin width within the 1 σ set. *MclustDA* classified 26 out of 40 of the training clusters as multiple (up to nine) component clusters.

Table C.3 *MclustDA* training and test results with 1 σ widths.

Widths	± 0.05	± 0.1	± 0.2	± 0.3	± 0.4	± 0.5
Training Error	0	0.02583	0.04675	0.05682	0.06782	0.07360
Test Error	0.3555	0.09142	0.07901	0.1050	0.09932	0.1027
Cluster Models	EEI 2	VEI 3	XXX 1	VII 2	XXX 1	XXX 1
	EEV 6	EII 2	XXX 1	EII 2	EII 2	EII 2
	EEI 9	XXX 1	XXX 1	EEV 2	XXX 1	XXX 1
	EEV 6	XXX 1	XXX 1	EEE 3	EEE 3	EEE 3
AUCs	0.9741	0.9690	0.9730	0.9659	0.9818	0.9792
	0.7931	0.8764	0.8884	0.8796	0.8472	0.8564
	0.6880	0.9718	0.9778	0.9617	0.9769	0.9807
Mean AUC	0.8184	0.9391	0.9464	0.9357	0.9353	0.9388

Predictions were made on an *N. meningitidis* three-strain assembly using a four-strain assembly as test data. The AUCs are shown for each cluster separately.

Table C.4 *MclustDA* training and test results with 1.5 σ widths.

Widths	± 0.05	± 0.1	± 0.2	± 0.3	± 0.4	± 0.5
Training Error	0	0.01765	0.06024	0.07324	0.09741	0.1020
Test Error	0.2810	0.09594	0.09255	0.1095	0.1084	0.1230
Cluster Models	XXX 1	XXX 1	XXX 1	XXX 1	XXX 1	VVV 2
	VII 5	EII 2	XXX 1	XXX 1	XXX 1	EII 3
	EII 5	XXX 1	XXX 1	EEV 3	VEV 2	VEV 2
	EII 2	XXX 1	XXX 1	EEE 2	EEE 2	EEE 2

Predictions were made on an *N. meningitidis* three-strain assembly using a four-strain assembly as test data.

Table C.5 *MclustDA* single component models

Single Component Models	**Explanation**
X	One-dimensional
XII	Spherical
XXI	Diagonal
XXX	Ellipsoidal

Since the clustering of the bins was often over fitted, this experiment was repeated with all training classification limited to one ellipsoidal cluster per training cluster (setting XXX). This restriction led to improvements for seven out of 10 of the parameter combinations but did not improve upon the best result of 7.9% test error for the combined widths of 1 σ and S ± 0.2. This width combination had automatically been

assigned a single ellipse per cluster.

C.3 Location of Clusters

Three-strain *E. coli Grinder* assemblies were produced with their strains in varying proportions and with substitution errors. These assemblies had strains with per-strain read depths of: 6, 6 and 48×; 6, 18 and 36×; 12, 12 and 24×; 6, 24 and 30×; 6, 12 and 42× and 20, 20 and 20× (Figure C.4). The clusters in these plots are stretched due to the higher, more realistic substitution error rates in these assemblies. This increases the variability of discrepancy rates for the unitigs in each cluster.

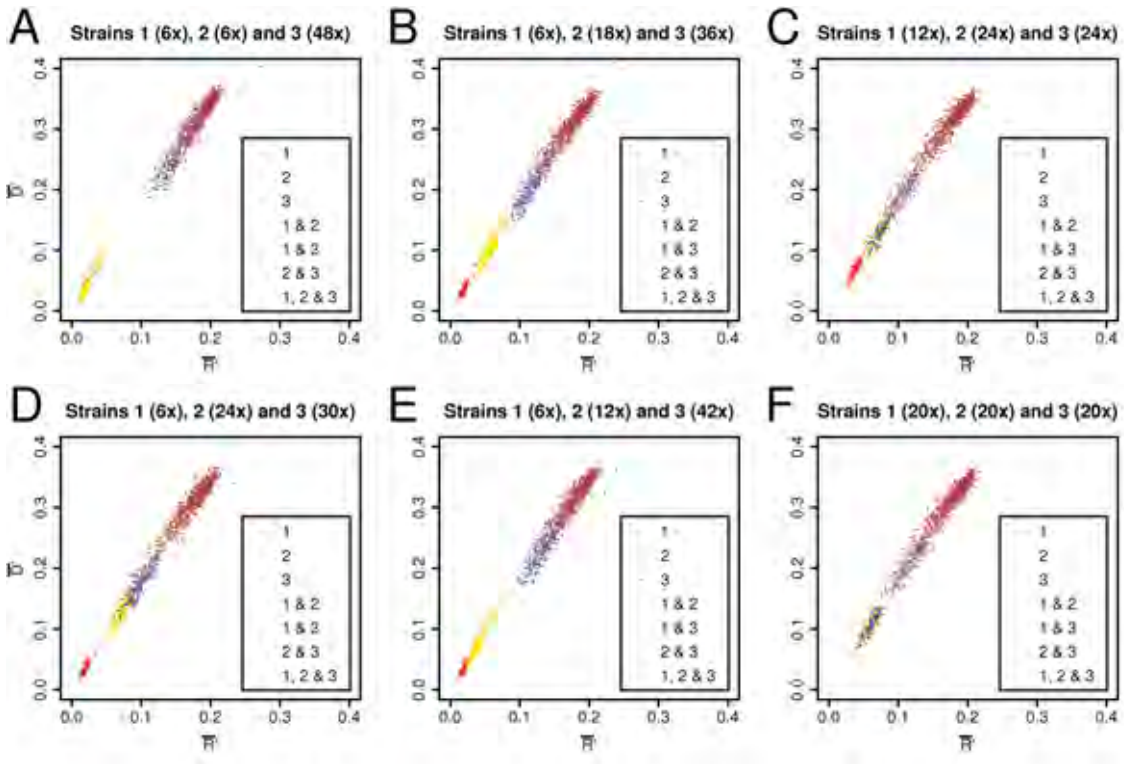

Figure C.4: Cluster location is predictable for three-strain assemblies with varying strain proportions.
Three-strain *E. coli Grinder* varying proportions assemblies with substitution errors.
Red contigs: $\boldsymbol{S} < 1.5$ and *E. coli* 55989 > 50%.
Yellow contigs: $\boldsymbol{S} < 1.5$ and *E. coli* APEC O1 > 50%.
Blue contigs: $\boldsymbol{S} < 1.5$ and *E. coli* ATCC 8739 > 50%.
Orange contigs: $1.5 \leq \boldsymbol{S} \leq 2.5$ and *E. coli* 55989 > *E. coli* ATCC 8739 and *E. coli* APEC O1 > *E. coli* ATCC 8739.
Purple contigs: $1.5 \leq \boldsymbol{S} \leq 2.5$ and 55989 > *E. coli* APEC O1 and *E. coli* ATCC 8739 > *E. coli* APEC O1.
Green contigs: $1.5 \leq \boldsymbol{S} \leq 2.5$ and *E. coli* APEC O1 > *E. coli* 55989 and *E. coli* ATCC 8739 > *E. coli* 55989.
Brown contigs: $\boldsymbol{S} > 2.5$.
Strain 1: *E. coli* 55989. Strain 2: *E. coli* APEC O1. Strain 3: *E. coli* ATCC 8739.
$\bar{R}'$: reads per unit of unitig length. $\bar{D}'$: discrepancies per unit of unitig length.

The locations of the clusters in these assemblies follow the same pattern as *S. aureus*. There is a strong linear relationship between the clonal clusters from each strain ($R^2 \geq 0.9999$) (Figure C.5A). This relationship also applies to the set of all clonal clusters ($R^2 = 0.9991$). The *E. coli* APEC O1 and *E. coli* ATCC 8739 clusters (green) contain a low number of unitigs which decreases the accuracy of their cluster centres and the linearity of their relationship ($R^2 = 0.7042$) (Figure C.5B). The low number of unitigs is due to these strains having a lower similarity than the other combinations of strains (Table 3.1). The clusters for the other two pairs of strains have a linear

relationship ($R^2 \geq 0.9962$), as does the set of all two strain clusters ($R^2 = 0.9604$). The pattern is also applicable across the combined dataset of clonal, two-strain and three-strain clusters ($R^2 = 0.9863$). The clusters in these figures have a steeper gradient due to the increased substitutions in these *Grinder* reads. This is because more substitutions leads to a large increase in the number of discrepancies detected in each unitig. Increased substitutions had a small negative effect on read depth, as some reads were no longer similar enough to be assembled together.

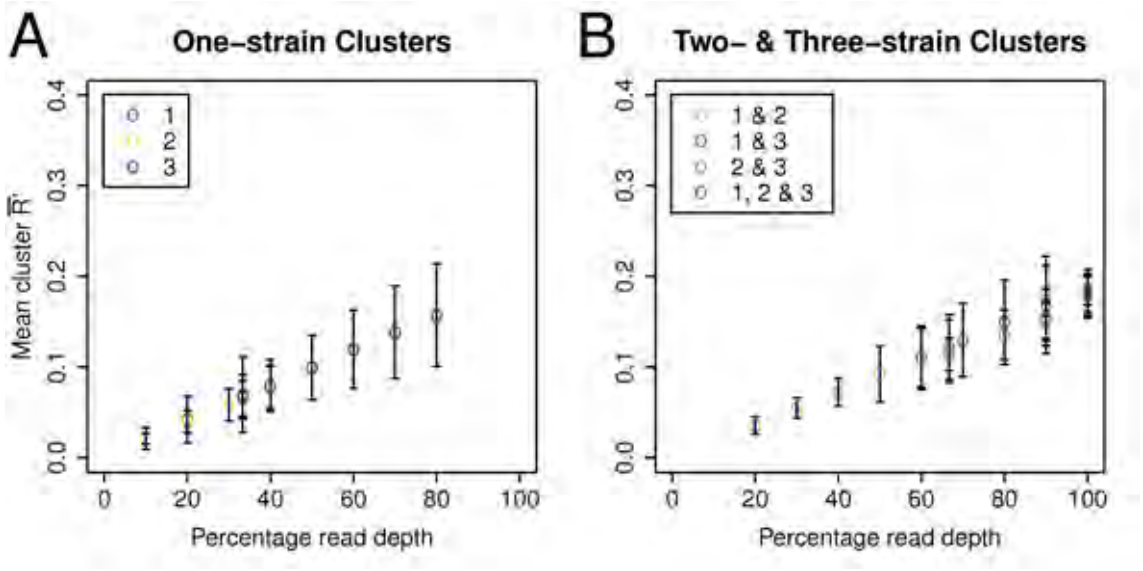

Figure C.5: The linear relationship between cluster location and strain proportions applies to three strain assemblies.
Three-strain *E. coli Grinder* varying-proportions assemblies with substitution errors. Mean cluster $\overline{R'}$ values versus cluster read depths.
$\bar{R}'$: reads per unit of unitig length.

C.4 Idealised Assemblies with Zero Sequencing Errors

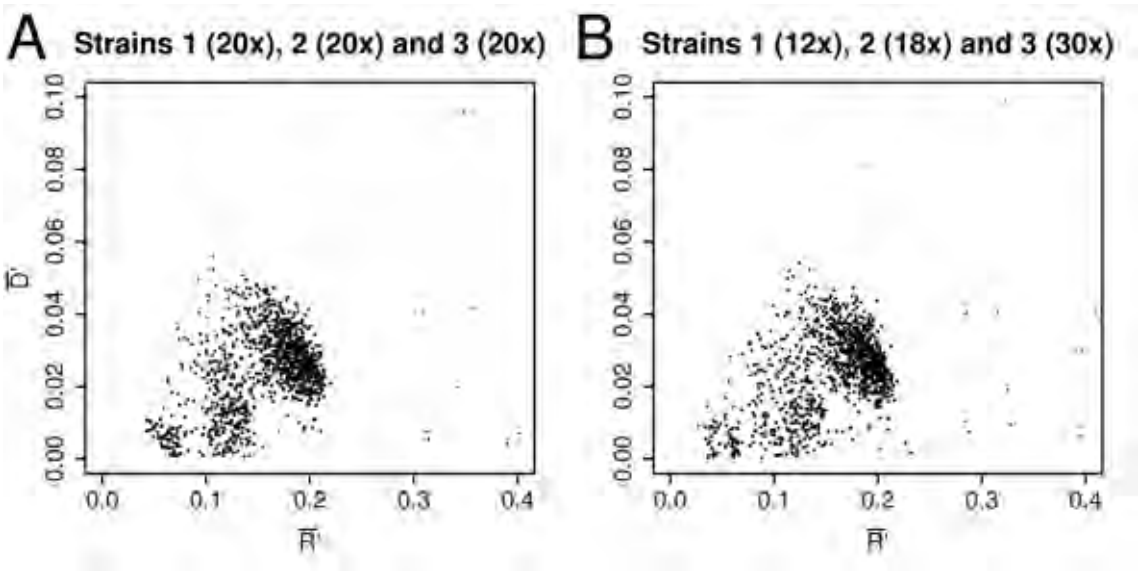

Figure C.6: The boundaries between clusters are not clear in idealised assemblies.
Three-strain *E. coli* assemblies with zero sequencing errors.
Strain 1: *E. coli* 55989. Strain 2: *E. coli* APEC O1. Strain 3: *E. coli* ATCC 8739.
$\overline{R}'$: reads per unit of unitig length. $\overline{D}'$: discrepancies per unit of unitig length.

C.4 Oligonucleotide Frequency Filtering

The program *Tetra* (www.megx.net/tetra; Teeling *et al.* 2004a) was investigated in conjunction with GC content and *BLAST* to filter the ANTRC230_0.1 data. *Tetra* calculates di-, tri-, tetra- and penta-nucleotide frequencies in order to compare DNA sequences. The use of longer oligonucleotides should better discriminate between species (Pride *et al.* 2003), but these require longer input sequence lengths to work effectively. This is because longer oligomers have exponentially more sequence combinations, which means that more samples are required to establish an estimate for the frequency of any given oligomer. The *Windows* (Microsoft Corporation, Redmond, WA, USA) version of this software contains fewer features than, and gave slightly different results in comparison to, the *Mac OS* (Apple, Cupertino, CA, USA) version.

To bin the hybrid assembly of 454 and Sanger reads, ROC plots of the di-, tri- and tetra-nucleotide frequencies; GC content and *BLAST* E-values against *P. vibrioformis* (Figure C.7) were produced. These were used to determine which measurement correlated most effectively with the previous binning of Sanger reads. The scaffolds in

the filtered Sanger assembly were converted to a list of unitigs in the unfiltered hybrid assembly. This was done by listing all the hybrid unitigs that contain at least one read from the Sanger scaffolds. This list of unitigs was used as the test data for the ROC plots.

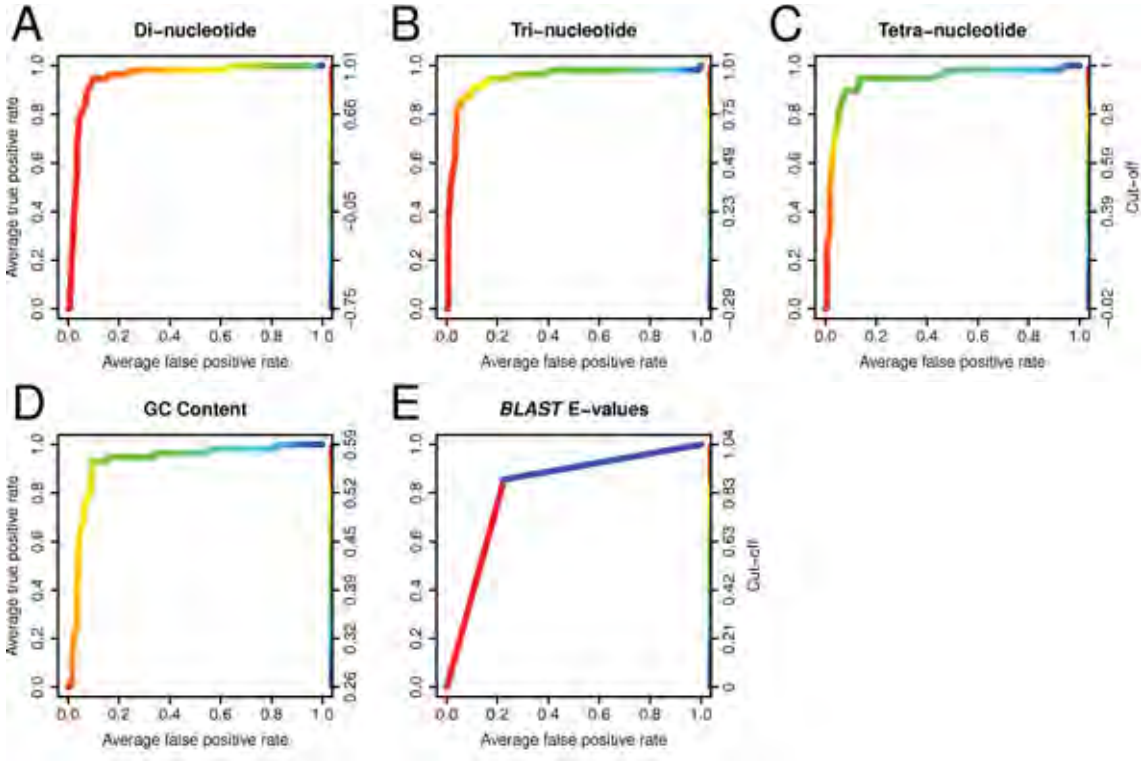

Figure C.7: Di-nucleotide frequencies make the most effective classifier.
ROC plots using unitigs with reads from the Sanger assembly as test data. Comparison of di-, tri- and tetra-nucleotide frequencies, GC content and *BLAST* E values.

Other ROC plots were made of the di-, tri- and tetra-nucleotide frequencies and GC content with *BLAST* hits used as the test data (Figure C.8).

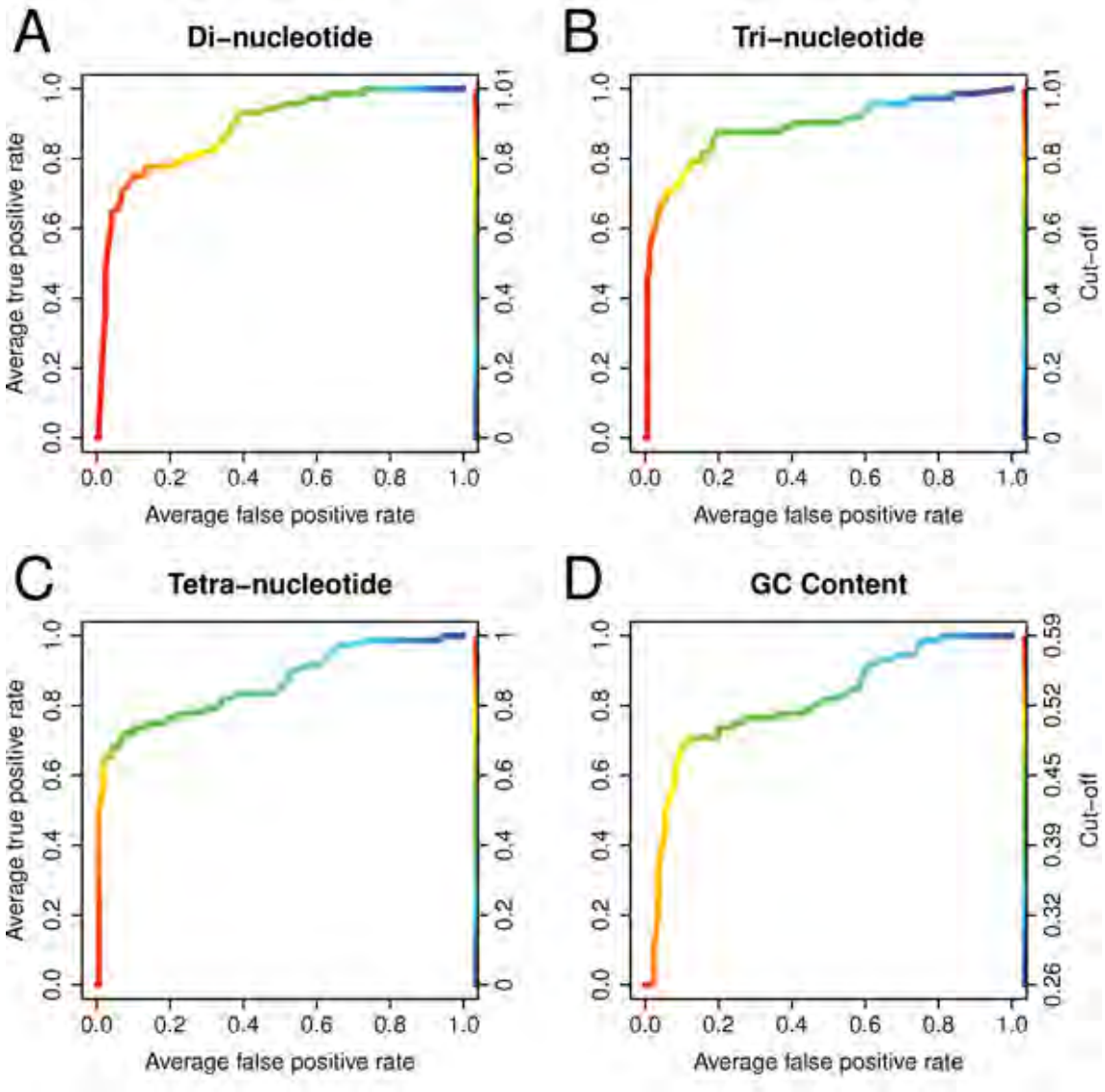

Figure C.8: Di- or tri-nucleotide frequencies make the most effective classifiers.
ROC plots using *BLAST E-value*s of the unitigs against *P. vibrioformis*. Comparison of di-, tri- and tetra-nucleotide frequencies and GC content.

From these plots, it was decided that di-nucleotide frequency (dimers) was the most effective method for filtering the ANTRC230_0.1 dataset. This is because dimers produced the classifier with the highest AUC value against either test data set (Table C.6).

Table C.6: Di-nucleotide frequencies produced the best predictions.

From	To	AUC
GC	Sanger	0.9267
	BLAST	0.8107
Di-	Sanger	0.9524
	BLAST	0.8863
Tri-	Sanger	0.9454
	BLAST	0.8892
Tetra-	Sanger	0.9406
	BLAST	0.8607
BLAST	Sanger	0.8602

Dimers also have the advantage of working well with shorter fragments. *Tetra* recommends 20 kb lengths for tetra-nucleotides. Since this work involves unitigs and maximizing the number of observations is preferred for cluster analysis, this would have been severely limiting. In a typical assembly (Figure 3.7C) a cut-off of 20 kb would remove 98.5% of the unitigs over 1 kb. For a 20 kb fragment tetra-nucleotide frequencies have an uncertainty of:

$$\frac{\sqrt{\frac{20000}{4^4}}}{\frac{20000}{4^4}} = 11.31\%.$$

The equivalent lengths for trimers and dimers are $\frac{20000}{4^4} \times 4^3 = 5$ kb and $\frac{20000}{4^4} \times 4^2 =$ 1.25 kb. The time taken to run *Tetra* on large datasets and limited access to Macintosh computers meant that only unitigs greater than 2.5 kb were used.

Tetra is available on *Linux*, *Mac OS* and *Windows* operating systems but does not provide the same features on all. On *Windows*, it can only generate tetra-nucleotide frequencies whilst on *Mac OS* it can generate di-, tri- and penta-nucleotide frequencies as well. It was not possible to install *Tetra* on the available *Linux* system. Additionally, it was assumed that *Tetra* would have the same limitations of features on *Linux* as it has on *Windows*. Due to the limited availability of Macintosh computers to run this software, a custom script *tetra.py* was written to implement *Tetra*'s core routine. The methods followed those of Teeling *et al.* (2004a). The frequency counts were checked against *Tetra* and the Pearson correlations using R. However, it was found that *Tetra* does not use z-scores as described in their papers for di-nucleotide frequency calculations. Instead, it uses an undescribed method termed "DRAB". In one *Tetra* paper, Teeling *et al.* (2004a) describe the z-score method for tetra-nucleotide frequencies but neither this paper nor the other *Tetra* paper (Teeling *et al.* 2004b) mention what is used for other frequencies.

Since the implementation of the “DRAB” method was not known, the di-nucleotide scores for *tetra.py* used normalisation. The classifiers produced with these dimers achieved even higher AUCs of 0.9807214 against *P. vibrioformis* and 0.9457008 against *BLAST*. This gives extra support for the choice of dimers.