University of Glasgow

Algorithms for Viral Haplotype Reconstruction and Bacterial Metagenomics

Resolving Fine-Scale Variation in Next Generation Sequencing Data

by

Melanie Schirmer

Submitted in fulfilment of the requirements for the Degree of
Doctor of Philosophy

School of Engineering
College of Science and Engineering
University of Glasgow

August 2014

Abstract

The discovery of DNA has been one of the biggest catalysts in genomic research. Sequencing has enabled us to access the wealth of information encoded in DNA and has provided the basis for ground-breaking achievements such as the first complete human genome sequence. Furthermore, it has tremendously advanced our understanding of life-threatening genetic disorders and bacterial and viral infections. With the recent advent of next generation sequencing (NGS) technologies, sequencing became accessible to the majority of researchers and made metagenomic sequencing widely available. However, to realise its true potential, sophisticated and tailor-made bioinformatic programs are essential to translate the collected data into meaningful information.

My thesis explored the potential of resolving fine-scale variation in NGS data. The identification and correction of artificial fine-scale variation in the form of biases and errors is imperative in order to draw valid conclusions. Furthermore, resolving natural fine-scale variation in the form of single nucleotide polymorphisms (SNPs) and closely related species or strains is critical for the development of effective treatments and the characterisation of diseases. In recent years, Illumina has emerged as the global market leader in DNA sequencing. However, biases and errors associated with this high-throughput sequencing technology are still poorly understood which has precluded the development of effective noise removal algorithms. In addition, many programs were not designed for Illumina data or metagenomic sequencing. Therefore, a better understanding of the idiosyncrasies encountered in Illumina data is essential and programs must be tested and benchmarked on realistic and reliable *in silico* data sets to reveal not only their true capacities but also their limitations.

I conducted the largest *in vivo* study of Illumina error profiles in combination with state-of-the-art library preparation methods to date. For the first time, a direct connection between experimental design factors and systematic errors was established, providing detailed insight into the nature of Illumina errors. Further, I tested various error removal techniques and developed a sophisticated Illumina amplicon noise removal algorithm, enabling researchers to choose optimal processing strategies for their particular data sets. In addition, I devised several simulation tools that accurately reflect artificial and natural fine-scale variation. This includes a flexible and efficient read simulation program which is the only program that can directly reflect the impact of experimental design factors. Furthermore, I developed a program simulating the evolution of a virus into a quasi-species. These programs formed the basis for two comprehensive benchmarking studies that revealed the capacities and limitations of viral haplotype reconstruction programs and taxonomic classification programs, respectively. My work furthers our knowledge of Illumina sequencing errors and will facilitate more accurate and effective analyses of sequencing data sets.

Contents

List of Tables

List of Figures

Acknowledgments

I would like to start by thanking my supervisors, Chris Quince and Bill Sloan, for giving me the opportunity to undertake this PhD, who encouraged me to explore different areas of research and to become an independent researcher. In addition, I would like to thank Unilever R&D for supporting my PhD project and giving me the opportunity to present my research at international conferences and workshops.

For useful advice and discussions on the DPMMs I would like to thank Keith Harris. Furthermore, for patiently answering my questions regarding the experimental work I would like to thank Linda D'Amore. Thanks also go to Sarah, Julie and Anne for their support during my short excursion into the wet lab.

I would also like to thank my colleagues at the University of Glasgow. Many of them have become good friends over the years and made this journey a great experience: Steph, Elisa, Sarah, Melina, Maria, Jill, Asha, Siding, Gu, Marnie, Ross, Graeme, James and Mathieu. Thank you for all the times you lent me an ear, offered encouragement or a welcome distraction from work.

Besonderer Dank geht an meine Familie. Insbesondere an meine Mutter, die immer an mich glaubt, nie an meinem Erfolg zweifelt und mich stets ermutigt nach den Sternen zu greifen. An meinen Bruder, der immer für mich da ist, jederzeit mit Ratschlägen zur Seite steht und bereit ist seine Begeisterung zu teilen. An Matthias, für seine Abenteuerlust in Schottland und Irland. Und an meine beste Freundin Daniela, auf die ich immer zählen kann.

Declaration

I declare that no portion of the work in this thesis has been submitted in support of any application for any other degree or qualification from this or any other university or institute of learning. I also declare that the work presented in this thesis is entirely my own contribution unless otherwise stated.

Melanie Schirmer

Glasgow, August 2014.

Nomenclature

16S rRNA	16S ribosomal RNA
5NDI	nested dual index with five random nucleotides
A	adenine
AGBT	Advances in Genome Biology and Technology
AT	*Anaerocellum thermophilum Z-1320 DSM 6725*
BT	*Bacteroides thetaiotaomicron VPI-5482*
BV	*Bacteroides vulgatus ATCC 8482*
BWA	Burrows-Wheeler Aligner
BX	*Burkholderia xenovorans (LB400)*
C	cytosine
CCS	circular consensus sequencing
CIGAR	Compact Idiosyncratic Gapped Alignment Report
CoM	Completeness Measure
CS	*Caldicellulosiruptor saccharolyticus DSM 8903*
ddNTP	dideoxynucleotides
DI	nested dual index
DNA	deoxyribonucleic acid
dNTP	deoxyribonucleotide
DP	Dirichlet Process
DPMM	Dirichlet Process Mixture Model
DSV	*Desulfovibrio desulfuricans subsp. desulfuricans str. ATCC 27774*
EBOV	*Zaire ebolavirus*
EF	*Enterococcus faecalis V583*
emPCR	emulsion PCR
env	envelope glycoprotein gene of HIV-1

FG	Fusion Golay
FMV	foot-and-mouth virus
FMVD	foot-and-mouth virus data set
G	guanine
GA	Genome Analyzer
H	Hellinger distance
HA	*Herpetosiphon aurantiacus ATCC 23779*
HCV	hepatitis C virus
HF	HiFi Kapa Taq
HIV	human immunodeficiency virus
HMP	Human Microbiome Project
indels	insertions and deletions
KL	Kullbeck-Leibler
LC	*Leptothrix cholodnii SP-6*
MB	balanced mock community
MCMC	Markov Chain Monte Carlo
MDS	multidimensional scaling
MUB	unbalanced mock community
N	Nextera
NE	*Nanoarchaeum equitans Kin4-M*
NGS	next generation sequencing
P	Parkinson Low Input
PCR	polymerase chain reaction
pdf	probability density function
Polony	polymerase-colony
PPS	PhyloPythiaS

Q5	Q5 neb Taq
RBS	*Rhodopirellula baltica SH 1*
RHO	*Rhodospirillum rubrum ATCC 11170*
RNA	ribonucleic acid
rRNA	ribosomal RNA
SAM	Sequence Alignment/Map
SBS	sequencing-by-synthesis
SI	nested single index
SiM	Similarity Measure
SMRT	Single Molecule Real Time
SMS	Single Molecule Sequencing
SNP	single nucleotide polymorphism
SOLiD	Sequencing by Oligonucleotide Ligation and Detection
SSE	sequence specific errors
SVM	support vector machine
T	thymine
TT	*Thermus thermophilus HB8*
TV	*Treponema vincentii I*
VI	variational inference
WGS	whole genome shotgun sequencing
WHO	World Health Organisation
XT	NexteraXT

Glossary

in silico

The term *in silico* refers to an experiment conducted or data set generated on a computer.

in vitro

The term *in vitro* refers to a biological process produced in a controlled experimental environment rather than in a natural setting or within a living organism.

in vivo

The term *in vivo* refers to a biological process occurring in a natural setting or within a living organism.

16S rRNA

The 16S ribosomal RNA genes (16S rRNA) encode the small subunit ribosomal RNAs in prokaryotes. Encoding such a fundamental process as translation, these genes are present in all prokaryotic organisms and are highly conserved. Among all rRNA genes the 16S rRNA gene has proven to be most informative. It has been widely used as a phylogenetic marker to study the evolutionary relationship between prokaryotes and is also used for the identification of organisms.

base pair (bp)

Adenine (A) and Thymine (T) are complementary nucleotide bases bond by two hydrogen bonds whereas Guanine (G) and Cytosine (C) are connected by three hydrogen bonds. A base pair refers to one of the pairs A-T or C-G. Also, the length of a DNA sequence is generally measured in bp which corresponds to the number of nucleotide bases in the sequence.

coverage

Sequencing coverage refers to the number of times a specific position on a genome or DNA fragment is sequenced, i.e. the number of reads that cover a certain position.

deletion

A deletion error during sequencing describes the event where a base on the sequenced fragment was skipped and is not included on the read.

DNA/RNA polymerase

DNA and RNA polymerases are enzymes that can assemble DNA or RNA, respectively. DNA polymerases are responsible for DNA replication whereas RNA polymerases carry out transcription.

error rate

The error rate refers to the frequency of insertions, deletions and/or substitutions that occur on the reads during the sequencing process. These rates can be inferred by comparing the reads to their respective reference/true sequence from which they originate.

fragmentation

During fragmentation DNA strands are broken up into smaller pieces. For the preparation of sequencing libraries this is typically achieved by sonification, shearing, nebulisation or enzymatic reactions.

insertion

An insertion during sequencing describes the event where a nucleotide was inserted/added to the read that does not occur on the sequenced fragment.

k-mer

A k-mer is a substring of length k, i.e. a DNA sub-sequence of length k.

multidimensional scaling (MDS)

Multidimensional scaling is a multivariate technique for visualising the level of similarity between multiple data sets. A NxN distance matrix is computed containing pairwise comparisons of all N data sets. The MDS algorithm then projects these objects into a n dimensional space (n<N) while preserving the pairwise distances as well as possible. For n=2 this can be visualised in a two dimensional scatterplot where the distances between objects reflect their level of similarity.

nucleotides (dNTPs)

Nucleotides are single units of the bases A, C, G and T and constitute the building blocks of DNA.

polymerase chain reaction (PCR)

Polymerase chain reaction is a method for the amplification of specific DNA sequences. The target sequence is exponentially amplified and billions of copies can be produced. The method relies on the ability of DNA polymerases to synthesise new complementary strands of DNA. A preexisting 3'-OH group is necessary in order for the DNA polymerase to add new nucleotides. These are provided through the addition of primers to the reaction which bind to a specific region of the DNA. The polymerase then commences with the synthesis of a new complementary DNA strand from the end of the primer.

primer

Primers are short pieces of single-stranded DNA that are complementary to the target DNA sequence.

quality score

Quality scores are designed to predict the probability of an error in base calling during sequencing. They are defined as $Q = -10 \log_{10}(P)$ where P is the error probability. High quality scores imply more reliable base calls. Factors such as signal intensity profiles and signal-to-noise ratios are used by the sequencing machine to compute a quality score for each base call.

read

A read refers to the part of a single DNA fragment that is inferred during sequencing and returned by the sequencer.

ribosomal RNA (rRNA)

Ribosomal RNAs are important structural and catalytic components of the ribosome which is responsible for the synthesis of proteins from RNA (translation). They are integral parts of the small and large subunit of the ribosome. The 16S rRNA is part of the 30S subunit and the 5S and 23S rRNA are part of the 50S subunit.

ribosome

The ribosome is responsible for assembling proteins in a cell. It consists of a small and large subunit which are built up from rRNA and proteins. Prokaryotes possess 30S and 50S ribosome subunits. Each subunit is made up of specific ribosomal RNAs (rRNAs) and ribosomal proteins.

sequencing library

A sequencing library is a collection of DNA fragments that is compatible with the sequencing system to be used. Library preparation usually involves fragmenting the input DNA (if necessary) and attaching adaptors to the fragment end(s) that contain the necessary elements for immobilising the fragments on a solid surface and for sequencing.

substitution

A substitution error during sequencing describes the event of a miscall of a base where the nucleotide on the read differs from the sequenced nucleotide on the fragment.

transcription

During transcription a particular segment of DNA is transcribed into RNA by an enzyme called RNA polymerase.

translation

Translation is the process where proteins are synthesised from RNA. Messenger RNA is decoded by the ribosome in order to assemble specific amino acids which are later folded into proteins.

viral haplotype

Viral RNA polymerases lack the proof-reading ability of DNA polymerases resulting in high mutation rates in RNA viruses. The copies of the viral genome of a RNA virus therefore often differ from the original genome as they contain single nucleotide polymorphisms and are referred to as haplotypes.

1 Project Outline & Research Objectives

1.1 Research Objectives

Motivation

DNA sequencing has revolutionised many research areas, notably in the fields of health, microbial ecology and engineering, and has emerged as a powerful tool to solve problems that were hitherto intractable. Significant developments in sequencing technologies have provided fundamental knowledge on the human genome and revealed previously unknown levels of diversity and insight into the microbial world [128]. The number of bacterial cells in and on our body is ten times more than the number of human cells and comprises a much higher number of different genes. Large scale sequencing projects like the Human Microbiome Project (HMP) launched in 2008, aim to characterise the human microbiome with the goal to collect a total of 3,000 bacterial reference genomes plus several viral and small eukaryotic microbes isolated from the human body. We are only beginning to understand the effects of these human-related bacterial communities on our health and well-being. Sequencing is transforming medical research and biomedical engineering in many ways. The development of new drug treatments and vaccines as well as research on antibiotic resistance are only a few examples [159][103][123]. Research on the microbial ecology of simple water purification systems, such as slow sand filters, and waste water treatment, such as anaerobic digesters, have the potential to provide rural areas in developing countries with access to clean water and sanitation systems. Further, research in agricultural genomics has put perennial grain crops within reach [68][117]. Perennial grain crops could make agriculture more sustainable, by enabling longer seasons, reducing soil erosion with deeper routing and reducing the use of pesticides on agricultural lands [5][90]. Figure 1.1 highlights some of these applications.

Figure 1.1: Sequencing applications in different research areas: applications range from medical research and water purification, to vaccine development and agriculture. (Pictures taken from [27],[20],[5],[8].)

Rapid advances in sequencing technologies have made sequencing and metagenomics accessible to the majority of molecular microbiology laboratories. Simultaneously, the advent of high throughput sequencing technologies has made mathematics and computer science indispensable tools to answer questions in molecular biology marking the start of a new interdisciplinary research area which will continue to change our way of life. The development of statistics and bioinformatic algorithms, however, has not kept pace with the rapid development of the technology and many programs are not designed for metagenomic data sets and newly emerging sequencing platforms. A better understanding of systematic errors and biases introduced by these platforms is crucial for the analysis. Furthermore, a thorough understanding of the capacities and limitations of available bioinformatic programs is vital in order to derive accurate hypotheses and correctly interpret results. In order to determine the accuracy of the results obtained from currently available bioinformatic algorithms, appropriate and realistic *in silico* data sets that reflect platform specific peculiarities are essential.

Main aims and objectives

The aim of this research was to explore the potential of resolving fine-scale variation in next generation sequencing (NGS) data. This includes artificial variation, such as errors and biases that are introduced during the library preparation and the sequencing process, as well as true variation that occurs in the form of single nucleotide polymorphisms (SNPs). Sequencing can provide a wealth of information about different organisms, however, in order to access this information we first need to identify and address biases and errors in the data and recognise true variation. In the following I will outline the objectives that were set to achieve this aim. Figure 1.2 provides an overview and indicates the respective chapters where these objectives were addressed.

Simulation tools:

- Development of tools that offer flexible and realistic simulations of fine-scale variation in microbial communities and viral haplotype populations:
  - Design and implementation of a NGS read simulation program that can accurately reflect artificial fine-scale variation in Illumina sequencing data and is able to simulate reads based on complex population structures.
  - Developing a viral quasi-species simulation program that mimics the evolution of a single virus into a haplotype population.

Benchmarking studies:

- These simulation programs will form the basis for two comprehensive benchmarking studies:

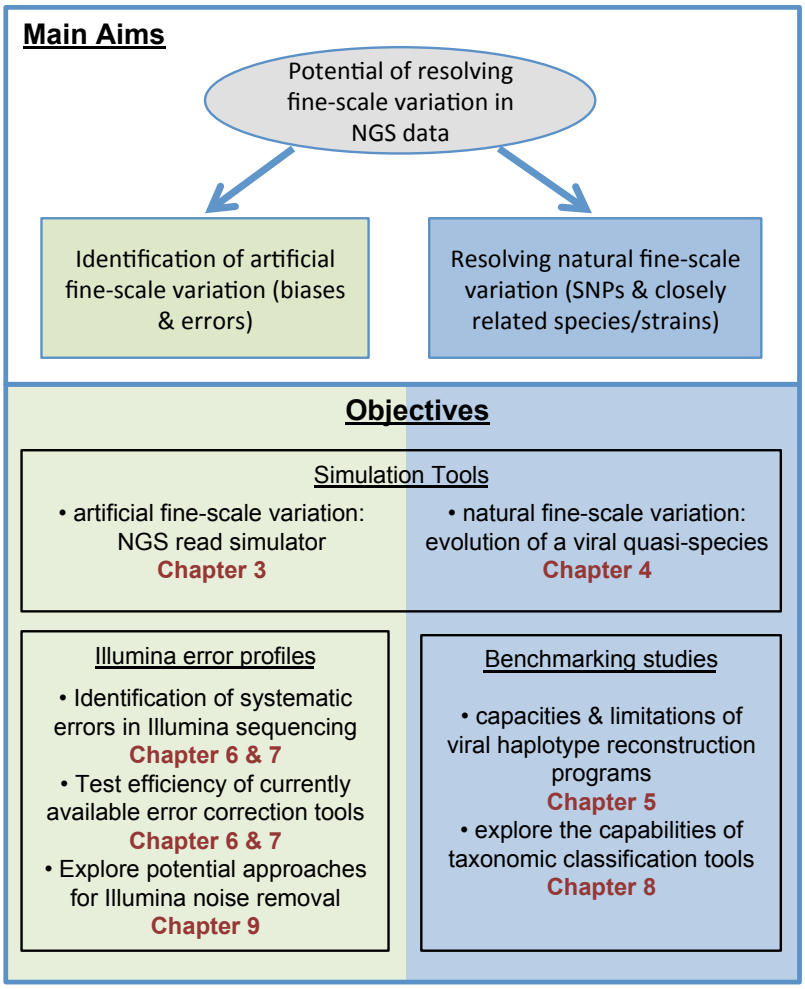


Figure 1.2: The figure summarises the main aims and objectives of my Ph.D. project and indicates the respective chapters where these objectives were addressed.

  - Identification of the advantages and limitations of currently available haplotype reconstruction programs based on a large range of *in silico* test data sets.
  - Validation of taxonomic classification tools based on *in vivo* metagenomic data sets and exploring the ability of the programs to correctly assign new organisms at various taxonomic levels.

Illumina error profiles:

- Characterisation of biases and error patterns in Illumina sequencing data:
  - Identification of systematic error patterns and biases based on a large *in vitro* study for amplicon as well as metagenomic sequencing data.
  - Testing the efficiency of currently available error correction and removal tools.
  - Exploring potential approaches for effective Illumina noise removal algorithms that are capable of addressing systematic errors.

1.2 Overview of Chapters

The following section will provide an overview of this dissertation with a short summary of each individual chapter.

Chapter 1 outlines my PhD project and highlights the importance of bioinformatic research with a brief description of the scope of applications. I also illustrate the connection of viral haplotype reconstruction and metagenomics, which are two focal points of my research. The chapter continues with an overview of my thesis chapters and concludes with a list of publications based on the work I carried out during my PhD.

Chapter 2 provides a brief history of the discovery of DNA followed by an overview of first and next generation sequencing technologies and an outlook on the development of third generation sequencing. This section continues by highlighting the importance of bioinformatics in realising the promise and potential of DNA sequencing and illustrates the impact and range of applications of sequencing.

Chapter 3 discusses the development of a flexible read simulation program for amplicon and metagenomic data sets that is capable of reflecting the peculiarities of Illumina sequencing platforms. This is the first available program that can explicitly simulate the impact of various experimental factors. I provide a large range of pre-computed Illumina error profiles covering a variety of experimental design factors to facilitate rapid simulations without the need for prior computations based on existing experimental data sets. I also supply various empirical insert size distributions for paired-end simulations and the user can simulate reads based on complex microbial communities under a user-defined abundance profile. In addition, my program can simulate Roche 454 and noise-free reads.

Chapter 4 highlights the importance of complex and realistic *in silico* data sets for the validation and benchmarking of bioinformatic programs. An important achievement during my PhD was the development of a complex algorithm to simulate an *in silico* viral haplotype population based on an empirical data set. Haplotypes differ by only a few single nucleotide polymorphisms (SNPs) and present a major difficulty for the immune system. Thus it is crucial to identify the viral diversity for vaccine and drug development. My algorithm infers a possible set of haplotypes, which conforms to the observed nucleotide frequencies in the empirical data set at every single position of the genome. I applied this algorithm to a foot-and-mouth virus data set to simulate an *in silico* population for my benchmarking presented in Chapter 5.

Chapter 5 presents the only available independent benchmarking study testing the accuracy and capacity of haplotype reconstruction programs in the context of populations of varying size and complexity. The simulations were based on the programs presented

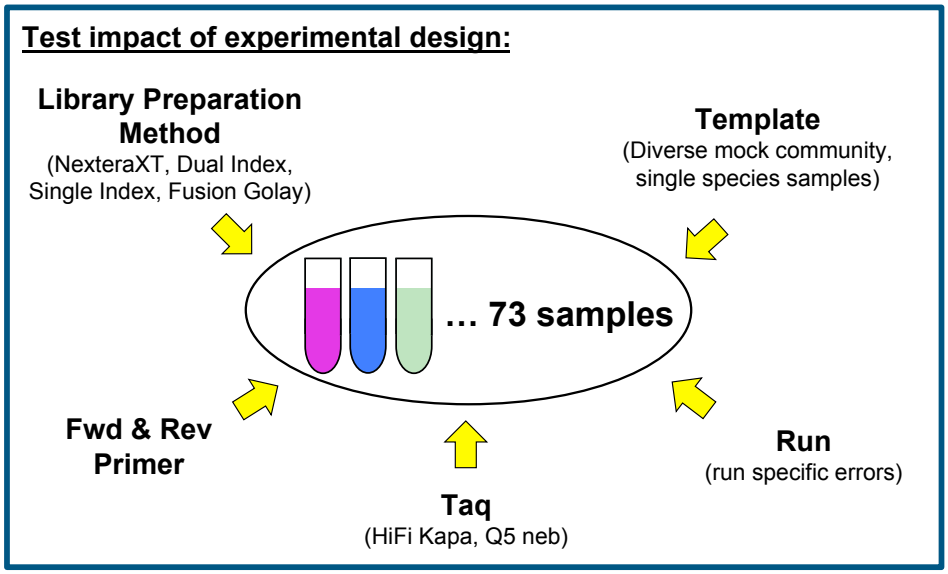


Figure 1.3: Overview of experimental design factors tested for Illumina amplicon data sets.

in Chapter 3 and Chapter 4. Furthermore, I developed a complex statistical framework for the evaluation that takes the number of successfully reconstructed haplotypes into account as well as the false positives by determining the number of mismatches compared to the closest true haplotype. My benchmarking demonstrates the limitations of the read graph approach, which forms the basis for the majority of haplotype reconstruction programs, and highlights the advantages of programs based on probabilistic models. However, I showed that none of the available programs was able to resolve low sequence divergence and all programs failed to recover rare haplotypes.

Chapter 6 discusses the necessity of a better understanding of biases associated with the different sequencing technologies as well as experimental factors in order to unlock the true potential of next generation sequencing technologies. I conducted the largest *in vitro* study on biases and error patterns to date and for the first time a connection between error patterns and experimental design factors was established. Figure 1.3 gives an overview of the experimental factors that I tested for amplicon sequencing on the Illumina MiSeq platform. I computed the position and nucleotide specific error patterns and showed that the library preparation method and the choice of primers are the most significant sources of bias causing distinct error patterns. In addition, I tested the efficiency of different error removal techniques and identified read overlapping as the most effective approach with further improvements if the reads are quality trimmed and error corrected prior to overlapping. These motif-based biases were also implemented in the current version of my read simulation program presented in Chapter 3.

In **Chapter 7**, I extended my study on error profiles to metagenomic sequencing. I tested additional Illumina sequencing platforms and included more library preparation

methods. Biases associated with limitations due to the sequencing chemistry and technology were revealed, as well as biases associated with the transposon technology. Furthermore, I tested different strategies for error removal for all three Illumina platforms.

Chapter 8 presents the results of my validation study of taxonomic classification algorithms for *in vivo* bacterial metagenomes. I start with an overview of the taxonomic classification tools developed in Alice McHardy's group, followed by the outline of the measurements for performance evaluation. The validation results demonstrate the advantages and disadvantages of the different approaches and highlight the impact of the various databases.

In **Chapter 9**, I introduce a sophisticated error correction algorithm for Illumina amplicon data sets that is based on a collapsed variational Dirichlet process mixture model. Error correction is a crucial step during the analysis of next generation sequencing data. However, due to the poor knowledge of systematic errors in Illumina data sets there is currently no established error correction method. My algorithm incorporates a nucleotide and position specific model to accommodate the peculiarities encountered in Illumina data that were identified in Chapter 6. The variational inference approximation facilitates computations for millions of reads from high throughput Illumina platforms.

Chapter 10 is the final chapter where I summarise the impact of my research and outline how this work will be continued in ongoing and future projects. I conclude this chapter with a brief outlook on the prospect of sequencing as well as bottlenecks that will be faced in the near future.

1.3 Overview of Publications

Journal Papers

M. Schirmer, W. T. Sloan and C. Quince, Benchmarking of viral haplotype reconstruction programs: an overview of the capacities and limitations of currently available programs. [Briefings in Bioinformatics, 2012]

J. Alneberg, B. S. Bjarnason, I. de Bruijn, **M. Schirmer**, J. Quick, U. Z. Ijaz, L. Lahti, N. J. Loman, A. F. Andersson and C. Quince, Binning metagenomic contigs by coverage and composition. [Nature Methods, 2014]

S. Haig, **M. Schirmer**, L. D'Amore, J. Gibbs, R. Davies, G. Collins and C. Quince, Stable-Isotope Probing and Metagenomics Reveal Predation by Protozoa Drives *E.coli* Removal in Slow Sand Filters. [ISME Journal, 2014]

Papers in Review

M. Schirmer, U. Z. Ijaz, L. D'Amore, N. Hall and C. Quince, Insight into Biases and Sequencing Errors for Amplicon Sequencing with the Illumina MiSeq Platform. [Nucleic Acid Research: In review]

J. M. Couto, U. Z. Ijaz, V. R. Phoenix, **M. Schirmer** and W. T. Sloan, Metagenomic sequencing of sediment samples from a subarctic lacustrine environment reveals de novo oxygen tolerant NiFe membrane bound hydrogenases that might be exploited in hydrogen technologies. [PLOS ONE: In review]

I. Gregor, J. Dröge, **M. Schirmer**, C. Quince and A. C. McHardy, PhyloPythiaS+: A self-training method for the rapid reconstruction of low-ranking taxonomic bins from metagenomes. [PLOS Computational Biology: In review]

Papers in Preparation

L. D'Amore, U. Z. Ijaz, **M. Schirmer**, N. Hall and C. Quince, Comparing next generation sequencing platforms and library preparation methods. [In preparation: Genome Biology]

M. Schirmer, W.T. Sloan and C. Quince, MicroSim: A motif-based next-generation sequencing simulator. [In preparation: BMC Bioinformatics]

M. Schirmer, U. Z. Ijaz, L. D'Amore, N. Hall and C. Quince, Metagenomics: Identification of error patterns in Illumina data and optimal processing strategies. [In preparation]

M. Schirmer, S. Haig and C. Quince, A pipeline for analysing and comparing the community structure of multiple metagenomic samples. [In preparation]

M. Schirmer, T. Abdelrahman, S. Al-Otaibi, C. Quince and E. Thomson, Exploring the Potential of Novel Sequencing Technologies for Viral Haplotype Reconstruction. [In preparation]

M. Schirmer, U. Z. Ijaz and C. Quince, Noise Removal in Illumina Amplicon Data using a Collapsed Variational Dirichlet Process Mixture Model. [In preparation]

Conference Publications & Talks

M. Schirmer, U. Z. Ijaz, L. D'Amore, N. Hall, W. T. Sloan and C. Quince, Metagenomic data analysis: Identification of error patterns in Illumina data and optimal processing strategies. Conference: ISME 2014. [Talk, Aug 2014]

M. Schirmer, U. Z. Ijaz, L. D'Amore, N. Hall, W. T. Sloan and C. Quince, Validation

of State-of-the-Art Library Preparation Methods for Illumina Amplicon Sequencing. Conference: NGS 2014. [Poster]

M. Schirmer, L. D'Amore, N. Hall, W. T. Sloan and C. Quince, Insights into Biases and Sequencing Errors of the Illumina MiSeq Platform. Conference: UK Genome Science Meeting. [Poster]

M. Schirmer, Viral Diversity Estimation. Conference: Metagenomics in Virology. [Talk]

M. Schirmer, L. D'Amore, N. Hall and C. Quince, Error Profiles for Next Generation Sequencing Technologies. Conference: The Next NGS Challenge. [Poster]

M. Schirmer, Benchmarking of Viral Haplotype Reconstruction Programs. Conference: Bertinoro Computational Biology. [Talk]

M. Schirmer, Dirichlet Process Mixture Models, Haplotype Reconstruction and Validation of Taxonomic Classification Algorithms. Seminar at the University of Dusseldorf. [Talk]

M. Schirmer, W. T. Sloan, D. Taylor and C. Quince, Benchmarking of Viral Haplotype Reconstruction Programs. Conference: RECOMB. [Poster]

2 Background and Introduction to DNA Sequencing

2.1 The Discovery of DNA

Two names that are often-quoted for the discovery of DNA are Watson and Crick. However, the cornerstone for their ground-breaking model was founded much earlier by the Swiss chemist Friedrich Miescher in the late 1860s. He was the first to identify deoxyribonucleic acid (DNA) in the nucleus of human white blood cells. After that, almost 60 years passed until the Russian scientist Phoebus Levene discovered that nucleic acids were composed of a series of nucleotides. He also proposed that each nucleotide in turn consists of a phosphate group, one of four nitrogen-containing bases and a sugar molecule which is now known to be either a ribose (in the case of RNA) or a deoxyribose sugar (in the case of DNA). Building on their work, Chargaff, an Austrian biochemist, showed in 1950 that the nucleotide composition of DNA varied among species and he found that adenine (A) and thymine (T) as well as guanine (G) and cytosine (C) were usually present in similar quantities. He did not realise yet that DNA is encoded as complementary strands. It was only a few years later, in 1953, that Watson and Crick published their fundamental model that describes a DNA molecule as a 3-dimensional double helix with two complementary strands held together by hydrogen bonds. Their model also characterises DNA double helices as right-handed and anti-parallel (the 5' end is paired with the 3' end). In addition, it shows that the outer edges of the nitrogen-containing bases are exposed and thus provide access through potential hydrogen bonding for other

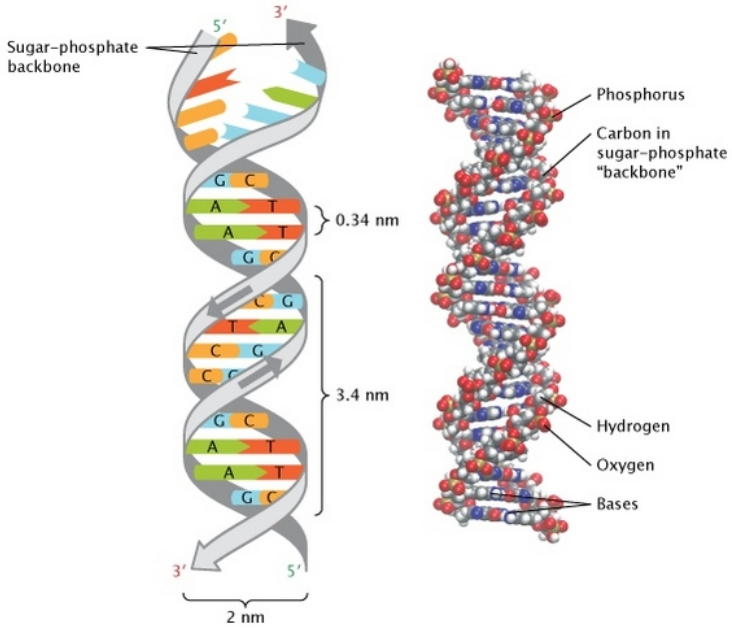


Figure 2.1: The DNA double helix: two complementary strands consisting of the four nucleotides A, C, G and T. (Taken from [126].)

molecules such as proteins. Their complex model of the structure of DNA (see Figure 2.1) still remains the same today. [126][158][95][49]

Thirty years later, advances in RNA sequencing enabled Walter Fries and colleagues to sequence the first gene in 1972 [83]. And just a few years after that the first DNA sequencing methods were developed that became widely used over the following decades. Since then, DNA sequencing technologies have fundamentally changed and influenced research in areas such as genetics, microbiology, biotechnology, engineering and forensics.

Cultivable bacteria have been studied in laboratories for many years, but it is estimated that 99% of all bacteria are not cultivable.[124][92] Thus the vast majority is still unexplored since classical methods cannot be applied. The study of microbial biodiversity is still in its infancy. Metagenomics provides a new approach to this problem. Metagenomics is the study of genetic material from environmental samples, a new research field made widely available with the advent of next generation sequencing, that provides novel and fundamental insight into various ecological systems. Those systems include soils, oceans and the human gut or skin, many of which were previously unstudied.

Table 2.1: Overview of sequencing technologies with their commercial launch date and current maximum read length (2014). [112]

First Generation Sequencing	1977	Sanger Sequencing	$\leq$ 1,000bp
	1977	Maxam-Gilbert Method	$\leq$ 100bp
Second Generation Sequencing	2005	454 Life Sciences	$\leq$ 1,000bp
	2005	Polony Sequencing	$\leq$ 2x13bp
	2006	Illumina	$\leq$ 2x300bp
	2007	SOLiD	$\leq$ 2x35bp
	2009	Complete Genomics	$\leq$ 2x35bp
	2011	Ion Torrent	$\leq$ 200bp
Third Generation Sequencing	2008	Helicos SMS	$\leq$ 100bp
	2011	PacBio	$\leq$ 30,000bp
	2014 (*)	Oxford Nanopore Technologies	$\leq$ 40,000bp (**)
	2014 (*)	Genia	(***)

* Announced launch date.
** Maximum read length presented in [100].
*** Information not available yet.

2.2 The Past, Present and Future of DNA Sequencing

The first fundamental methods for DNA sequencing were developed in the 1970s. Since then, the methodology and technology have advanced rapidly, transforming research in many areas. Table 2.1 gives an overview of the currently available methods that will be outlined below.

First Generation Sequencing

Sanger sequencing, an enzymatic method using DNA polymerase, was first published in 1975 by Sanger and Coulson [137]. Two years later they introduced a more efficient and easier chain termination method [138] that employs radioactive or flourescently labeled dideoxynucleotides (ddNTP) acting as chain terminators. Around the same time Maxam and Gilbert presented a non-enzymatic method involving less complex preparations for the sequencing but produced shorter reads [107].

Sanger Sequencing

The Sanger method works with chain-termination. First the DNA is purified and denatured followed by bacterial cloning or polymerase chain reaction (PCR) amplification. PCR was first introduced in the 80s and significantly shortened the process of DNA amplification. After amplification the solution is divided into four tubes and the polymerase enzyme is added, catalysing the synthesis of new DNA strands from the template DNA. To each tube one of the four dideoxynucleotides is added as well as the normal nucleotides. ddATP, ddGTP, ddCTP and ddTTP terminate the DNA strand extension if one of them is incorporated during polymerase instead of the normal nucleotides A, G, C or T. Therefore, there is a random chance that the polymerase stops at any given position and results in fragments of varying length. As each tube only contains one ddNTP, we know the identity of the last incorporated nucleotide in each fragment. Again the double stranded DNA is denatured and then separated by size with gel electrophoresis. One lane is used for each of the four nucleotides (see Figure 2.2) and provides the necessary information to infer the nucleotide sequence of the complementary DNA strand. At the position marked in Figure 2.2 this would be "CGAT". Several improvements have been implemented since the method was first introduced. The process was, for example, automated by labelling the four ddNTPs with different colours which facilitates running all reactions in a single tube. Also a laser can be used to auto-

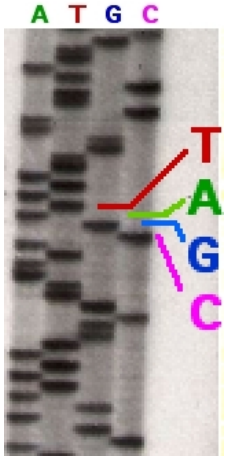


Figure 2.2: Sanger sequencing: gel electrophoresis of DNA. (Figure taken from [6].)

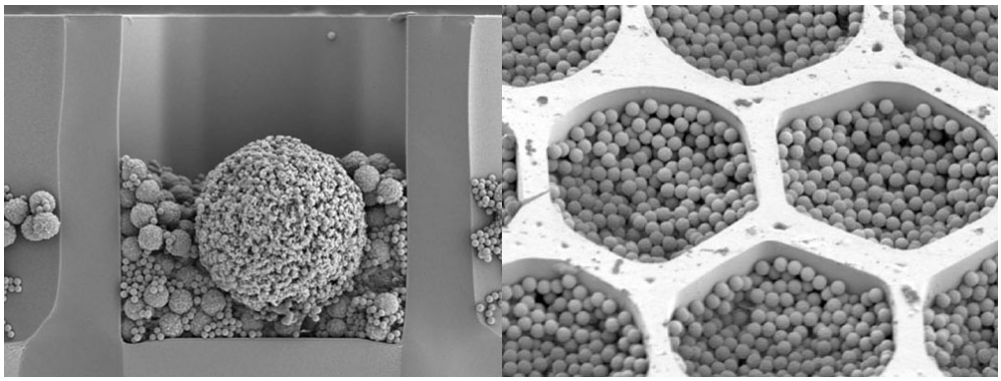


Figure 2.3: 454 pyrosequencing: each well can hold one bead (left). The plates contain luciferase (right) which is the enzyme that catalyses the light-emitting reaction. (Image taken from [2].)

matically detect the signal. This method can currently produce reads of up to 1,000bp. Long reads and low error rates still remain an advantage of Sanger sequencing today.

The Maxam-Gilbert Method

The Maxam-Gilbert method involves chemical modification of the DNA sequence. Breaks are caused by different chemical reactions each splitting a specific base or pair of bases (G, A+G, C, C+T). This results in reads of up to 100bp. Due to the need of hazardous chemicals and the lack of possible automatisation of the process the method is rarely used nowadays.

Low throughput and high cost-per-base are the main limitations of first generation sequencing. This makes the detection of low frequency variants very expensive. However, Sanger sequencing is still utilised for applications where high throughput is not required and long reads combined with very low error rates [36] are essential.

Second/Next Generation Sequencing (NGS)

The main advantage of NGS is the potential for massive parallelisation and automation, making large scale sequencing projects possible. The high throughput produced by these technologies provided new research opportunities such as sequencing microbial communities and environmental samples.

454 Pyrosequencing

Pyrosequencing was developed in 1996 and has been commercially available since 2005. The method is based on a principle called sequencing-by-synthesis. First the double stranded DNA is denatured and fragmented. During 454 sequencing each individual fragment is attached to a bead which is enclosed in a water droplet within an oil phase. PCR amplification coats the beads with copies of the respective fragment. The beads are then localised in wells on a plate. This process is also referred to as emulsion

PCR (emPCR). Nucleotides are then sequentially flowed over the plate, one type of nucleotide at a time. Multiple nucleotides can get incorporated at the same time if a homopolymer is encountered at the current synthesis position of the template DNA fragment. The addition of each nucleotide causes the release of pyrophosphate which provides the energy for the enzyme luciferase to produce light. The strength of the light signal can be used to infer the number of nucleotides that were added. This step is where most 454 sequencing errors arise resulting in significantly higher error rates in homopolymeric regions of length three or more.

This process is extremely fast since over one million wells can be treated simultaneously. Hence, over one million DNA fragments can be sequenced in parallel. Pyrosequencing started off with read lengths of up to 100bp [142]. Nowadays, 454 sequencing technologies can achieve read lengths of up to 1,000bp and as much as one million reads [1]. 454 pyrosequencing has dominated the sequencing market for many years. But as other sequencing methods become more established with lower error rates, lower cost-per-base and comparable read lengths, Roche decided to withdraw the GS FLX 454 pyrosequencing platform from the market by mid 2016.

Illumina

Illumina sequencing is another sequencing-by-synthesis method and was first available in 2006. During the library preparation the DNA is fragmented and tagmented. The adapter that is added to the template during the tagmentation includes binding sites for the sequencing primers, optional indices to label the sample and enable multiplexing, and complementary oligos that allow the fragment to bind to the flow cell. The flow cell is a glass slide that is coated with a lawn of two types of oligos. The single stranded DNA molecules bind to the flow cell and the complementary strands are synthesised.

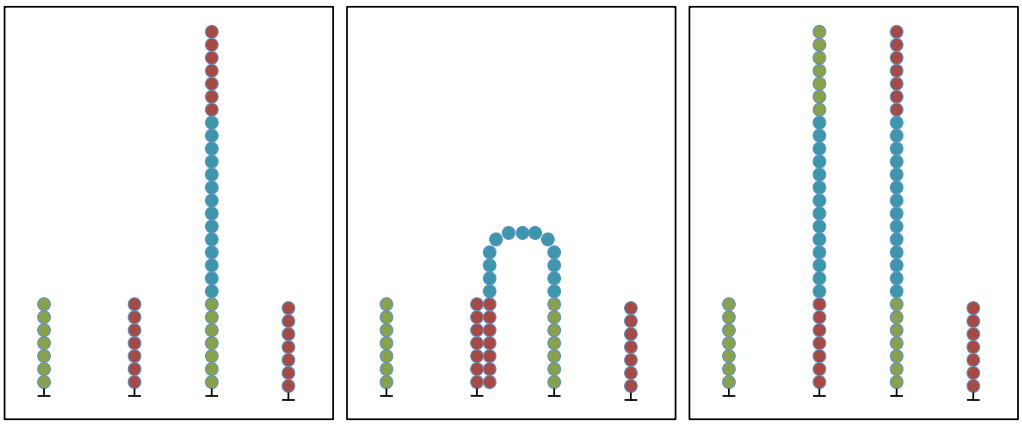


Figure 2.4: Bridge amplification: the fragment is bound to the flow cell (left), bends over and hybridises to complementary oligos on the surface (middle). The DNA fragment is then copied and the double stranded DNA is denatured, resulting in two single stranded copies of the fragment (right).

The double stranded DNA is subsequently denatured and the original fragments are washed away. This is followed by a process called bridge PCR amplification. The company Solexa initially developed this technique in the 90s and was later acquired by Illumina. During bridge amplification the fragments bend over and hybridise to the second type of oligos on the flow cell. Following the synthesis of the complementary strand, the double stranded bridge is then denatured resulting in two single stranded copies of the fragment which are both bound to the flow cell. (See Figure 2.4 for more details.) The iteration of this process produces dense clusters of copies around each initial fragment (see Figure 2.5). Finally the reverse strands are cleaved off and washed away. [22]

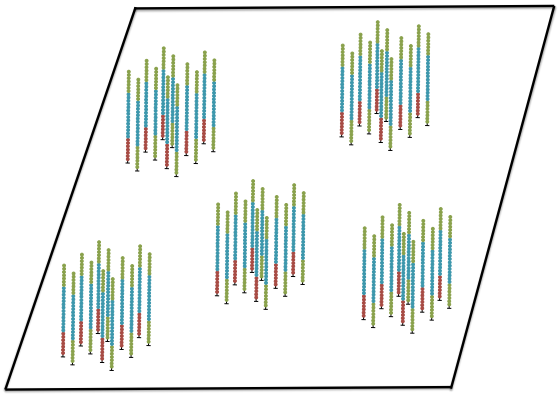


Figure 2.5: Cluster generation during Illumina sequencing.

Now, sequencing of the forward strands can commence with the extension of the first sequencing primer (see Figure 2.6). Fluorescently labeled reversible terminator-bound dNTPs are used for the polymerisation. Only one base is added in each cycle due to the 3' termination of the incorporated nucleotide. Hence, the number of cycles coincides with the read length. Two lasers are used to excite the dye attached to each nucleotide. The same laser is used for A/C (red) and G/T (green), respectively. The emission

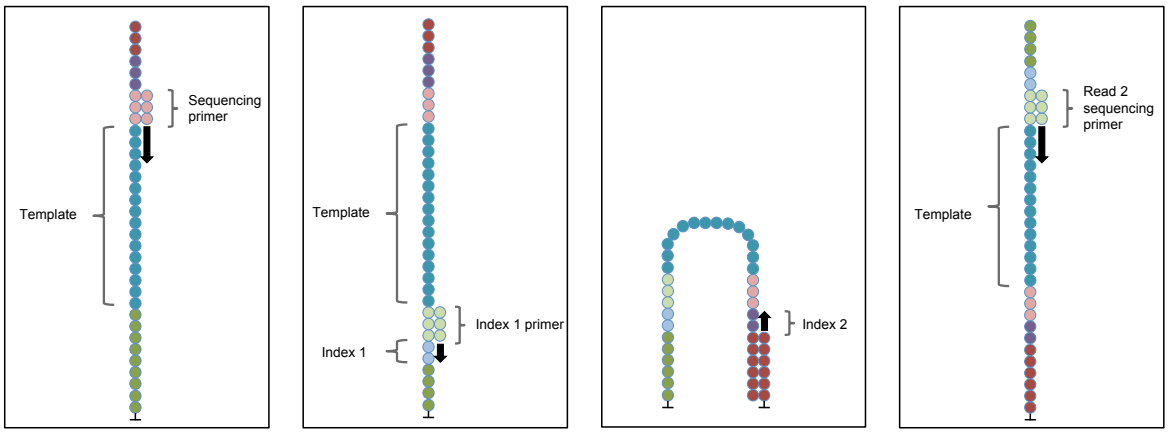


Figure 2.6: Sequencing-by-synthesis: The sequencing of the forward strand commences with the extension of the first sequencing primer (a). After that the Index 1 primer allows the synthesis of Index 1 for subsequent multiplexing. The fragment then folds over and binds to the flow cell. Next, Index 2 is read (c) followed by the synthesis of the complementary strand of the fragment. The double-stranded DNA is denatured, the forward strand is cleaved off and washed away. Lastly, the read 2 sequencing primer allows the synthesis of the reverse read (d).

spectra is recorded and the nucleotides are identified based on the signal intensity. All fragments within a cluster are read simultaneously and enhance the base calling signal. The signals of hundreds of millions of clusters are read concurrently enabling massive parallelisation. After the completion of the first read, the read product is washed away. Next, the Index 1 read primer is introduced and hybridised to the template to record Index 1 (Figure 2.6b). The product is again washed away and the template folds over and binds to the oligos on the flow cell. Index 2 is read next (Figure 2.6c). Afterwards the complementary strand is synthesised, the double stranded DNA is denatured and the forward strand is washed away (Figure 2.6d). Now, sequencing of the reverse strand can commence. [22]

Illumina started off with very short reads. However, Illumina has evolved as the market leader over recent years with improved read lengths of up to 2x300bp, true paired-end reads, higher throughput than 454 and lower cost-per-base. As only one base is read at a time there are no issues related to homopolymers. The main source of errors for Illumina sequencing are substitution errors as will be further discussed in Chapter 6.

Over the years more and more sequencing technologies have entered the market but only occupy a comparably small percentage of the market share. EmPCR forms the basis for most sequencing technologies including 454, Polony sequencing, SOLiD and IonTorrent.

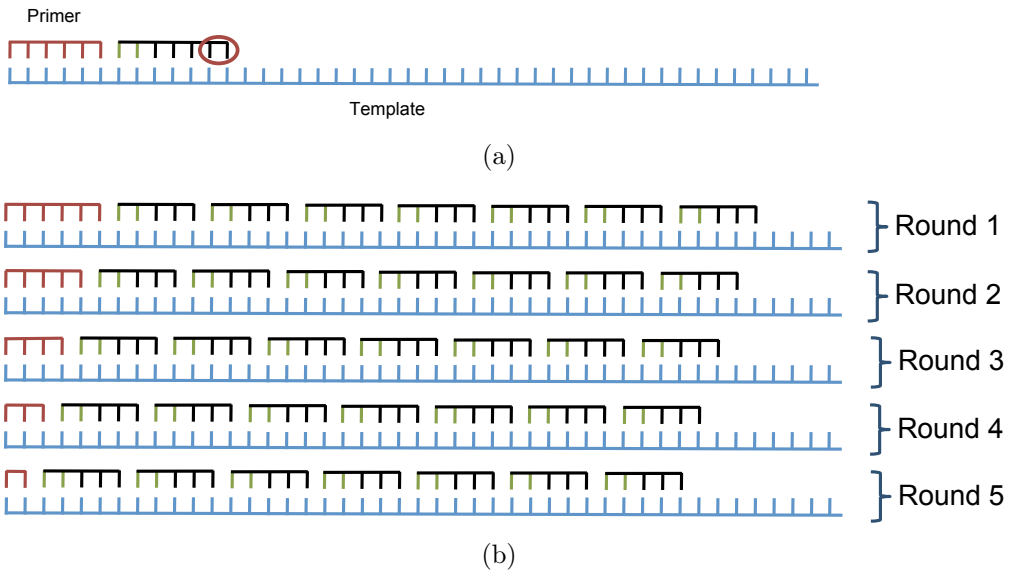


Figure 2.7: SOLiD sequencing-by-ligation: (a) Di-base probes are ligated to the DNA template. The first two bases (indicated in green) are interrogated. The fluorescent marker is attached to the last two bases and cleaved off after ligation. (b) Seven di-base probes are ligated to the template DNA. This process is repeated over five rounds where the primer is offset by one base in each round resulting in dual measurements of each base.

Polony sequencing

Polony (polymerase-colony) sequencing is an open-source sequencing chemistry developed at Harvard Medical School. Although producing millions of reads the range of applications for this technology is limited due to read lengths of only 2x13bp. Polony sequencing is implemented in the Danaher Motion Polonator G007 platform. The method is not used widely but has aided in establishing other sequencing chemistries including SOLiD.

SOLiD

SOLiD (Sequencing by Oligonucleotide Ligation and Detection) uses a sequencing-by-ligation approach. Fluorescently labelled di-base probes of eight nucleotides in length compete for ligase binding to the DNA fragment. The probes are labelled with four dyes and each dye represents four possible nucleotide sequences. After a di-base probe is ligated to the template DNA the dye is cleaved off (last two bases, see Figure 2.7a). This is repeated over seven cycles synthesising 35bp in total. The process is repeated over five rounds. In each round the previously synthesised strand is removed and a new primer is hybridised offset by one base. This provides dual measurements of each base (see Figure 2.7b).

The colour scheme needs to comply with a set of rules in order to unambiguously infer the nucleotide sequence (see [14] for details). Suppose we want to sequence the following DNA fragment "GTACTAGGAC" with the following base colour scheme [14]:

dye	0	1	2	3
	AA	AC	AG	AT
	CC	CA	GA	TA
	GG	GT	CT	CG
	TT	TG	TC	GC

This would result in the sequencing output (including information on the first base): "G131232021". The above table can also be re-written such that the rows indicate the first base and the columns indicate the second base:

	A	C	G	T
A	0	1	2	3
C	1	0	3	2
G	2	3	0	1
T	3	2	1	0

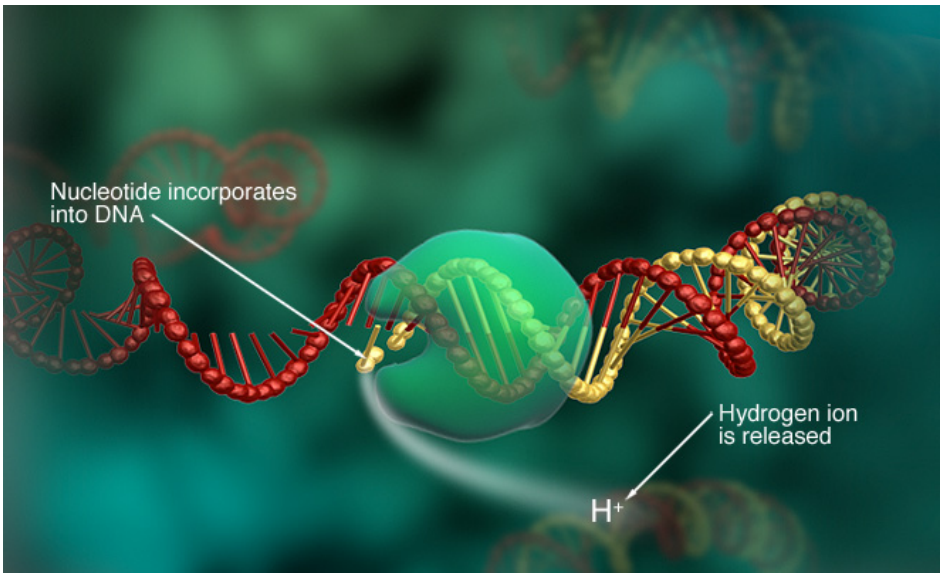


Figure 2.8: The Ion Torrent sequencing technology. (Figure taken from [23].)

Knowing that the first base is a "G" and the first recorded colour is "1", the second base must be "T". Knowing that the second base is a "T" and the second colour is "3", the third base must be "A". This can be continued to infer the complete sequence. Low operating costs need to be balanced with very short read length.

Ion Torrent

Ion Torrent employs semiconductor sequencing. The method is similar to pyrosequencing but rather than measuring light emissions the hydrogen ions are monitored. The approach is implemented in two platforms: the Ion PGM and the Ion Proton. Each time a base is incorporated into a DNA strand by polymerase a hydrogen ion (H^+) is released (see Figure 2.8). Deoxynucleotides (A, C, G or T) are sequentially flowed over the microwells. The release of the hydrogen ion leads to a change in the pH inside the microwells which is detected by an ion sensor. This allows the omission of the imaging step and facilitates shorter sequencing times. Similar to pyrosequencing, multiple nucleotides can be added at once if homopolymers are encountered. The ion release is proportional to the number of bases added. For Ion Torrent the majority of errors are also associated with homopolymeric regions. Both, the Ion PGM and Ion Proton, achieve average read lengths of around 200bp [112]. The platforms produce relatively short reads and higher error rates compared to Illumina [54] .

Complete Genomics

Complete Genomics is a life science company that developed a sequencing platform intended for human genome sequencing. The technology relies on DNA nanoball sequencing and has been commercially available as a service since 2009. The DNA is isolated and then fragmented. Adapter sequences (Adapter 1) are then ligated to

both ends of the fragments followed by PCR amplification. The adapter sequences are then modified to create complementary sequences to allow circularisation of the fragment. The DNA is cleaved 13bp to the right of Adapter 1 and the process is repeated with Adapter 2. This time the resulting circularised product is cleaved 13bp to the left of Adapter 1 and Adapter 3 is ligated to the sequence. After circularisation the DNA is now cleaved at two positions (26bp left of Adapter 3 and 26bp right of Adapter 2) followed by the addition of Adapter 4. The final product is circular and contains all four adapter sequences (see Figure 2.9). The circular fragments are copied by rolling circle replication resulting in long single stranded strings of DNA comprising several copies of the circular template. The single strand folds onto itself due to palindromic sequences in the four adapter sequences forming a tight ball of DNA. These nanoballs are then attached to microarray flow cells.

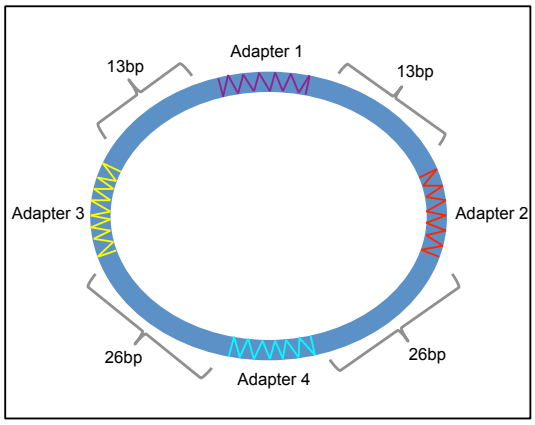


Figure 2.9: DNA nanoball sequencing: final circular product containing all four adapter sequences.

The probe-anchor ligation sequencing chemistry allows to read up to 10bp adjacent to each site of the four adapters. The anchor oligonucleotides are complementary to the adapter ends and bind to the template DNA. The probes are fluorescent 10mer DNA sequences with degenerated nucleotides in all but one position. The process is repeated five times with different probes to determine position one to five. The second adapter contains five degenerate nucleotides and is used to determine position six to ten. The addition of the degenerate nucleotides limits the distance between the nucleotide of interest and the ligation point of the sequencing and anchor probes. This step reduces errors as the fidelity of the ligase decreases with the distance from the ligation point. The sequencing yields mate-paired reads of 2x35bp. Again, the main limitation for this method is the short read length. [58]

Third-Generation Sequencing

The third generation of sequencing technologies engineer real time single-molecule sequencing methods with the aim of achieving higher throughput and longer read lengths. This eliminates the need for PCR amplification and should facilitate faster sequencing times. Most of these methods are currently under extensive development.

Helicos Single Molecule Sequencing (SMS)

Helicos SMS was the first commercially available single molecule fluorescent platform.

The DNA is fragmented and denatured into 100-200bp strands. A polyA tail is added to the 3' end of each template strand and in addition the strands are labelled with a fluorescent nucleotide which is attached to the polyA tail. The templates are then hybridised to a flow cell containing oligo T universal capture sites that are immobilised onto the flow cell surface. Templates can be packed at a very high density as single molecules are detected, providing high throughput. The flow cell surface is then illuminated with a laser showing the location of each fluorescently labelled template. The labels are removed after the location of templates are imaged. When the DNA polymerase commences, flourescently labelled terminating nucleotides are consecutively flowed over the flow cell surface and emit a light signal if a nucleotide is incorporated. The remaining nucleotides are washed away and after the imaging step the flourescent labels are removed before the next type of nucleotide is flowed over the cell surface. Every strand is sequenced independently avoiding problems related to phasing and prephasing observed in technologies that utilise an amplification step to generate clusters. Helicos SMS produces billions of reads and avoids PCR-induced errors. But due to short reads of 33-100bp [24] and relatively high error rates (substitution rate 0.2%, indel rate 1-3% [151]) its applications are limited.

Pacific Biosciences

Pacific Biosciences developed a sequencing platform using Single Molecule Real Time (SMRT) sequencing. In contrast to second generation sequencing it is not necessary to pause the DNA polymerase to detect the incorporated nucleotide. Each SMRT cell contains tens of thousands of chambers, so called zero-mode waveguides, which are illuminated from below. A waveguide is a physical structure which guides the electro-

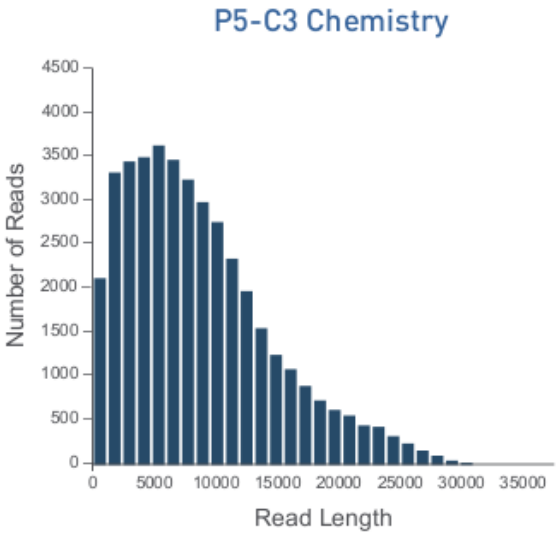


Figure 2.10: PacBio read length distribution (taken from [7]): with the P5-C3 chemistry average read lengths of 8.5kb can be achieved. The throughput per SMRT cell is ≈ 375 Mb.

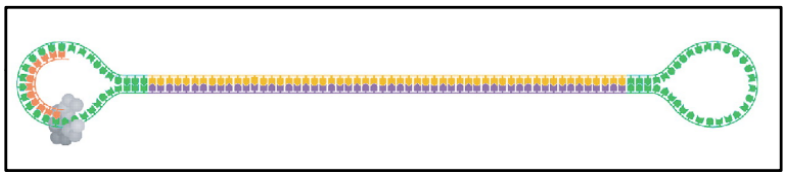


Figure 2.11: PacBio SMRTbell template (adapted from [152]): two single stranded hairpin adapters are added to both ends of the double stranded DNA template. These hairpin adapters contain the priming site for the polymerase enzyme.

magnetic waves of the light and the small aperture causes the optical field to decay exponentially inside the chamber. This results in a very confined observation volume which enables the machine to directly monitor the activity of the DNA polymerase and to detect the incorporation of each individual nucleotide. In this manner thousands of single-molecule sequencing reactions can be monitored simultaneously.

A DNA polymerase complex is immobilised at the bottom of each chamber and is used to sequence a single molecule of DNA. Fluorescently labelled nucleotides are introduced to the chambers during the polymerase. Here, the fluorescent labels are attached to the terminal phosphate of the nucleotides. The label is automatically clipped off by the polymerase enzyme and the emission of the light is detected by a sensor. The colour indicates the incorporated nucleotide.

According to Pacific Biosciences the PacBio RS II systems can produce reads with an average read length of $\approx$ 8.5 kbp with the longest reads reaching lengths of over 30 kbp (see Figure 2.10). Each SMRT cell can yield about 50,000 reads [7]. High error rates of $\approx$ 11% [3] have so far limited the application of PacBio reads. By introducing circular consensus sequencing (CCS) these error rates can be greatly reduced. For CCS,

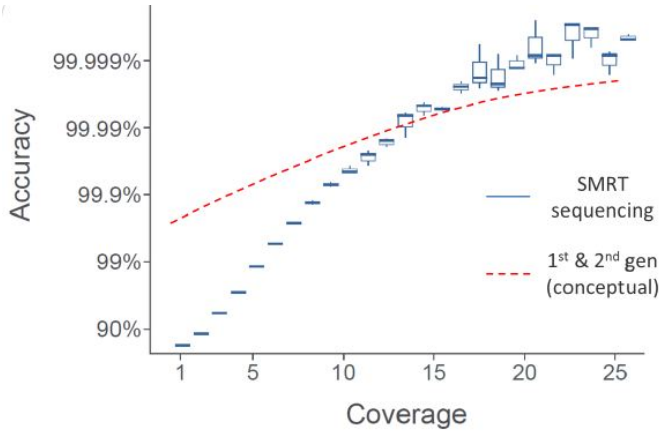


Figure 2.12: PacBio error rates: the initial error rates of $\approx$ 11% can be significantly reduced with circular consensus sequencing. The figure shows how the accuracy improves with the number of passes. (Figure taken from [3].)

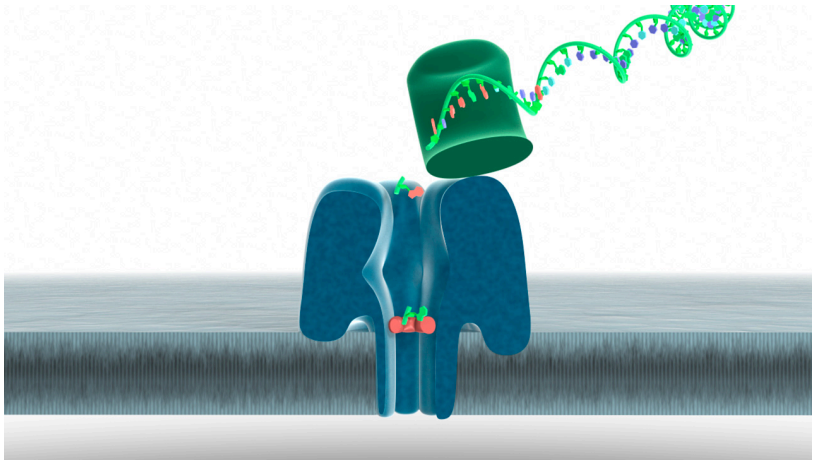


Figure 2.13: Oxford Nanopore Technologies: exonuclease sequencing. (Figure taken from [15].)

single stranded hairpin adapters are ligated to both ends of the double stranded DNA, producing so called SMRTbell templates (see Figure 2.11) [152]. The hairpin adapters provide a site for primer binding where sequencing commences. Depending on the length of the insert the enzyme can potentially go around the hairpin and the template can be sequenced multiple times. As the polymerase is directly monitored any template size can be sequenced.

CSS can be used for error correction as errors seem to occur randomly. At least three passes are required to build a consensus sequence and with five or more passes error rates can be reduced to $\leq 1\%$ (see Figure 2.12 for more details). However, reads that are shorter than 3x the size of the template that is sequenced will be discarded. Therefore, read length and throughput need to be balanced.

Another option for error correction is the combination of PacBio reads and Illumina reads. The short Illumina reads can be mapped onto the long PacBio reads and used for error correction. This combines the benefit of the lower error rates with longer read lengths.[86][154]

Oxford Nanopore Technologies

Another approach for single-molecule sequencing involves the utilisation of nanopores [136][15]. Nanopores are small holes with a diameter of approximately one nanometer. These can be either transmembrane cellular proteins or artificial holes in a silicon layer. The nanopores are immersed in conducting fluid with an electric current passing through them which is very sensitive to the size and shape of the pore. In Figure 2.13 the blue coloured protein represents the nanopore. A processive enzyme (shown in green) cleaves the single stranded DNA and ensures that only one nucleotide at a time passes through the protein nanopore. Each of the four nucleotides causes a characteristic change in the magnitude of the current and thus the DNA sequence can be inferred. Oxford

Figure 2.14: Oxford Nanopore's MinION: a library is loaded onto the flowcell inside the MinION. (Taken from [16].)

Nanopore Technologies distinguishes three kinds of DNA processing. The processing type described above is called *exonuclease sequencing*. *Strand sequencing* does not cleave the DNA into single bases but passes the whole strand through the nanopore whereas *solid state sequencing* uses a synthetic nanopore instead of a protein [15].

Nanopore sequencing has the potential to offer real time sequencing without deterioration of accuracy. The addition of a hairpin structure to the end of the double stranded DNA enables the additional uninterrupted sequencing of the complementary strand and therefore sequencing information from the sense and antisense strand can be obtained in one step. A sequencing chip can contain multiple arrays of nanopores facilitating parallelisation of the sequencing process. Also, the DNA molecule should not be damaged during the sequencing process which would allow re-sequencing of the molecule.

The company is currently developing two platforms: the GridION and the MinION. However, neither platform is commercially available at the moment. The first data sets from the GridION were presented by Oxford Nanopore Technologies at the Advances in Genome Biology and Technology (AGBT) meeting in February 2012. They presented error rates rates of 4% and forecasted improvements to below 1%, sequencing hundreds of kb per second and read lengths of 100 kb [71]. The proposed timeline was not kept by the company and another two years passed until further news. In addition to the larger GridION sequencing system the company is developing the MinION, a portable device that can be directly plugged into a computer (Figure 2.14). The MinION was announced to retail for less than US$900 in 2012 [71]. The device contains a single flow cell containing a sensor chip which in turn contains an array of wells. Each well is an independent electronic channel. A membrane comprising the nanopores lies across the wells to process the single molecules. [16]

At the start of 2014 the MinION access programme was launched by Oxford Nanopore

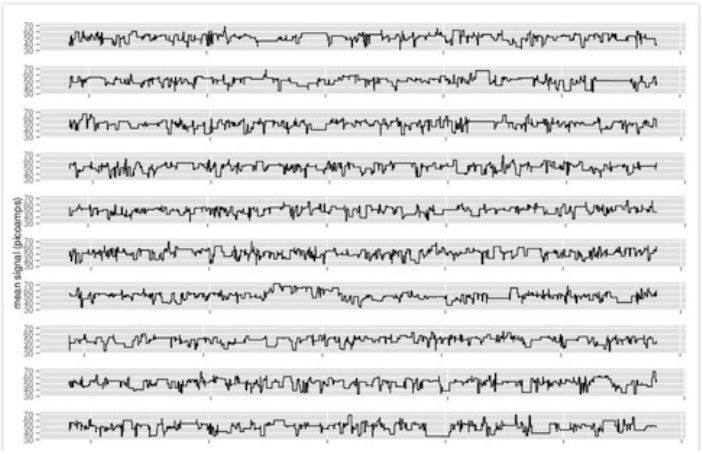


Figure 2.15: MinION measurement output: the change in current (picoamps) while sequencing *P. aeruginosa*. (Taken from [26].)

Technologies. Within the scope of this programme several hundred users are currently testing the new MinION system. Figure 2.15 shows the change in current recorded during one of the first sequencing runs on the MinION. At the AGBT in February 2014 it was reported that albeit high error rates more than 80% of the reads had perfect 50-mer sections highlighting the potential of this new technology. The commercial launch of the technology was announced for mid 2014.

Genia

In mid 2014 the company Genia Technologies, Inc. was purchased by Roche who also owns 454 Life Sciences. The technology is currently still under development. This method is also based on nanopore sequencing but instead of measuring the DNA as it traverses through the nanopore, this technique employs a DNA polymerase. The nucleotides are equipped with four different sized tags. These tags are released whenever a nucleotide gets incorporated. Each tag is then measured as it travels through the nanopore, generating the sequencing information. The commercial release is currently planned for the end of 2014.

2.3 Connection between Viral Haplotype Reconstruction and Metagenomics

I studied algorithms for viral haplotype reconstruction as well as bacterial metagenomics. In the following I will describe how these problems are closely related and the challenges that are encountered.

Pathogenic and non-pathogenic bacteria can belong to the same family and can have very similar DNA sequences. For example *Shigella* is a pathogen that can cause serious and

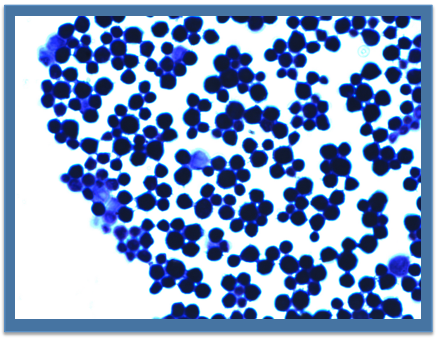


Figure 2.16: Picture of a non-pathogenic strain of *S. aureus* under an inverted fluorescence microscope.

widespread diseases. It is closely related to *Escherichia coli*, though most *E. coli* strains are harmless and even part of a healthy gut flora. Both, *Shigella* and *E. coli*, belong to the family of *Enterobacteriaceae*. In order to choose the right treatment for a disease it is necessary to be able to classify closely related bacteria correctly. Having very similar genomes and in particular almost identical 16S rRNA genes we need to be able to distinguish variants that differ by only a few base pairs. Fukushima et al. [63] report that the 16S rRNA gene of the two *Shigella* strains *S. sonnei* and *S. flexneri* differ by only 0.1%. Those two bacteria are the major cause for shigellosis, which is a severe and widespread illness with up to 150 million cases reported worldwide each year [46, p.1040]. Their similarity to the 16S rRNA gene of *E. coli* is 99.9% and 99.8%, respectively - but only 99.7% to *S. boydii* which is another *Shigella* strain.

Targeting the ribosomal gene to differentiate on the strain level has proven to be challenging. The 16S rRNA gene is highly conserved and relatively easy to sequence. It has been utilised to analyse the phylogenetic relationships of different organisms and has provided great insight into the diversity and structure of microbial communities. The 16S rRNA gene evolves much slower than any of the protein-coding genes. This is advantageous for distinguishing different organisms but a problem for differentiating on the strain level. Different strains of an organism can have almost identical 16S rRNA genes, but differ by a few SNPs on a protein coding gene, turning a harmless bacterium into a pathogenic one.

Another example that highlights the importance of resolving fine-scale variation and differentiating on the strain level is *Staphylococcus aureus*, which is a bacterium that is commonly found on the human skin and is often part of a normal skin flora without causing symptoms (see Figure 2.16). At the same time *S. aureus* can cause a wide range of diseases, ranging from minor skin diseases to very serious illnesses like pneumonia and meningitis. It causes the majority of bacterial skin infections in humans [87]. McCaig et al. [108] report that in the USA 11.6 million ambulatory care visits are due to *S. aureus* infections. More and more strains of *S. aureus* are becoming multi-drug-resistant [80]. It has been shown that the closely related bacterium *Staphylococcus epidermidis* destroys and inhibits biofilm formation and nasal colonisation by *S. aureus* [80]. The ribosomal genes of *S. epidermis* (GenBank D83363.1) and *S. aureus* (GenBank D83357.1) differ by only 21bp over a total gene length of 1,476 bp. Studies have shown that in general the 16S rRNA gene is not suitable to distinguish different *Staphylococcus* species [66].

Currently available algorithms have difficulties in distinguishing different variants of closely related organisms and can thus underestimate the diversity and misclassify bacteria. At the same time sequencing errors can artificially inflate the diversity in the sample. It is often important to identify strains from the same species, e.g. if we are dealing with a novel pathogenic strain. The recent *E. coli* outbreak in Germany in 2011 is just one example where a new pathogenic *E. coli* strain needed to be analysed quickly in order to find an effective treatment and to locate the origin of the outbreak. [131]

These problems are closely related to the study of viral haplotype reconstruction where one can encounter even lower diversity. In March 2014, an outbreak caused by a novel strain of *Zaire ebolavirus* (EBOV) in Guinea was reported to the World Health Organisation (WHO), which has caused 670 fatal cases to date (as of July 31, 2014 [132]). The virus presents a major public health issue in sub-Saharan countries with a high fatality rate between 30-90%, depending on the virus species [34]. Sequencing analysis has assisted in the characterisation of the new strain and in locating the origin of the outbreak. In addition, the statistics involved with the assembly will be of interest to both areas - in particular the question of how much coverage and which read length is necessary to unambiguously determine the haplotypes. In the context of microbial communities this would refer to the number of species present in the community.

2.4 Bioinformatics: Realising the Promise and Potential of DNA Sequencing

Bioinformatics is an integral part of any sequencing project and is crucial in order to turn big data sets into meaningful information and insight. The development of appropriate bioinformatic tools is essential in realising the potential and for accessing the information contained in sequencing data.

Unprecedented advances have been made in the speed and throughput of NGS technologies over recent years. Moore's law is often used to evaluate how fast a technology is moving forward. It describes the observation that transistor density is growing exponentially and doubles every two years. Any technology that shows a similar trend is considered to perform exceedingly well. Figure 2.17 shows that sequencing has outcompeted Moore's law with the introduction of NGS. With the help of Sanger sequencing the first human genome was sequenced for billions of dollars and marked a milestone in 2001. A major goal since then was the $1,000 genome which has been accomplished in 2014 with the HiSeq X, the newest generation of Illumina sequencing platforms.

However, these costs do not include the development and improvement of sequencing pipelines and bioinformatic tools for the downstream analysis. The cost of sequencing

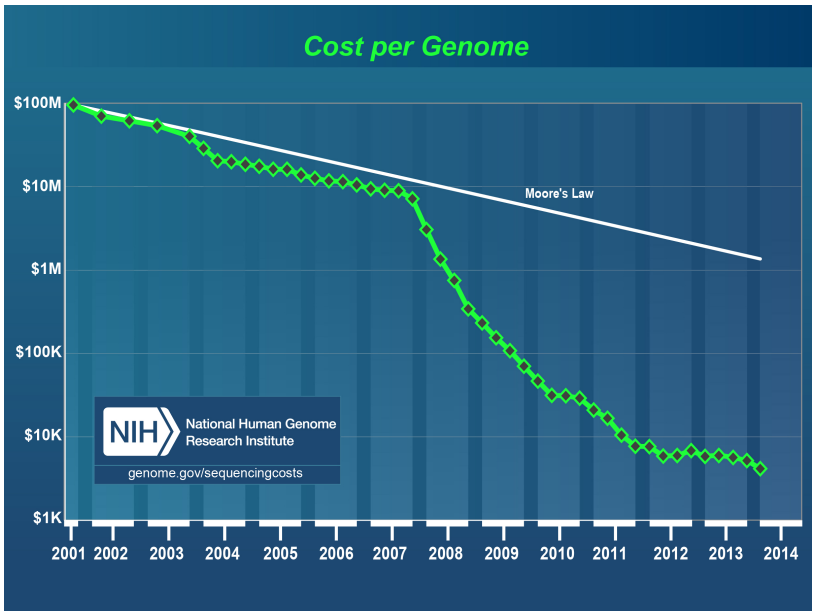


Figure 2.17: Development of sequencing costs since 2001 in comparison to Moore's law. (Note: the y-axis is log scaled.) (Taken from [19].)

has dramatically decreased and platforms continuously improve throughput. In addition, the advent of benchtop sequencers has allowed even a small laboratory to acquire their own sequencer and sequencing platforms can now be found all over the world (see Figure 2.18). As a result, the amount of data that needs to be analysed has grown exponentially and generates increasingly high demands on bioinformatic tools for the data analysis. Sequencing more and more samples is meaningless if we are not able to reveal and correctly interpret the information concealed in the data.

The cost associated with data management and computational equipment should not be neglected. These costs will drastically increase in the near future as the throughput of next generation sequencing platforms continues to increase faster than the growth of transistor density and sequencing costs decrease enabling a higher number as well as larger projects. Data analysis needs to address sequencing errors, sequencing assembly and alignment, the identification of variants and the interpretation of results. However, sequencing data can only be turned into meaningful information if methods are available that can deal with the idiosyncrasies of the different platforms and the size of the ever-growing data sets. Complex data sets require sophisticated and specific tools for the analysis. The development of analysis tools lags behind the technological advances and will be the next challenge in order to unlock the true potential of DNA sequencing.

Past experiences have highlighted the need for platform specific analysis tools. When 454 Life Sciences first introduced their NGS sequencing platform the sample diversity was greatly overestimated as sequencing errors were mistaken for true variation [129][77]. Pyrosequencing is a very powerful tool that has advanced research in many areas and

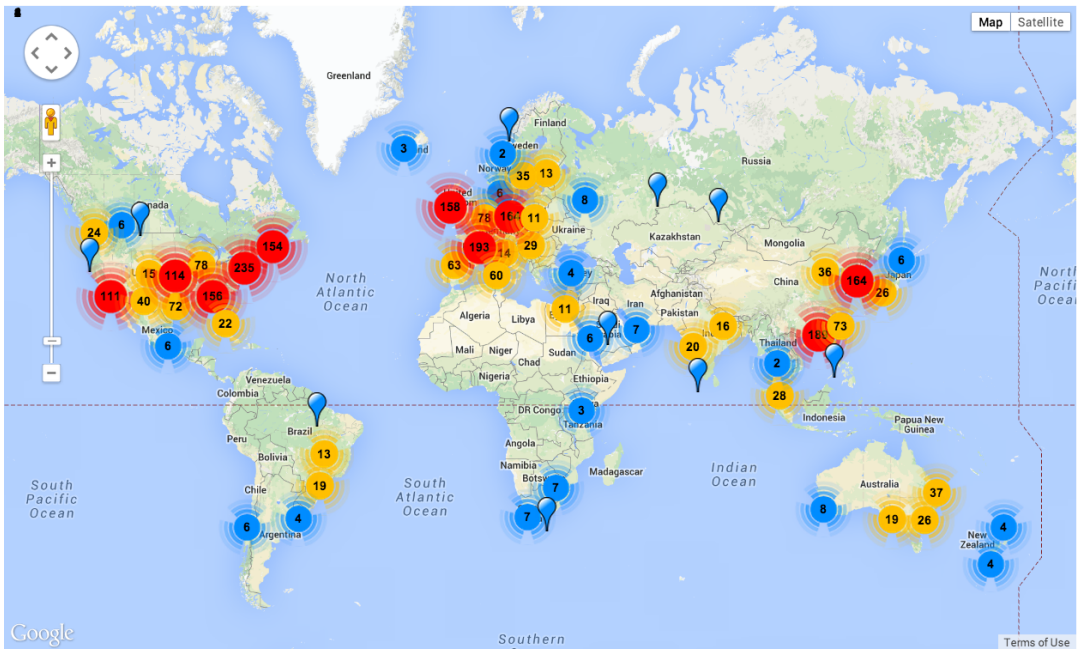


Figure 2.18: Overview of sequencing platforms worldwide (including 454, GAII, HiSeq, MiSeq, Ion Torrent, PacBio, Proton, SOLiD and Polonator). (Taken from [9].)

provides new insight - if the data is handled correctly.

454 Life Sciences dominated the sequencing market for years, however, with increased read lengths, higher throughput and lower costs Illumina has become the sequencing platform of choice for many researchers. The two technologies differ fundamentally and computational tools need to be designed to handle the peculiarities of the respective technology. So far, Illumina profiles are poorly understood and this issue needs to be addressed (see Chapter 6 and 7). The great number of NGS sequencers worldwide (see Figure 2.18) and the fact that many third generation sequencing platforms are still in the experimental development phase, further highlights the need to advance methods and tools for NGS technologies.

Next generation sequencing poses a great opportunity but a lot of work is still needed to reveal its true capability. Accurate and efficient bioinformatic tools are the key to realise these capabilities. Sequencing has the potential to continue to transform our way of living in many areas as we will outline in the next section.

2.5 The Impact of NGS and its Applications

New Insight into the Microbial World with Metagenomics

So far we have only scratched the surface of the microbial world but the importance of microbes cannot be underestimated. Microorganisms can be found almost everywhere and are essential for life on earth. They have adapted to different environments that are

extremely diverse - ranging from volcanoes and acid mine drainage to the deep ocean. They preserve the atmosphere by generating oxygen from carbon dioxide and are responsible for about half of the photosynthesis on earth [28]. The number of microbes on and in our body is ten times more than the number of human cells and they are essential for our well-being. For example, they synthesise important vitamins and amino acids and shape our immune system. Soil microbes are vital for plant growth and microbial communities help to keep our groundwater clean. These are just a few examples of how important microbial communities are for life on earth. Yet our knowledge of microbial communities is very poor. Large numbers of previously unknown organisms were discovered and metagenomics allows us to study the enormous functional gene diversity in the microbial world. This provides a much broader characterisation of the different organisms and their capacities than phylogenetic surveys based on a single gene such as the 16S rRNA gene. But we still need to develop techniques to deal with these large amounts of data produced by shotgun metagenomics. Problems that we need to address include how to separate reads from different organisms, how to compare microbial communities and how to infer the population structure from the reads, especially in regards to low-abundance species.

The majority of microbially mediated processes are based on a complicated network of interactions and gene regulation within complex communities, which makes gaining a deep understanding of microbial communities very important [28]. Connectivity and interdependence of the community members are major factors that shape the structure of a microbial community. Many microbes live in a symbiotic relationship with other organisms affecting their diversity and abundance levels in the community. At the same time the interdependencies in a community depend on the microbes that are present. Also, since a population of cells passes through different growth cycles - including the lag phase, exponential phase, stationary phase, and death phase - time and space have an impact on the community structure and the symbiotic relationships of the microbes. At the same time changes over time and space depend on the initial community structure as well as their interactions. In addition, external factors such as the availability of nutrients and environmental conditions shape the community structure. Traditional methods like culturing do not capture information about these complex networks and the community structure. This outlines the importance of studying microbes and microbial communities in the context of their environment and puts further emphasis on the importance of metagenomics.

Applying metagenomics to understand microbial communities also has many applications in industry ranging from wastewater treatment to energy production and sanitation systems for developing countries as well as agriculture. In the health sector metagenomics allows us to study the influence of the microbiome on health and can aid in drug

and vaccine development as well as personal care products. New fields of research have evolved including biotechnology, bioengineering and biomedical engineering. Even areas like crime scene analysis were transformed with the advent of forensic biology.

Personal Genomics

NGS has the potential to identify rare diseases and characterise their role in common diseases. It allows us to gain further understanding of the cancer genome, revealing previously unknown genetic variation. It also offers new possibilities to target viral diseases such as HIV and hepatitis C by analysing viral quasi-species (see Chapter 5). The $1,000 genome was a big step on the way to personal genomics and opens up new vistas for the development of more effective medicines with less side effects.

3 A Read Simulation Program for Microbial and Viral Genomics

3.1 Abstract

More realistic *in silico* data sets are essential for the rigorous assessment of programs and tools and can assist in the optimal design of sequencing experiments. We developed a read simulator for amplicon and metagenomic data sets that is capable of reflecting the motif-based nature of errors encountered in Illumina data. Experimental factors such as library preparation and choice of primers, in the case of amplicon data sets, have a fundamental impact on the error patterns encountered in Illumina sequencing data [140]. They determine the main motifs associated with a large fraction of errors and thus constitute a critical bias (see Chapters 6 and 7). Our program *MicroSim* is the first program that can explicitly simulate the impact of various experimental factors. With a motif-based approach we can simulate peculiarities specific for Illumina platforms which have become some of the most utilised platforms worldwide. Additionally, we provide a large range of up-to-date Illumina error patterns with the program, enabling the quick and efficient simulation of *in silico* data sets without the need for prior computations and existing sequencing data sets. MicroSim offers the flexibility necessary for the simulation of complex community structures that are encountered in microbial and viral samples. The user can input multiple genomes and specify the corresponding abundance distribution. The desired number of reads is passed to the program as an input parameter and the user can choose between paired-end and single-end reads which are uniformly distributed across the genome of origin. Several empirical insert size distributions are available for the simulation of metagenomic paired-end data sets. In addition, 454 noise can be simulated with or without PCR errors as well as noise free reads. The program is implemented in C and can be run across multiple CPUs which allows the efficient simulation of large data sets reflecting the capabilities of current high-throughput sequencers. MicroSim is a platform independent software and available under the FreeBSD license.

This chapter is partly based on the publication:

> Melanie Schirmer, William T Sloan, and Christopher Quince. **MicroSim: A motif-based next-generation read simulator.** (In preparation)

The program was used to simulate read data sets for the following publications:

> Melanie Schirmer, William T Sloan, and Christopher Quince. **Benchmarking of viral haplotype reconstruction programs: An overview of the ca-**

pacities and limitations of currently available programs. (Briefings in Bioinformatics, page bbs081, 2012.)

J. Alneberg, B. S. Bjarnason, I. de Bruijn, M. Schirmer, J. Quick, U. Z. Ijaz, L. Lahti, N. J. Loman, A. F. Andersson and C. Quince, **Binning metagenomic contigs by coverage and composition.** (Nature Methods, 2014)

I. Gregor, J. Dröge, M. Schirmer, C. Quince and A. C. McHardy, **PhyloPythiaS+: A self-training method for the rapid reconstruction of low-ranking taxonomic bins from metagenomes.** (In preparation)

Original Contributions

I designed and implemented the algorithm for the read simulation. In addition, I generated a large range of Illumina error profiles for the current version, identified the library and primer specific motifs and inferred several insert size distributions for paired-end simulations. The flowgram distributions for the 454 simulation and the PCR transition matrix were taken from Quince et al. [130]. For the initial version the 75bp Illumina error distribution was taken from Huang et al. [75].

3.2 Introduction

The rise of next generation sequencing (NGS) platforms has transformed research in genomics and formed the basis for many genetic discoveries. Accurate and reliable bioinformatic programs play a vital role in the analysis of NGS data sets. For the development of these programs, comprehensive *in silico* data sets are indispensable as they are essential for the validation and benchmarking of these analysis tools. *In silico* data sets can also assist in finding the optimal design strategy for NGS experiments as they allow researchers to test the effects of various factors such as level of coverage, read length and sequencing technology before conducting the experiment. This can save money and time as well as enhance the value of an experiment. However, in order to achieve these objectives, the *in silico* data sets must closely reflect the properties of real sequencing data. Therefore, we need a good representation of biases and systematic errors associated with the different sequencing technologies and experimental parameters.

Our approach incorporates new findings based on an extensive study of error patterns that revealed the impact of library preparation method and primers on the formation of errors in Illumina data sets. We calculated realistic error profiles that effectively mimic these biases. Simulation tools also need to be able to reflect the high throughput

capabilities of the currently available platforms and the complexity of the samples. To simulate the complex population structure encountered in microbial and viral genomics, a user needs to be able to specify detailed abundance profiles for the input genomes. Many of the earlier programs such as ART [75], ArtificialFastqGenerator [62], pIRS [74], wgsim [10] and Mason [73] can only take one input genome which drastically limits the complexity of the community that can be simulated. Others can only simulate 454 reads (e.g. flowsim [37]) or use simplistic Illumina error profiles, thus limiting their scope of application (e.g. MetaSim [134]). NeSSM [82] and Grinder [30] are more flexible programs but do not reflect the motif-based nature of Illumina errors. The only other program that uses a motif-based approach is GemSIM [109]. However, the only error models that are supplied with the program are based on the Genome Analyzer and outdated by current standards. Error models based on the current Illumina platforms can be created by the user but require the availability of suitable sequencing data sets. If these data sets are not based on known organisms, an additional, sequencing-unrelated bias is introduced by the limitations of the databases used to align these reads. Also, GemSIM is only designed for the simulation of metagenomic data sets. Amplicon data sets, however, are still important for the in-depth study of sample diversity. We generated ready-to-use and up-to-date, library and primer specific error models for the simulation of *in silico* reads, based on a large range of mock sequencing experiments.

When we developed the initial version of our program, all programs lacked the flexibility to simulate the complexity of viral and microbial communities. This initial version was part of the work presented in Chapter 5 [141]. For this implementation we adapted a position and nucleotide specific error profile for 75bp Illumina reads from the simulation program ART [75]. In contrast to ART our program allows multiple input genomes and the user can specify an abundance distribution to reflect complex community structures to simulate amplicon as well as metagenomic data sets. In addition to Illumina reads, our program can also simulate 454 reads, noise-free reads and PCR errors for amplicon data sets. For the subsequent version we updated the Illumina error profiles based on our research on Illumina error patterns (see Chapter 6). New insight into the error patterns and biases encountered in Illumina sequencing lead to several conceptual updates with regards to the Illumina read simulation. Position and nucleotide specific profiles for several library preparation methods were incorporated in this version. Due to the spikes observed in the amplicon data sets, these error profiles implicitly mimic the bias associated with motifs. In addition, we directly implemented a motif-based approach where the likelihood of an error occurring is dictated by the 3-mer preceding the base.

3.3 Methods and Algorithms

All required input parameters are passed to the program in a text file. The parameter file can be conveniently stored with the simulated data for future reference. The input comprises the following information:

1. Input reference sequences (genomes or amplicons) in fasta format.
2. Abundance distribution: either a uniform abundance distribution can be selected or the user can specify a file containing the frequencies for each organisms provided in (1).
3. Desired number of reads.
4. Either Illumina, 454 or noise free reads can be simulated:
    - For noise free reads, the mean read length and standard deviation can be specified.
    - For 454 reads the number of cycles and the flow order of the nucleotides can be specified, optionally PCR noise can be added to the reads. Note, that Illumina amplicon profiles automatically include PCR noise.
    - A choice of Illumina error profiles (details in next section) is available and single or paired-end reads can be simulated.
5. Amplicon and metagenomic sequencing data can be simulated.
6. An insert size distribution can be chosen for metagenomic simulations.
7. The seed for the random number generation can be specified. Multiple sequencing runs for the same sample with the same experimental parameters can be simulated by varying the seed.
8. Number of CPUs/threads for parallel computations.
9. The prefix for the output files can be specified.

The program starts by reading in the reference sequences from the specified file. Ambiguous nucleotides are disregarded for the simulation. The program then iterates over the total number of reads. In each iteration the origin of the read is chosen at random with probability proportional to the specified frequency distribution of the reference sequences. Alternatively, the user can choose to use a uniform abundance distribution. For the random number generation we used the GSL-GNU scientific library [64]. For metagenomic sequencing the reference sequences are randomly assigned to represent the plus or minus strand of the organism.

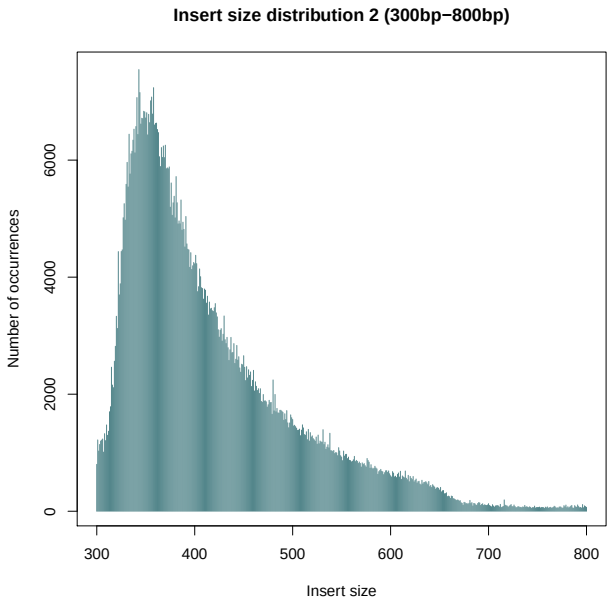


Figure 3.1: Insert size distribution for the simulation of fragments between 300 and 800bp (average fragment length: 416bp). The graph demonstrates the preference towards smaller fragments during the sequencing.

The simulation starts by determining the fragment size in the case of a paired-end whole genome shotgun sequencing (WGS) simulation by sampling a value from the insert size distribution. There are three distributions supplied with the program based on empirical data sets. The first distribution covers a range from 300 to 600bp with an average fragment size of 391bp, the second distribution ranges from 300-800bp (average 416bp, see Figure 3.1) and the third distribution covers a larger range of fragments with sizes ranging from 250 to 1,000bp (average 532bp).

The starting position of the read is drawn from a uniform distribution and must be smaller than the genome length minus the fragment size. Next, the noise is added to the read. Details for the different noise distributions are provided below. PCR errors are simulated with the help of a transition matrix as specified in Table 3.2. After the completion of the R1 read, the R2 read is analogously simulated from the complementary strand. Figure 3.2 displays a paired-end read and highlights the origin of the R1 and R2 read, respectively. All R1 reads originate from the plus strand of the sequenced fragment. The R2 reads start at the 5' end of the complementary minus strand.

For noise-free reads the read length is sampled from a normal distribution specified by the user. The read length for 454 reads can be indirectly controlled by adjusting the number of cycles. The nucleotides are flowed sequentially over the plate. During 454 sequencing, the default order is T→A→C→G but can be controlled by the user. Each nucleotide that is flowed over the plate corresponds to one cycle.

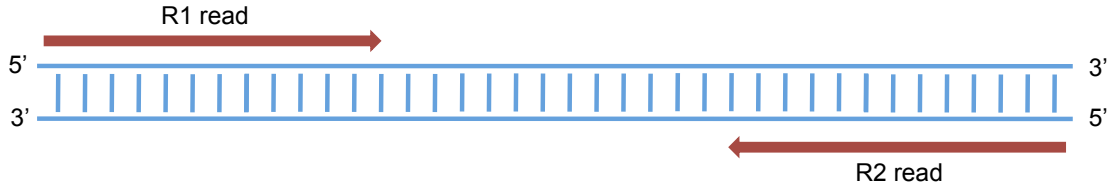


Figure 3.2: Paired-end read schematic: the R1 read originates from the plus strand of the fragment and the R2 read originates from the complementary minus strand.

The program can be run in parallel to facilitate the efficient simulation of large data sets. We parallelised the program using OpenMP. The required number of reads is equally distributed between the number of threads specified in the parameter file. Each thread uses its own random number generator as the GSL library is not thread safe. The seed provided in the parameter file is used to draw random numbers for the initialisation of the different threads. The read simulated by an individual thread is output at the end of each iteration. As the output order can differ for R1 and R2 reads, the reads need to be sorted at the end of the simulation for parallel computations. A script for sorting the reads is supplied with the program. Each read contains details on the genome from which it originates, strand information and the read number. The simulation of one million 250bp Illumina reads from 20 bacterial genomes takes on average 8 minutes and 40 seconds on a single core (CentOS 6. x86_64, AMD Opteron 6174 @ 2.2Ghz, 256GB DDR3 1067Mhz RAM). Increasing the number of cores to 10, reduces this time to 59 seconds and on 20 cores one million reads can on average be simulated in 38s (see Figure 3.3).

Different Error Profiles

The next section describes the different noise profiles for Illumina and 454 reads. Errors in 454 sequencing were already well characterised prior to the development of my simu-

Table 3.1: Overview of the different error profiles for the read simulation program and the respective studies in which they were used to create *in silico* data sets.

Error Profile	Chapter & Publications (P)
454 error profile	Chapter 5 Benchmarking of viral haplotype reconstruction programs (P)
75 bp Illumina *	Chapter 5 Benchmarking of viral haplotype reconstruction programs (P)
position & nucleotide specific Illumina error profile	Binning metagenomic contigs by coverage and composition (P) PhyloPythiaS+ (P)
motif-based Illumina error profile	

* No longer supported in the current version of the program.

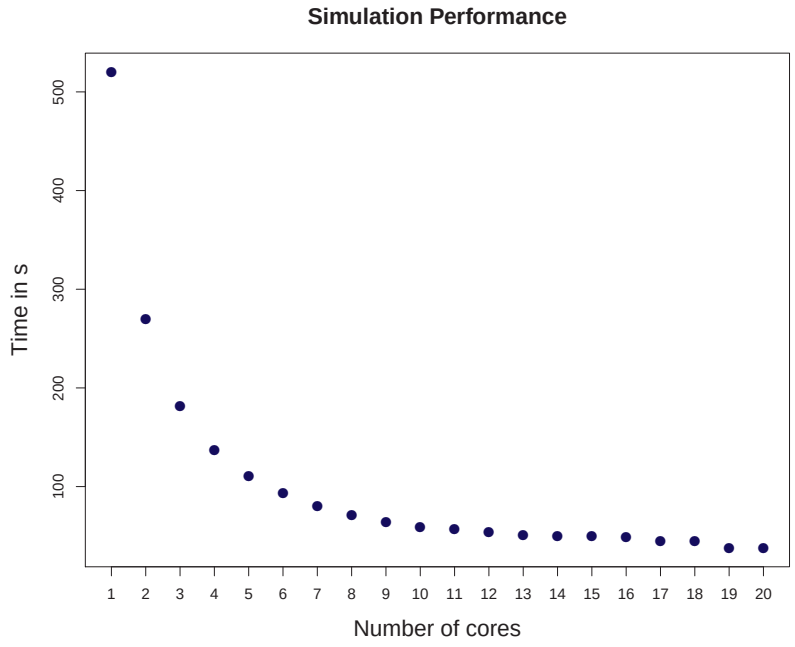


Figure 3.3: MicroSim: average time required to simulate one million reads based on 20 genomes in relation to the number of cores used for the computations. Averages are based on 50 simulations. On 20 cores the simulation took on average 38 seconds.

lation program. Therefore no updates for the 454 error profile were required. Ongoing research on sequencing errors in Illumina data throughout my Ph.D. project resulted in two conceptual updates for the Illumina error profiles. Note, that the initial 75bp Illumina error profile is no longer supported in the current version of the simulation program. Table 3.1 gives an overview of the different profiles and the studies in which they were used.

Simulating 454 noise

The nucleotide sequence of the read is converted to flowgram values before the noise is added. For example, for the default nucleotide flow order (T→A→C→G) the sequence "AGGTTTG" would be converted to "01023001" and represents the result of eight cycles. For homopolymer lengths ranging from 0 to 9, we utilise different empirical probability

Table 3.2: Transition matrix for the simulation of PCR errors (taken from [130]). Here, X represents all nucleotides besides the one that is currently considered.

To: / From:	A	C	G	T
A	1- $\sum_X P(A \to X)$	exp(-11.619259)	exp(-7.748623)	exp(-11.694004)
C	exp(-11.619259)	1- $\sum_X P(C \to X)$	exp(-12.852562)	exp(-7.619657)
G	exp(-7.748623)	exp(-12.852562)	1- $\sum_X P(G \to X)$	exp(-10.964048)
T	exp(-11.694004)	exp(-7.619657)	exp(-10.964048)	1- $\sum_X P(T \to X)$

distributions. Each distribution specifies how likely we are to observe flowgram values between 0.00-0.01, 0.01-0.02 and so on up to 99.99-100.00 for the respective homopolymer length. So by drawing a random number between zero and one we can simulate the observed flow value for the encountered homopolymer. The program outputs the reads in ssf format, containing the simulated flowgram values, as well as the corresponding fasta file, for which the flow values were rounded to the closest integer.

Simulating 75bp Illumina reads

The following noise distribution was used to create the test data sets presented in Chapter 5. The noise distribution is intended for the simulation of substitution errors in 75bp paired-end reads and was adapted from the simulation program ART [75]. As ART does not support multiple input genomes and complex community structure, we were not able to use the program directly.

For each position on the read the noise distribution specifies how often a certain quality score was observed. The quality score is designed to reflect the probability that a base is called incorrectly. For each position the quality score of the simulated read is sampled from the respective distribution. Based on the quality score, a random number between 0 and 1 then determines if a substitution occurs and the substituting nucleotide is chosen at random. For insertions a fixed probability of 0.00009 is used across the whole read and the inserted nucleotide is chosen at random. The probability for a deletion is 0.00011 and constant across all positions. The reads are outputted in fastq format.

This approach is limited by the assumption that the quality scores truly reflect the correctness of the respective base. As later studies have shown (see Chapter 6), in particular for amplicon sequencing quality scores do not accurately reflect the probability of an error. Also, Illumina read lengths have greatly increased over the past years requiring more up-to-date distributions with longer read lengths.

Simulating Illumina reads with position and nucleotide specific error profiles

We enhanced the error simulation by computing separate error distributions for all nucleotides. In addition to nucleotide and position specific substitution models we also model insertions and deletions according to a distribution and the simulations no longer rely on the quality scores. These extensions allow us to reflect that some nucleotides are more error prone than others. Furthermore, the substituting nucleotide in turn is drawn from a position-specific distribution as we also recorded a preference in connection with the substituting nucleotide. For insertions, nucleotide specific distributions allow to determine the inserted nucleotide.

These distributions differ for different Illumina platforms and are also specific for the employed library preparation method. We provide pre-computed distributions for the MiSeq and HiSeq for amplicon as well as metagenome sequencing. The amplicon distributions simultaneously model the effect of PCR errors and the impact of different primers on the error patterns. The pre-computed profiles include the following library preparation methods: Fusion Golay, NexteraXT and Dual Index and the distributions are designed to simulate 2x250bp.

Simulating Illumina reads with motif-based error profiles

Here, we directly simulate the impact of the motifs on the error formation. So far we only indirectly modelled this effect in the case of the position specific amplicon distribution through the accumulation of errors (spikes) at certain positions along the read.

The probability of an error occurring (substitution, insertion or deletion) depends on the nucleotide itself as well as on the motif preceding the current base. Therefore, we computed separate distributions over all 64 possible 3-mers for each type of nucleotide. These distributions specify the error probability at the current position based on the type of nucleotide and the associated motif. In the case of substitutions the substituting nucleotide is picked at random. Distinctive distributions are used for R1 and R2 reads, respectively. The reads are outputted in fasta format. In addition the program returns the total number of errors which can be used to estimate the overall error rate. The pre-computed error profiles include amplicon and metagenomic data, MiSeq and HiSeq, and NexteraXT, Dual Index and Fusion Golay.

3.4 Conclusion and Future Work

We introduced a program that is able to efficiently simulate realistic Illumina data sets by utilising a motif-based approach. We also provide a large range of default distributions based on mock community data sets, that offer researchers a huge degree of flexibility for the simulation without the need for prior computations based on real sequencing data. Also, the program is designed such that new error distributions can be easily added as new library preparation methods and new technologies become available.

Other existing simulation programs do not offer the same degree of flexibility or require the user to create error distributions prior to the simulation. Furthermore, restrictions of one input genome limit the possible complexity of the simulation. Theoretically multiple genomes could be concatenated for these simulation programs. However, this process is cumbersome and would substantially limit the realisable complexity and accuracy of the population structure.

In the future we will further extend my program to incorporate sequencing coverage bias as well as sample coverage bias for multiplexing samples. The genome coverage bias is due to the different number of hydrogen bonds between complementary nucleotides. There are two hydrogen bonds for the complementary bases adenine (A) and thymine (T) and three hydrogen bonds between guanine (G) and cytosine (C). Thus it is harder to break GC bonds. For instance the melting temperature of DNA depends highly on the GC content of the specific genome. A relation of GC-content and the under-representation of species has been observed as well as a connection of GC-rich parts within a genome and coverage variation within a genome.

4 Modelling an *In Silico* Viral Haplotype Population Based on an Experimental Data Set

4.1 Abstract

Simulated data sets are essential to evaluate the performance of programs and algorithms. We need *in silico* data sets that closely reflect the structure and properties of experimental data sets and mimic realistic conditions. The currently available test data sets show a lack of complexity and do not reflect the evolutionary structure of viral haplotype populations. Here, we devised and implemented a new algorithm that simulates a possible set of haplotypes based on information from an experimental next generation sequencing (NGS) data set of a viral sample. The single nucleotide polymorphisms (SNPs) of the resulting simulated population and their frequencies are consistent with the SNPs observed in the experimental data. The program simulates the evolution of the haplotypes from a single wildtype and the viral population mimicks the diversity of the *in vivo* population from which the sample was taken. The algorithm is implemented in C permitting time efficient simulations. We applied our algorithm to a foot-and-mouth virus (FMV) data set that was sequenced on the Genome Analyzer and present the simulation results in this chapter. The FMV *in silico* haplotype population constituted one of the test data sets for our benchmarking study on viral haplotype reconstruction programs (see Chapter 5) and is part of the associated publication [141].

Original Contribution

I designed and implemented the algorithm for creating complex *in silico* populations based on NGS sequencing data. To my knowledge this is the first algorithm that incorporates a position-specific variation model across the whole genome. The algorithm can be used in combination with any NGS read data set in order to simulate the evolution of a single haplotype into a complex quasi-species. The SNPs and their frequencies concur with the SNPs observed in the experimental data set. We applied our algorithm to a foot-and-mouth virus sequencing data set to produce a complex test data set for benchmarking studies. The NGS data set used for the simulation was provided by Dr Marco Morelli and Prof. Dan Haydon [160].

4.2 Introduction

The lack of proof checking during replication causes RNA viruses to have mutation rates about a million times larger than within human cells [118]. This results in a population of closely related genomes, a so-called *quasi-species*. The high mutation rate is dangerous

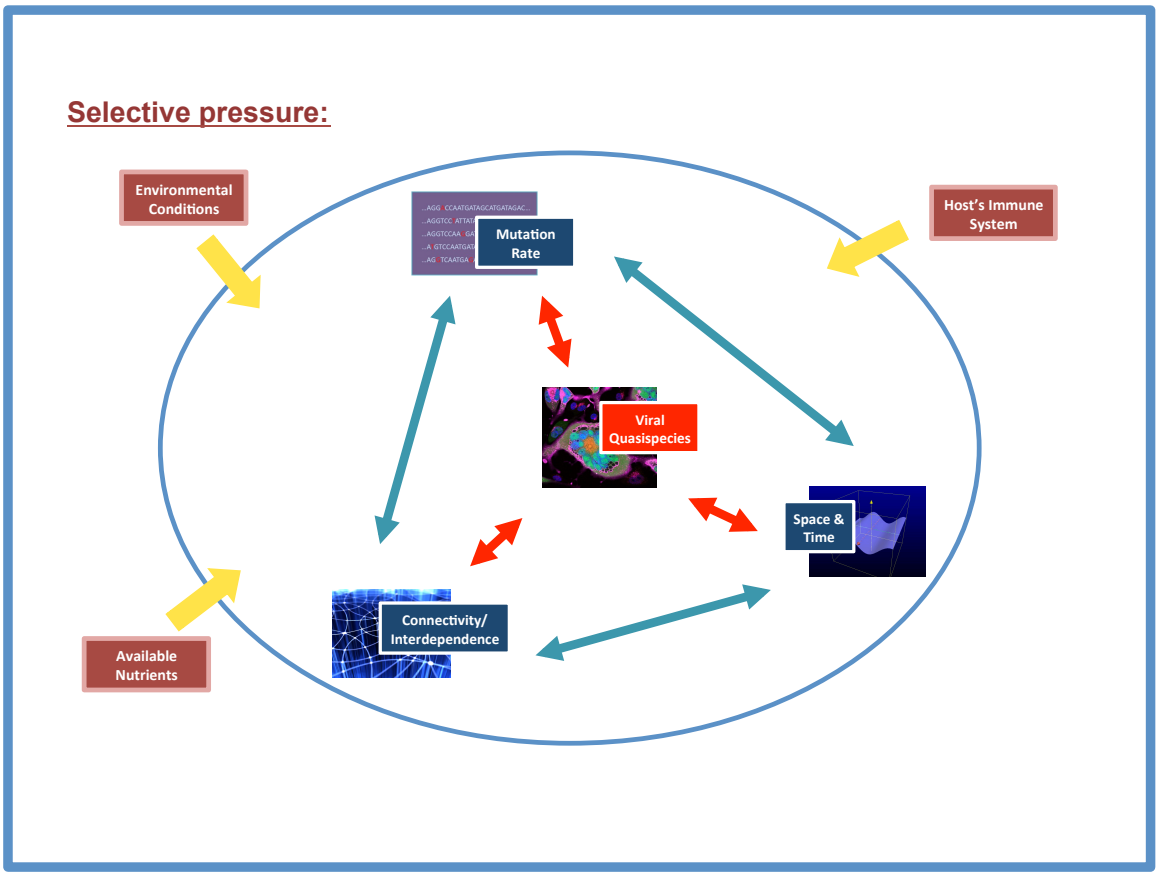


Figure 4.1: Complexity of a viral quasi-species: External and internal factors shape a viral quasi-species. Changing environmental conditions, the host's immune system and the availability of nutrients impact the population structure and constitute important external factors. Internal factors include the mutation rate as well as interdependencies between the viral haplotypes causing changes in the population structure over space and time.

for the virus, since it results in many non-viable clones, but it also provides the virus with a large number of potentially beneficial mutations allowing it to adapt quickly to changing environments during infections. It is likely that the haplotypes that enable the virus to survive selective pressure pre-exist in the population [42]. Thus it is important to determine all haplotypes in order to develop effective treatments and vaccines.

Viral evolution includes several bottlenecks [47]. After n cycles of replication selective pressure (e.g. the immune system or environmental changes) causes the extinction of the majority of haplotypes with high abundance levels. After this the population grows again for the next n cycles where evolution is mainly driven by the high mutation rates until (due to selective pressure) the majority of highly abundant haplotypes becomes extinct again. Every bottleneck causes major changes in the structure of the viral population and haplotypes with previously low abundance levels can become dominant. This further emphasises the need to infer the haplotypes with low frequencies when reconstructing a haplotype population from sequencing data.

Vignuzzi et al. work with the hypothesis that a viral quasi-species is more than a col-

lection of diverse mutants. They show that haplotypes interact and that all members of a population together contribute to the characteristics of the population [155]. They tested this hypothesis by limiting the genomic diversity in a population. For this they utilised a virus with an enhanced fidelity polymerase. Their study provides evidence of the complementary relationships between members of the quasi-species. In addition, their analysis indicates that selection takes place at the population level rather than on individual variants. They established a connection between mutation rate, population dynamics and pathogenesis. This supports the theory that the diversity in a viral population is essential for adaptation to new environments and to survive selective pressures such as the immune system. They found that the diversity of the viral population correlates with enhanced pathogenesis.

Figure 4.1 summarises the factors that shape a viral quasi-species. Selective pressure is a major factor influencing the structure of a viral population: Changing environmental conditions, the host's immune system (CRISPRs for bacteria [46, p.257]) and availability of nutrients are important outside factors that shape a viral community. At the same time the viral population changes over space and time due to the high mutation rate as well as the interactions between the haplotypes that are present in the community.

The development of next generation sequencing (NGS) technologies creates the opportunity to respond quickly to outbreaks and gain a better understanding of viral populations. Sequencing enabled us for the first time to analyse the community structure of a viral population and to detect low-abundant haplotypes. In order to assess the sensitivity and ability of programs to reconstruct complex population structures, we need *in silico* test data sets that reflect the complexity and challenges encountered in real viral samples. In the following we present an algorithm to simulate the evolution of a viral population from a single strain by incorporating the diversity revealed by a next generation sequencing data set.

4.3 Algorithm for Simulating a Haplotype Population

The input data can be inferred from any viral sample that was sequenced on any NGS platform. The following information needs to be extracted for every position of the genome and constitutes the input for my program:

- the reference nucleotide at every position
- the frequencies of the observed SNPs at the respective position

We will discuss an example of a suitable experimental data set in detail in Section 4.4.

The goal is to infer a set of *possible* haplotypes with SNPs that are consistent with the nucleotide frequencies of the experimental data. So the SNPs of the *in silico* haplotypes and their frequencies will be the same as in the experimental data set. For this we iteratively construct a probabilistic tree, starting with the wildtype (i.e. a chosen reference sequence for the simulation) and consider one SNP at a time until all polymorphisms are incorporated into the tree. The leaves of the resulting tree contain the haplotypes of the *in silico* population. Note, that the set of haplotypes produced by the algorithm depends on the seed for the random number generation. For different seed values the algorithm will generate different sets of haplotypes.

In order to simulate a population that closely reflects a real population, a high sequencing coverage in the experimental data set is advantageous as this is necessary to identify SNPs of low-frequency haplotypes. The experimental data set that we used to simulate the foot-and-mouth virus population had a high coverage of about 4,900 for each position in the genome.

Algorithm:

The following steps summarise the theoretical concept of the simulation.

- Starting point: We start with the reference sequence and assume that the infection was initiated by a single viral strain. Thus the root of the tree contains the reference sequence and occurs with a frequency equal to one. We refer to the reference sequence as the *wildtype*.

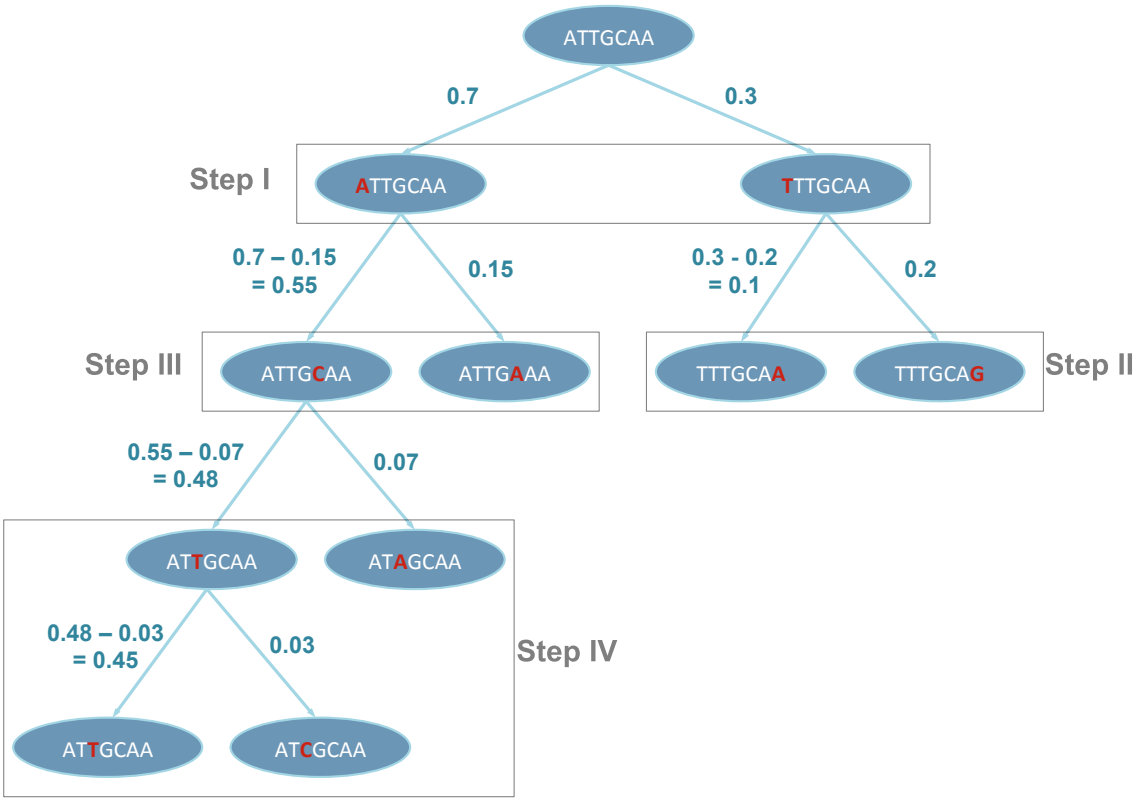
- We consider the polymorphisms in descending order with regards to their frequencies and only one polymorphism at a time. If multiple polymorphisms occurred at a certain position, then we consider the sum of all polymorphisms at this position to determine the order. (Note: Only the nucleotides that differ from the reference sequence are taken into account. In most cases the nucleotide that coincides with the reference sequence shows the highest frequency.)

- To incorporate a polymorphism into the tree a parent node is chosen at random from the set of current leaves taking their frequencies/weights into account:
  The set of current leaves of the tree is the set of possible parent nodes for the next SNP. Note, that the sum of the weights of all current leaves is equal to one and the weight of a node can be interpreted as the probability for the node to occur. To decide on the next parent node we draw a random number $x \in [0, 1)$. Suppose we currently have l leaves denoted by $n_0, \ldots, n_{l-1}$ and that they were numbered from left to right as they occur in the tree. Then the next parent node is the leaf

n_j such that

$$\sum_{i=0}^{j-1} n_i \leq x < \sum_{i=0}^{j} n_i \qquad , 0 \leq j < l \tag{1}$$

- Two new child nodes are added for each SNP at this position.
  - Suppose there is only one SNP present at the current position. Then we add two child nodes such that the left one contains the sequence of the parent node and the right one contains the sequence with the SNP.
  - If there is more than one polymorphism at a single position, then we consider them separately and start with the most frequent one and add it in the same way as described above. The parent node for the second SNP is the left child node, which was added during the incorporation of the first SNP. If there is a third polymorphism at this position we proceed analogously by choosing the left child node of the second polymorphism as the parent node.
- Weights:
  - Wildtype: The wildtype is very stable and usually dominant within the population between two bottleneck events. It is already adapted to the current environmental conditions and will have a high replication rate. This is taken into account by temporarily decreasing the weight of the wildtype by a constant factor (e.g. for the following simulation we chose a factor of 0.01). Thus the wildtype is less likely to be chosen as the haplotype that accommodates the next mutation. This ensures that the algorithm preserves the wildtype. We normalise the weight distribution of all nodes to account for the decreased weight of the wildtype node.
  - Incorporation of the polymorphism: If the weight of the parent node is larger than the frequency of the SNP, then we obtain the weight of the left child node by simply subtracting the frequency of the SNP and the weight of the right child node corresponds to the frequency of the SNP. If the weight of the parent node is too small, then we choose an additional parent node at random (as described above) to incorporate this SNP with the remaining frequency value.
- After considering all observed polymorphisms, the leaves of the tree correspond to the haplotypes present in the population and the weights of the leaves correspond to the frequencies of the haplotypes.



		A	C	G	T	Step
1	**A**	0.70	0.00	0.00	0.30	I
2	**T**	0.00	0.00	0.00	1.00	
3	**T**	0.07	0.03	0.00	0.90	IV
4	**G**	0.00	0.00	1.00	0.00	
5	**C**	0.15	0.85	0.00	0.00	III
6	**A**	1.00	0.00	0.00	0.00	
7	**A**	0.80	0.00	0.20	0.00	II

Figure 4.2: Illustration of the tree construction for the simulation of a possible set of haplotypes: a tree is constructed by iteratively adding one SNP in each step. The set of leaves of the final tree represent the set of haplotypes in the population. The table specifies the frequencies of the polymorphisms at each position.

Illustrative example (see Figure 4.2):

- Suppose we want to construct a tree for the following small part of a reference sequence:

  ATTGCAA

- Polymorphisms were observed as shown in the table in Figure 4.2. For simplicity the coverage is omitted and frequency values with two decimal places were chosen.

- The root of the tree contains the reference sequence "ATTGCAA" and is assigned a weight equal to one.
- The polymorphisms (e.g. the nucleotides that differ from the reference sequence) are considered in descending order according to their frequencies (as shown in the table in Figure 4.2). Here, the nucleotides coinciding with the reference sequence showed the highest frequency at each position - these are not considered for determining the order.
- Step I: Incorporate the most frequent polymorphism, i.e. the polymorphism occurring at position 1. The root is currently the only leaf and thus chosen as parent node. There is only one SNP at position 1, thus two nodes are added - the left child node contains the sequence without the SNP (i.e. the sequence of the parent node), the right node contains the sequence with the SNP ("TTTGCAA"). See the tree in Figure 4.2 for further details.
- Step II: The polymorphism at position 7 in the genome is considered next. Suppose a random number was drawn such that the node "TTTGCAA" is chosen as the next parent node. (The wildtype node is the leftmost leaf in the tree. For the parent selection its weight was temporarily decreased by a constant factor.) Add two child nodes in the same way as in step I.
- In step III we proceed analogously and assume that the random number yields the node "ATTGCAA" as the next parent node.
- In step IV we have to incorporate two SNPs into the sequence. We consider them separately and start with the most frequent one, where an "A" was observed in 0.07% of all reads. Assuming the random number yields the node "ATTGCAA" as the next parent node, two child nodes are added accordingly to this node. The left child node contains the sequence of the parent node and the right child node contains the sequence with the SNP. The parent node for the second SNP (where a "C" was observed in 0.03% of all cases) is the left child node, which was added for the first SNP and another two child nodes are added analogously.

4.4 The Experimental Foot-and-Mouth Virus Data Set

The foot-and-mouth virus is a RNA virus - a type of virus that evolves rapidly due to its high replication rate and poor proofreading ability. The single bases of the genome show high sequence variability while different haplotypes often differ by only a few nucleotides. The high mutation rates make it extremely hard to target the virus for treatment or prevention of the disease and the outbreak in the United Kingdom in 2001 showed that

the FMV still poses a major economical risk in livestock producing countries. During the outbreak over 10 million sheep and cattle were killed to stop the disease. The crisis resulted in a loss of £3.1 billion to agriculture and the food chain and it is estimated that another £3 billion was lost in tourism as a result [150].

We used an experimental data set that was part of a study by Wright et al. [160] on the within-host diversity of the virus. They took samples of the viral populations from two feet lesions and the inoculum (mouth) of a single animal and sequenced them. Reverse transcription was performed and the samples were amplified with the help of polymerase chain reaction (PCR). Each of the three samples was split in half and analysed on different sequencing runs on the Illumina Genome Analyzer (GA). The first run produced reads of about 50bp, the second run was performed after an upgrade of the GA and yielded longer reads of about 70bp. The reference genome was available from previous studies using Sanger sequencing (GenBank accession no. EU448369). The reads were trimmed and filtered and the longer reads from the second run were additionally trimmed to a length of 50 nucleotides to enable a direct comparison of the two runs [160]. Also, the first and last five nucleotides of each read were more error prone and thus removed, producing reads of about 40 bp for each run. In addition, any read was discarded if its average error rate per nucleotide based on the quality scores exceeded 0.2%.

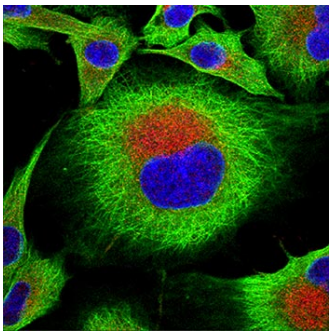


Figure 4.3: FMV (red) during replication near the nucleus (blue) of an infected cell. (Taken from [18].)

The input data on which we base the simulated population needs to comprise the following information: the reference nucleotide needs to be specified for every position of the genome, we need the sequencing coverage at the respective position and the frequencies for observing substitutions with A, C, G and T, respectively, for every position. This information can, for example, be obtained by aligning the reads against the reference genome.

Example:

A typical row in the input data provides the following information:

pos	ref	cov	A	C	G	T
1787	A	8690	0.997238	0.000000	0.002417	0.000345

Table 4.1: Typical row of input data for the algorithm.

At position 1,787 of the genome we find an "A" in the reference genome. This position was covered by 8,690 reads. In $\approx 99.72\%$ of the cases an "A" was observed, in $\approx 0.24\%$ a "G" was observed and in $\approx 0.03\%$ it was a "T". ("C" was never observed.)

Next generation sequencing allows us to identify SNPs, even if the SNPs are only present in a small fraction of the population, and thus facilitates new insight into the evolution and selection process in a viral population. For more details on the data set and previous analyses see [160].

4.5 Adjusting the Reference Sequence and Filtering the Data Set

The reference sequence (EU448369) showed a long homopolymer of 12bp flanked by 5 unknown nucleotides on each side. We compared this part of the sequence to other foot-and-mouth virus genomes available on GenBank. Carrillo et al. [48] sequenced and analysed 103 isolates of the foot-and-mouth virus. Their analysis revealed a poly(C) tract (90% C) of 100 to 420 nucleotides. The genome sequences they found and our reference sequence only differed by a few SNPs and our reference sequence additionally lacked the first 17bp at the start of the genome. Direct comparison with one of the genomes (AY593816.1) shows a perfect alignment with our reference sequence across the first 1,000bp including the homopolymer. We replaced the unknown nucleotides in our reference accordingly. In addition we removed the poly(A) tail for our further analysis resulting in an overall length of 8,162bp.

We based the construction of the *in silico* haplotype population on the first sequencing run of the first samples [160]. SNPs occurred at more than 75% of all positions in the genome. We calculated the upper bound of the mutation rate for the foot-and-mouth virus using the formula by Nowak [116]. In particular RNA viruses seem to have mutations rates that are very close to this error threshold [118]. Beyond this threshold the population is not viable as essential genetic information cannot be maintained. Nowak also assumes that random mutations as well as selection are the major factors that shape a viral population and that selection acts on the quasi-species as a whole rather than on a single haplotype.

Observed mutation rate:

First, we calculated the average accuracy rate per base. For this we assumed that the consensus sequence corresponds to the genome from which the haplotypes originated. Then the probability that the base is copied correctly corresponds to the frequency with which the reference nucleotide was observed at this position. Averaging over all positions

in the genome we obtained the following average accuracy per base for the data set:

$$P(\text{no mutation}) = 0.986617 \text{ per base per generation}$$

Thus the mutation rate for the foot-and-mouth virus data set is:

$$P(\text{mutation}) = 0.013383 \approx 1.34 \times 10^{-2} \text{ per base per generation} \tag{2}$$

Upper threshold for the mutation rate:

According to Nowak [116] the upper threshold for the mutation rate - beyond which the haplotypes are not viable - is:

$$\begin{aligned} m &< \frac{1}{1-q} \\ \Leftrightarrow \quad q &> 1 - \frac{1}{m} \end{aligned}$$

where m is the length of the genome and q is the per-base accuracy of replication.

With $m = 8,162$bp for the foot-and-mouth virus we obtain:

$$q > 0.999877481 \text{ per base per generation} \tag{3}$$

This corresponds to an upper threshold for the mutation rate of approximately

$$1.22 \times 10^{-4} \text{ per base per generation} \tag{4}$$

which is similar to mutation rates reported for other RNA viruses (e.g. hepatitis C) [47] but is much smaller than the average mutation rate (see Equation (2)) for the foot-and-mouth virus data set.

There are several possible explanations for this. A very high mutation rate produces many non-viable and inactive haplotypes. Nonsense or missense mutations can be deleterious for the haplotypes if they affect essential functions. In addition, some of these "observed SNPs" might be errors which occurred during the sequencing process or as a result of PCR amplification. Taking the coverage into account, we checked how often a mutation was actually observed. We then filtered the data set and only took mutations into account that were observed at least twice. This makes it less likely that the SNP is a sequencing error and can be interpreted as a sign that the haplotype replicated and is thus viable.

Mutation rate after filtering:

After filtering, the number of polymorphisms in the data set decreased by approximately 50% and the average accuracy rate per base increased to

$$q \approx 0.9994353$$

which corresponds to a mutation rate of

$$5.65 \times 10^{-4} \tag{5}$$

This mutation rate is very close to the error threshold (4) and similar to mutation rates reported by other authors [160] [69].

4.6 An *In Silico* Foot-and-Mouth Virus Haplotype Population

We used our algorithm to create a haplotype population that is based on the SNP frequencies of the filtered foot-and-mouth virus data set (see Section 4.5) with a factor of 0.01 for the wildtype (see Section 4.3). The high coverage (×4,873) of the FMV data set provided a good basis for the simulation as it can reveal SNPs even for low abundant haplotypes. The filtered data set contains 5,479 SNPs occurring at 3,952 positions of the foot-and-mouth virus genome. (Note, that up to three SNPs can occur at the same position in the genome.) SNPs were observed at ≈ 48.42 % of the position in the genome and all SNPs were observed at least twice.

The algorithm constructed a population with 4,359 haplotypes with an average of six polymorphisms per haplotype. The maximum was 22 polymorphisms in a haplotype. Figure 4.4 shows the distribution of polymorphisms per haplotype across the data set. The majority of haplotypes have 4-13 SNPs. In our simulated population the wildtype appears with a frequency of ≈ 0.2796 and the majority of the haplotypes are present at very low frequencies of <0.01%.

4.7 Discussion and Future Work

Our algorithm yielded a population with a highly complex structure that reflects the number of SNPs observed in an experimental data set. Although the simulation is based on an experimental data set, it is difficult to verify whether the simulated population reflects the structure of the true population. Nevertheless, the simulated data set will be useful for our benchmarking study. With more than 4,000 haplotypes this data set will be challenging for the viral haplotype reconstruction programs. It will be interesting to see how much the programs are able to reconstruct under these conditions and to study the

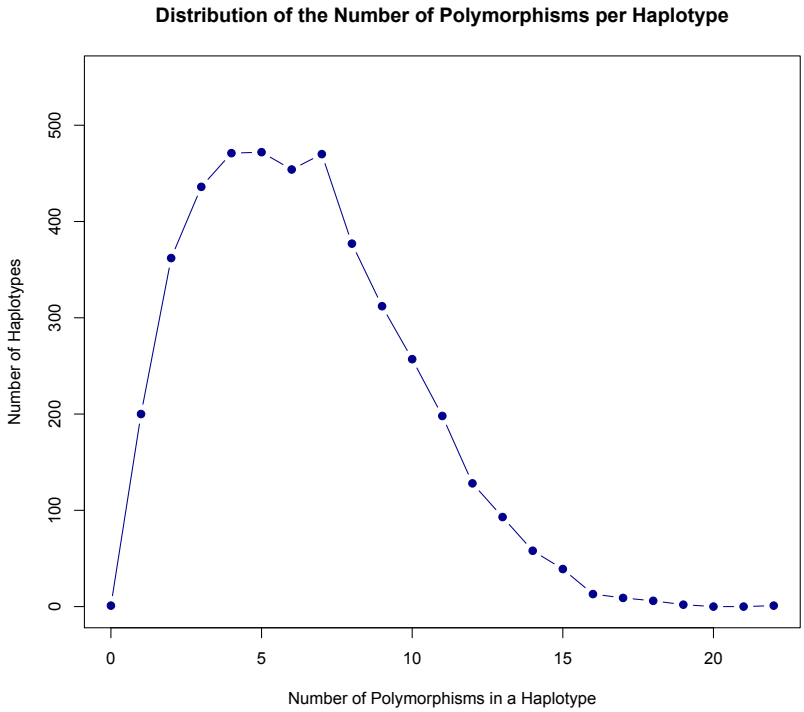


Figure 4.4: Distribution of the number of polymorphisms per haplotype for a simulated foot-and-mouth virus population.

impact of many low abundant haplotypes on the reconstruction - in particular the effect on the false positive rate. Currently, we do not know what a "typical" viral community looks like. But with one billion estimated variants occurring in each individual multiple times a day, we expect to find highly complex populations in nature.

Wright et al. [160] applied a simple error correction approach. We showed that the mutation rate based on their error-corrected reads exceeded the upper threshold beyond which haplotypes are not considered to be viable. The development of more sophisticated error models is a crucial step as it will greatly improve the quality of the data and can ensure that the observed mutations are true SNPs. We will discuss various error correction and removal strategies in Chapters 6 and 7.

For our algorithm we chose the next haplotype that will encounter a mutation at random and only take the haplotype frequencies into account. As we only model mutations that are assumed to produce viable haplotypes, the mutated haplotype is assigned a frequency greater than zero and proportional to the frequency of the SNP. If we only take the abundance levels of the currently present haplotypes into account for choosing the candidate for the next mutation, then the haplotypes of the resulting population show very similar frequencies and the wildtype is no longer dominant in the population. We introduced a weight in order to preserve the wildtype (see Equation 4.3). The wildtype is adapted to the environmental conditions and should (at least initially) show

a greater level of fitness than the mutants. This could, for example, lead to a faster replication rate. An extended model that incorporates different levels of fitness for the haplotypes might provide further insight into the population structure.

Our simulation algorithm assumes that the infection was caused by a single strain (the wildtype) and for our simulation we worked with an experimental data set where the infections was caused by a single FMV strain. A possible extension of the model would be the simulation of infections that are caused by multiple strains. Also, the introduction of multiple SNPs in the same step could change the population structure and is subject to future research.

5 Benchmarking of viral haplotype reconstruction programs: an overview of the capacities and limitations of currently available programs

5.1 Abstract

Viral haplotype reconstruction from a set of observed reads is one of the most challenging problems in bioinformatics today. Next generation sequencing (NGS) technologies enable us to detect single nucleotide polymorphisms (SNPs) - even if the haplotypes appear at low frequencies. However, there are two major problems. First, we need to distinguish real SNPs from sequencing errors. Second, we need to determine which SNPs occur on the same haplotype, which cannot be inferred from the reads if the distance between SNPs on a haplotype exceeds the read length. We conducted an independent benchmarking study that directly compares the currently available viral haplotype reconstruction programs. We also present nine *in silico* data sets that we generated to reflect biologically plausible populations. For these data sets we simulated 454 and Illumina reads with our own metagenomic read simulation program and applied the programs to test their capacity to reconstruct whole genomes and individual genes. We developed a novel statistical framework to demonstrate the strengths and limitations of the programs. Our benchmarking demonstrated that all the programs we tested performed poorly when sequence divergence was low and failed to recover haplotype populations with rare haplotypes.

This chapter is based on the publication:

> Melanie Schirmer, William T Sloan, and Christopher Quince. **Benchmarking of viral haplotype reconstruction programs: An overview of the capacities and limitations of currently available programs.** (Briefings in Bioinformatics, page bbs081, 2012.)

Original Contributions

Figure 5.1 provides an overview of the individual steps of the benchmarking. I conducted the first independent benchmark study of viral haplotype reconstruction programs. My study offers a direct comparison of the available programs and highlights their capabilities and limitations. I designed and simulated various *in silico* populations that cover a broad range of conditions based on observations from real data sets. The data sets cover varying levels of sequence divergence, population size, whole genome versus single gene analysis and different abundance distributions. They will not only be useful

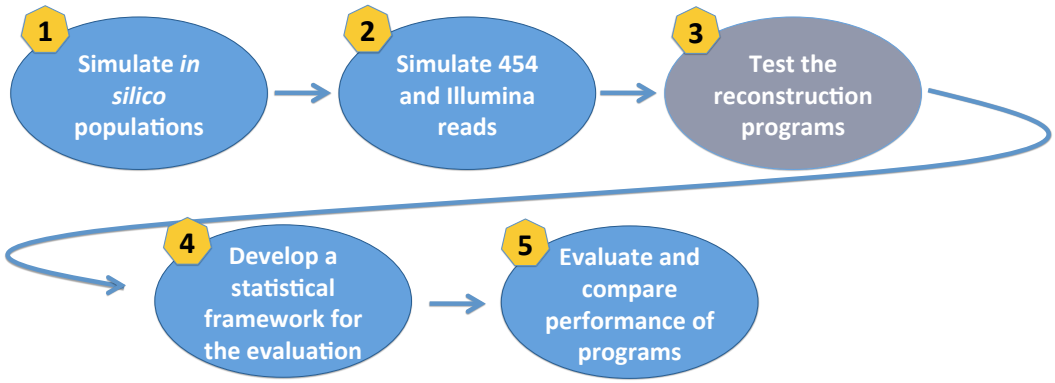


Figure 5.1: Project overview of the benchmarking study. My original contributions are marked in blue.

for my benchmarking study but also for the evaluation of other programs that aim to infer information about the underlying population structure based on next generation sequencing data. One of my main original contributions presented in this chapter is the development of a novel statistical framework for the evaluation of the reconstructed populations where I assessed the accuracy of the reconstructed haplotypes and simultaneously penalise the reconstruction of false-positives.

5.2 Introduction

RNA viruses are among the most dangerous pathogens for humans and animals. Human immunodeficiency virus (HIV), hepatitis C virus (HCV) and the foot-and-mouth virus (FMV) are just some examples of RNA viruses that pose major health threats. There is no general treatment available and for many viruses we have not been able to develop effective vaccines. RNA viruses are able to mutate quickly and cause acute epidemics with novel strains. In order to develop successful and preventive treatments and to act quickly when a new viral strain occurs, we need more detailed and accurate information about the population structure, the mutations and the viral haplotypes for the specific infection. The development of next generation sequencing (NGS) technologies opens up the opportunity to respond quickly to outbreaks and gain a better understanding of viral populations. However, there are significant challenges that we need to overcome to reconstruct viral populations from NGS data.

The lack of proof checking during replication causes RNA viruses to have mutation rates about a million times larger than within human cells [118]. This results in a population of closely related genomes, a so-called *quasi-species*. Bull et al. [47] studied the RNA virus hepatitis C. They predict that about 10^9 variants with one or two single nucleotide polymorphisms (SNPs) are likely to arise in each individual multiple times a day. But they also report that the actual observed complexity is much lower, which is probably

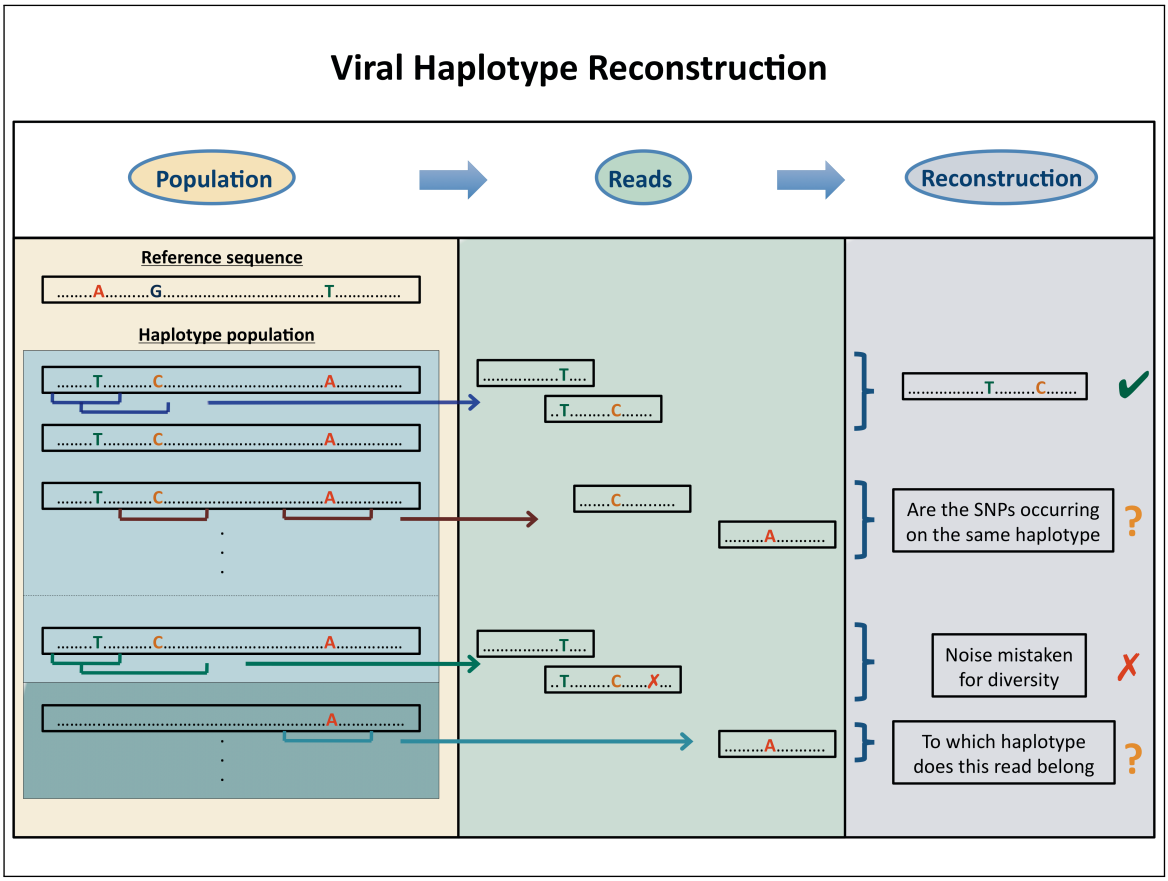


Figure 5.2: Schematic diagram representing the process of reconstructing viral haplotypes from next generation sequencing reads.

due to the reduced fitness of many mutated variants. In their test series more than half of the substitution events occurred at frequencies of <1% (except for one test subject). They estimated the mutation rate of hepatitis C to be around 1.2×10^{-4}. We compare this rate later to the nucleotide substitution rate of an experimental foot-and-mouth virus data set.

The high mutation rate is dangerous for the virus, since it results in many non-viable clones, but it also provides the virus with a large number of potentially beneficial mutations allowing it to adapt quickly to changing environments during infection. It is likely that the haplotypes that enable the virus to survive selective pressure pre-exist in the population [42]. Thus, it is essential to determine all haplotypes in order to develop effective treatments and vaccines.

With NGS technologies we are now able to detect SNPs in a viral population - even for low abundance haplotypes. However, reads from any sequencing technology contain platform specific noise, which we need to distinguish from real diversity in order to be able to reconstruct the haplotypes accurately. Another major challenge arises due to the short lengths of NGS reads, which can make it difficult, and sometimes impossible, to

Table 5.1: Comparison of different 454 and Illumina sequencing instruments in 2012 (adapted from [67]&[21]).

Instrument	Run time	Millions of reads/run	Read length	Yield Mb/run	Reagent cost/run	Reagent cost/Mb
454 FLX Titanium	10 h	1	400[1]	500	$6,200	$12.40
454 FLX +	18-20 h	1	700[1]	900	$6,200	$7.00
Illumina HiSeq 2000	8 days	1,000	36-100[2]	200,000	$20,120	$0.10
Illumina GAIIx	14 days	320	35-150[3]	96,000	$11,524	$0.12
454 GS Junior	10 h	0.10	400[1]	50	$1,100	$22
Illumina MiSeq	26 h	3.4	25-150[4]	1,200	$750	$0.74

[1] Average read length
[2] Possible read lengths: 36bp, 50bp, 100bp
[3] Possible read lengths: 35bp, 50bp, 75bp, 100bp, 150bp
[4] Possible read lengths: 25bp, 35bp, 100bp, 150bp

determine which SNPs reside on the same haplotype. The three columns in Figure 5.2 give a schematic of the different steps during the process of reconstructing a population from NGS data. In the first column we can see two haplotypes occurring at different abundances. They have one SNP in common. The next column displays a set of observed reads obtained from NGS technologies including sequencing noise. The third column presents different scenarios that can occur during the reconstruction. In the first scenario the reconstruction is successful. We encounter two reads that contain SNPs and have a sufficient overlap to be assembled correctly into a contig of the first haplotype. In the second scenario, the distance between SNPs exceeds the read length which means we cannot map the reads to a haplotype based on read overlap. In the third scenario, noise is mistaken for diversity. And in the forth, we cannot infer the origin of the read as the SNP occurs on both haplotypes.

Here, we study the currently available haplotype reconstruction programs and benchmark their performance across various *in silico* data sets. It is important to know their capabilities and highlight scenarios that might expose their limitations. Our test data sets were deliberately selected to challenge the reconstruction programs with different sequence divergences and haplotype abundance distributions. Thus we could assess the programs' abilities to reproduce the underlying population structure. We also evaluated the accuracy of the reconstructed haplotypes by outlining how many haplotypes are reconstructed with zero, one and two mismatches. It is important to identify the number of false positives in the reconstructed population to assess the overall accuracy of the reconstruction.

We simulated 454 and Illumina reads for all of the test data sets with our metagenomic read simulation program introduced in Chapter 3. Table 5.1 compares the run time (for

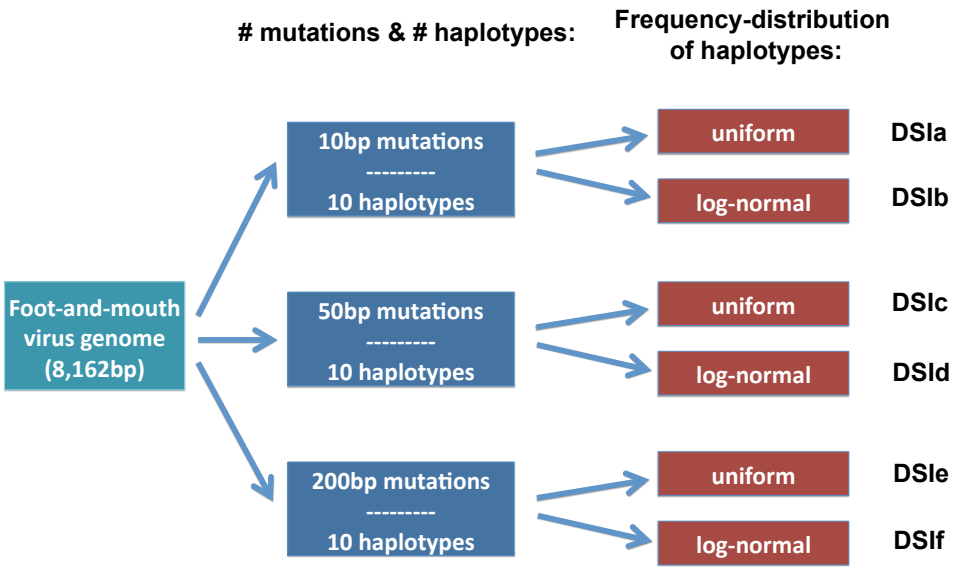


Figure 5.3: Overview of the simulation of data sets DSIa - DSIf. We introduced 10, 50 and 200 mutations, respectively, into the foot-and-mouth virus genome and simulated 10 haplotypes for each population. The haplotypes were mixed according to a uniform and a log-normal distribution resulting in six data sets in total.

maximum read length), number of reads per run, read length, yield per run and reagent cost for Illumina and 454 sequencers. The great advantage of 454 reads is that they are longer than Illumina reads which could be important (as seen in Figure 5.2). On the other hand, Illumina reads provide much higher coverage at much lower cost. None of the programs that we tested was specifically designed for Illumina reads, however all of them take Illumina fasta files or fastq files. We tested the programs on the Illumina reads to examine their potential for viral haplotype reconstruction.

5.3 The Test Data Sets

We generated nine *in silico* haplotype populations that vary in sequence diversity and number of haplotypes. In addition, different frequency distributions were considered for these populations. The data sets are based on observations from real FMV and HIV data sets. For FMV, we targeted whole genome reconstruction and for HIV we considered single gene reconstruction. The details of the data sets, including the sequence divergence and frequency distributions of the haplotypes, can be found in Table 5.2.

Simulating the evolution of a single foot-and-mouth virus sequence (DSIa - DSIf)

We created six data sets based on the experimental foot-and-mouth virus data set (FMVD) introduced in Chapter 4. Nucleotide substitution rates are organism specific, gene specific, vary between the four nucleotides (e.g. purine-purine substitutions are more likely than purine-pyrimidine substitutions) and depend on the codon position

Table 5.2: Overview of all test data sets, including the number of haplotypes in the population, the number of mutations on each haplotype and their frequency distributions. We used the Levenshtein distance to evaluate the pairwise sequence divergence between haplotypes. The Levenshtein distance is the minimum number of substitutions and indels to turn one sequence into another. Note, that the reference sequence is not part of the haplotype population and that the same mutation can occur on more than one haplotype.

Data sets	# haplot.	genome size	# mutations per haplotype	divergence	freq. distr.
DSIa	10	8,162bp	10bp	0.23%	uniform
DSIb	10	8,162bp	10bp	0.23%	log-normal
DSIc	10	8,162bp	50bp	1.12%	uniform
DSId	10	8,162bp	50bp	1.12%	log-normal
DSIe	10	8,162bp	200bp	3.98%	uniform
DSIf	10	8,162bp	200bp	3.98%	log-normal
DSIIa	44	2,256-2,581bp	2-328bp	0.08%-12.71%	uniform
DSIIb	44	2,256-2,581bp	2-328bp	0.08%-12.71%	log-normal
DSIII	4,359	8,162bp	1-41bp	0.01%-0.50%	empirical

[161][146]. The situation is even more complicated in the case of viruses as genes can overlap. Many models work with a substitution matrix where changing probabilities between nucleotides can vary depending on the initial base and the mutated base. We went one step further for the FMVD and inferred position specific substitution rates for the entire genome.

We started with the consensus sequence of the foot-and-mouth virus and simulated its evolution into ten haplotypes by introducing mutations at 10, 50 and 200 positions in the genome to create data sets of varying diversity. The position specific nucleotide frequencies from the experimental data set can be interpreted as a discrete probability distribution, giving the probability for each nucleotide (A, C, T and G) to occur at a specific position in the genome. For each mutation, we chose a position in the genome at random with probabilities proportional to the number of observed SNPs. A second random number specified the mutation according to the probability distribution for this position in the following way:

Let $x \in \mathcal{U}[0,1)$ be a random number. We denote with m_1, m_2 and m_3 the possible mutations of the reference nucleotide into one of the other three nucleotides. We then choose the mutation m_j such that

$$\sum_{i=1}^{j-1} P(m_i) \leq x < \sum_{i=1}^{j} P(m_i) \qquad \text{with } j \in \{1,2,3\} \tag{6}$$

Example:

Here, we consider the example presented in Table 4.1 of section 4.4. The reference nucleotide is an "A" at this position and mutations into "G" and "T" were observed at least twice, a mutation into "C" was never observed. We already know that a mutation occurs at this position and we know the relative frequencies of "C", "G" and "T" for this position. Normalising yields the following discrete distribution:

C	G	T
0.000000	0.875091	0.124909

Table 5.3: Normalised frequency distribution.

Now we draw a random number x $\sim \mathcal{U}[0,1)$ which determines the mutation for this position according to equation (6). Here m_1 denotes the change of A into C and $P(m_1) = 0$, m_2 denotes the change of A into G and $P(m_2) = 0.875091$ and m_3 is the change of A into T and $P(m_3) = 0.124909$.

For each of the three sets of haplotypes we simulated a population with *uniformly distributed* haplotype abundance levels and *log-normal distributed* abundance levels. See Figure 5.3 for an overview of the simulation process. The log-normal distribution is visualised in Figure 5.4.

The HIV-1 envelope gene (DSIIa & DSIIb)

We generated two *in silico* populations of the envelope glycoprotein (env) gene of HIV-1 from data used to asses whether an HIV-1 infection was initiated by a single or multiple viral strains [94]. For one of the patients 44 sequences were isolated (GenBank accession number EU577344 - EU57787). We used the first sequence as a reference sequence. In comparison to the reference sequence 40 sequences had between 2 and 4 mutations, one had 6, another had 18 and one had major deletions that resulted in a Levenshtein distance of 328 (see Figure 5.5). We constructed populations from these sequences where the abundances were distributed uniformly and log-normally ($\mu = 1$, $\sigma = 2$).

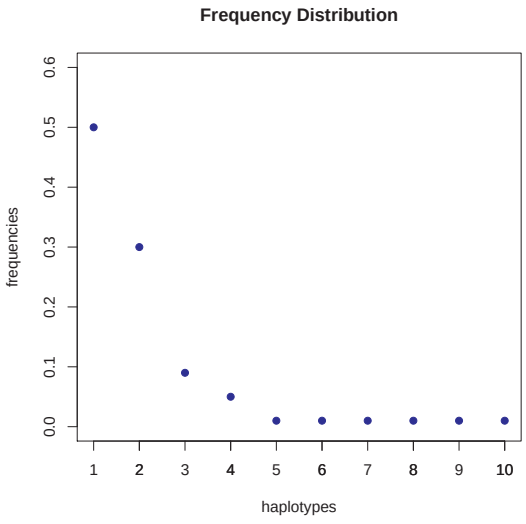


Figure 5.4: Frequency distribution of the haplotypes for data sets DSIb, DSId and DSIf.

An in silico foot-and-mouth virus population (DSIII)

We generated a more complex *in silico* viral population from the experimental FMVD. A detailed description of the algorithm and the data set can be found in Chapter 4. The SNPs and their frequencies in this *in silico* data set are consistent with the experimental data. The algorithm yielded a population of 4,359 haplotypes with an average of six polymorphisms per haplotype and a maximum of 22 polymorphisms within a single haplotype where the same SNP can occur on multiple haplotypes. The wildtype is dominant in the population and appears with a frequency of ≈0.2796 and the majority of the haplotypes are present at very low frequencies making up less than 1% of the whole population. This data set is very challenging for a haplotype reconstruction program, however highly complex haplotype populations occur in nature.

454 read data sets

For each of the data sets DSIa - DSIf FlowSim (V 0.3) was used to generate 120,000 reads with an average read length of 492bp that included 454 sequencing noise and PCR noise. PCR noise was added, since the genomes or part of the genomes in viral samples are often PCR amplified prior to sequencing to increase the amount of input DNA. This provides a high coverage for this data set:

$$c = n \times \frac{l}{g} = 120,000 \times \frac{491.876}{8,162} \approx 7,232$$

Here c denotes the *coverage*, n and l denote the *number of reads* and the *average read length*, respectively, and g denotes the *length of the genome*. Note, that the adapters were removed before the reads were inputted into the programs. For DSIIa 40,000 reads were generated.

We used the functions "clonesim" followed by "kitsim" and "flowsim" to simulate 454 reads including sequencing noise. Clonesim simulates the shearing step. Kitsim attaches synthetic sequences to the ends of each clone to simulate the emPCR primers contained in the 454 Titanium adapters. In the last step the function flowsim converts the input clones into a series of light signals and the flow values of the homopolymer lengths are adjusted according to a flow distribution. In the last step quality filters are applied and the output is an sff file.

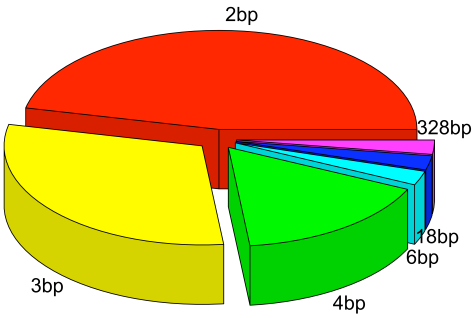


Figure 5.5: Sequence divergence of the 44 HIV-1 sequences.

We used our own metagenomic read simulation program to generate 454 reads for the

more complex non-uniform distributions as none of the existing programs at the time were able to take different abundance levels into account. The read lengths follow a normal distribution ($\mu = 500\text{bp}, \sigma = 80\text{bp}$) which reflects an experimental distribution. We simulated 40,000 reads for DSIIb and 120,000 reads for DSIII, again with 454 and PCR noise.

Table 5.4: Overview of the currently available haplotype reconstruction programs. The first three programs (ShoRAH, PredictHaplo and QuRe) were included in our benchmarking.

Program	Whole gene/ genome reconstr.	Method
ShoRAH [162]	✓	- read-graph approach in combination with solving the maximum weight matching problem for path selection - reconstructs a minimal set of haplotypes - Dirichlet process mixture model for error correction
PredictHaplo [4][125]	✓	- non-standard clustering problem in combination with a Dirichlet process mixture model - reconstructs the most likely set of haplotypes
QuRe [127]	✓	- error correction according to a Poisson distribution with different parameters for homopolymeric & non-homopolymeric regions - infers multinomial distribution for locally reconstructed haplotypes and uses those for global reconstruction
ViSpA [33] *	✓	- read-graph approach, where the most probable paths in the graph are selected - reconstructs the most likely set of haplotypes
V-Phaser[102] & V-Profiler[72]	– **	- V-Phaser uses covariation & an EM alg. to recalibrate quality scores to detect the SNPs for every position - V-Profiler calculates the frequency of each triplet codon of the accepted nucleotides and constructs the haplotypes
QuasiRecomb [163]	– **	- jumping hidden Markov model taking recombination events and SNPs into account - samples the haplotypes from the inferred distribution of haplotypes

* could not be installed and is unsupported
** only attempts reconstruction over a small local window (window size < read length)

Illumina read data sets

We used our program to simulate 75bp Illumina reads and utilised the corresponding Illumina error profile from [75]. We simulated one million reads for the whole-genome test data sets (DSIa - DSIf and DSIII) and 400,000 reads for the single-gene test data sets (DSIIa and DSIIb).

5.4 Existing Viral Haplotype Reconstruction Programs

Table 5.4 gives an overview of the currently available viral haplotype reconstruction programs (date: November 2012). Four of these programs attempt whole gene/genome reconstruction: ShoRAH, PredictHaplo, QuRe and ViSpA. Two of them, V-Phaser/V-Profiler and QuasiRecomb, attempt reconstruction over local windows that are smaller than the read length. Here, we only benchmarked whole gene/genome programs. Unfortunately ViSpA could not be installed and is unsupported, hence three programs were benchmarked.

ShoRAH

The name ShoRAH stands for **Sh**ort **R**ead **A**ssembly into **H**aplotypes [162]. ShoRAH can take 454 or Illumina reads in fasta format as input and three parameters: the parameter a, the window size w and the number of iterations j. We used version 0.5.1 of ShoRAH. ShoRAH performs read alignment, error correction, global haplotype reconstruction and frequency estimation. Errors are corrected with a Dirichlet Process Mixture Model (DPMM) and local haplotypes are reconstructed over "small" windows. They use the read-graph approach described by Eriksson et al. [60] for the global reconstruction. Following a parsimony principle ShoRAH tries to reconstruct the smallest set of haplotypes that explains the observed reads by solving a maximum weight matching problem on the constructed read-graph.

PredictHaplo

We used version 0.2 of PredictHaplo for the 454 reads and version 0.4 for Illumina. The program takes a reference sequence and a fasta or fastq file containing the reads as input. In addition known "true" haplotypes can be passed to the program as well as the maximum read number for the local windows and an "entropy threshold" that specifies the smallest expected haplotype frequency. The haplotype reconstruction problem is treated as a non-standard clustering problem where the reads are the data points and the unknown haplotypes the cluster centroids. Local reconstruction starts at the point of highest coverage over a window small enough such that all reads overlap and the window size is increased in every iteration step. A DPMM is used to infer the unknown number of haplotypes and a Gibbs sampler to determine the haplotypes.

QuRe

The reads are inputted in fasta format together with a reference sequence against which the reads are aligned with the Smith-Waterman-Gotoh local alignment algorithm. The errors are assumed to follow a Poisson distribution with different parameters for homopolymeric and non-homopolymeric regions. The reads are first corrected and then local haplotype reconstruction is performed on an optimal overlapping window with maximum read coverage and maximum sequence divergence where the window size does not exceed the read length. The local frequencies form a multinomial distribution that is used together with the information from overlapping reads to achieve global reconstruction. The algorithm uses a variation of Zagordi's probabilistic clustering algorithm to infer a probabilistic clustering of the reconstructed haplotypes.

ViSpA

The **Vi**ral **Sp**ectrum **A**ssembler works with a reference sequence and a set of 454 reads. Placeholders are used for insertion and deletions if reads cannot be aligned uniquely and sequential multiple alignment is performed against the "extended" reference sequence. A consensus sequence is constructed according to the majority of aligned reads. The reference sequence is iteratively replaced by the consensus sequence in order to align reads that could not be aligned previously. Reads are then corrected and a read-graph is constructed as described in [60]. Each path in the read-graph from the source to the sink corresponds to a possible haplotype that is consistent with the observed (corrected) reads. By weighting the consensus of the reads and estimating the probability that two overlapping reads belong to the same sequence, they select the most probable paths in the read-graph. An expectation maximisation (EM) algorithm is used to estimate the frequencies of the sequences.

V-Phaser

The current version of V-Phaser is implemented in Perl. The program aims at recognising phased variants. An EM algorithm is used to recalibrate base quality scores in every iteration. Reads are then corrected by phase and quality filtering and the algorithms looks for patterns of variants in phase. A composite Bernoulli model is used to incorporate individual base quality scores, allowing error rates to vary from base to base. As input the program requires a .qlx alignment format for which the software RC454 is recommended which corrects read errors. It requires the fasta and qual files of a 454 data set and a consensus assembly for which the program AV454 (a module of the Arachne assembler) is recommended. The V-Phaser script is then applied to the final qlx file. The output is required by the V-Profiler script from which the haplotypes can be obtained.

5.5 Measures for the Evaluation: Similarity & Completeness

Two novel measurements were developed to evaluate the accuracy of the haplotype reconstruction programs. These measures not only take the number of successfully reconstructed haplotypes into account but also their frequencies, the number of mismatches, the number of false positives and the reconstructed length. We first introduce a measurement for the similarity of the reconstructed haplotypes and the true haplotypes, where only the fraction of the genome is taken into consideration that is covered by the reconstruction; the second measurement reflects the completeness of the reconstructed haplotypes.

Similarity Measure (SiM)

We need a measurement that takes the distance of each reconstructed haplotype to its closest true haplotype into account and at the same time penalises the reconstruction of too many or too few haplotypes. We achieved this by defining a probability distribution based on the reconstructed population and the "true" population, respectively. We then use the Hellinger distance to quantify the similarity between the two distributions. Here, we only take the reconstructed part of the true haplotype sequence into account and measure the distance between two sequences by computing the *Levenshtein distance*. By allowing i mismatches for a reconstructed haplotype to "equal" a true haplotype we measure how close the reconstructed population is to the underlying "true" population.

Let P_1 denote the set of true haplotypes. Then the frequencies of the true haplotypes represent a discrete probability distribution over P_1, which we denote with f:

$$f : P_1 \rightarrow]0, 1]$$

The set of reconstructed haplotypes is denoted with P_2 and analogously the frequencies of the reconstructed haplotypes represent a discrete probability distribution g:

$$g : P_2 \rightarrow]0, 1]$$

We now take the sum of the two sets:

$$P := P_1 \cup P_2$$

which is the set of all true haplotypes combined with the reconstructed haplotypes that do not match any of the true haplotypes.

We can now extend the probability distribution $f(x)$ to the set $P = p_1, p_2, \ldots, p_{|P|}$ by setting the frequency of any haplotype $p_k \in P \setminus P_1$ to zero. We denote this distribution with $\tilde{f}(x) : P \rightarrow]0, 1]$. Analogously, we can define the extension of the probability

distribution $g(x)$ to P and denote it with $\tilde{g}(x) : P \rightarrow]0, 1]$. So the two distributions $\tilde{f}$ and $\tilde{g}$ overlap on the set of correctly reconstructed haplotypes.

Let $i \in \mathbb{N}$ denote the number of allowed mismatches. Then a reconstructed haplotype matches a true haplotype if the Levenshtein distance is $\leq i$. So for $i > 0$ an element $p_k \in P_1$ and an element $p_j \in P_2$ can be mapped to the same element in P if the Levenshtein distance $Ldist(p_k, p_j) \leq i$. When allowing mismatches, more than one reconstructed haplotype can match to the same true haplotype. In that case we add up the frequencies of all matching reconstructed haplotypes for the distribution $\tilde{g}(x)$.

We define the **Similarity Metric *(SiM)*** in terms of the *Hellinger distance* of the probability distributions $\tilde{f}$ and $\tilde{g}$ as follows

$$SiM_i := 1 - H(\tilde{f}, \tilde{g})$$

with:

$$H^2(\tilde{f}, \tilde{g}) = \frac{1}{2} \sum_{x=p_0}^{p_{|P|}} (\sqrt{\tilde{f}(x)} - \sqrt{\tilde{g}(x)})^2$$

So SiM_0 enforces strict similarity, where a reconstructed haplotype must match the "true" haplotype exactly. Also note that we have $0 \leq SiM_i \leq 1$ where zero corresponds to the maximal distance between two distributions (if none of the reconstructed haplotypes matches a true haplotype) and a similarity of one corresponds to the minimal distance (if the true population was exactly reconstructed).

Completeness Measure (CoM)

The completeness measure returns the average percentage of the sequence length that a reconstructed haplotype recovered of a true haplotype:

$$CoM = \frac{1}{n_{rec}} \sum_{i=0}^{n_{rec}-1} \frac{\text{length of reconstructed haplotype}}{\text{length of closest true haplotype}}$$

where n_{rec} denotes the total number of reconstructed haplotypes.

5.6 Benchmarking Results

We benchmarked PredictHaplo, ShoRAH and QuRe on 454 and Illumina reads. In general, PredictHaplo tended to underestimate the number of haplotypes for the larger populations, whereas ShoRAH greatly overestimated the number of haplotypes in the populations (e.g. for DSI: 25-55x the real number of haplotypes). The same was true for QuRe where the population size was up to 25x the size of the real population. Also, for

ShoRAH different sets of parameters yielded very different results on our data sets. On data set DSIc, for example, we ran ShoRAH with ten different sets of parameters. A Levenshtein distance of 1 was achieved for the *best* reconstructed haplotype on the "best run" but for the "worst run" the Levenshtein distance was 156. PredictHaplo requires fewer parameters and the effect of different choices on the Levenshtein distance is in the range of 1-2. QuRe takes three different parameters. Two of them are specific for the sequencing technology and default values are provided for 454.

The similarity measure for the 454 read data sets

We summarised the results in Table 5.5. The first two columns show our *in silico* data sets and the average mutual sequence divergence of the haplotypes in the respective population. It is important to interpret the results (especially the Levenshtein distance) with regards to the sequence divergence within the population. For each program we state the results for the best run (after testing the program with various parameters) including the Levenshtein distance, the number of reconstructed haplotypes and their length. ShoRAH has an additional column that specifies the number of analyses we ran with different parameters followed by the number of analyses that did not complete in brackets (i.e. where ShoRAH aborted computations at some point during the analysis). We ran PredictHaplo with a value of 2,000, 3,000 and 4,000 for the number of reads that are considered over a local window and used an entropy threshold of 0.005. PredictHaplo completed all of the analyses. For QuRe we used the default settings for the homopolymeric and non-homopolymeric 454 error rates (0.0044 and 0.0007, respectively) and the default number of iterations (10,000 iterations). An increase to 30,000 and 50,000 iterations resulted in much longer running times but no improvement. The results for the similarity measure are visualised in Figure 5.6 and Figure 5.7.

Figure 5.6 includes the SiM_i results (i=0, i=1 and i=2) on the 454 reads for PredictHaplo (purple), QuRe (green) and ShoRAH (blue) and the results for PredictHaplo on Illumina reads (red) for each data set. The uniformly distributed data sets (e.g. DSIa) are plotted next to their corresponding log-normal distributed data sets (e.g. DSIb). Overall, PredictHaplo showed the best performance and was able to increasingly reconstruct the population as sequence divergence increased.

In case of the low sequence divergence in DSIa and DSIb, none of the programs were able to reconstruct any of the haplotypes with up to two mismatches and the majority of the returned haplotypes show a Levenshtein distance that exceeds the mutual sequence divergence in the population. For the uniformly distributed data set DSIc (with 50 mutations on each haplotype) PredictHaplo reconstructed one haplotype with one mismatch and for the corresponding log-normal distributed data set DSId three haplotypes were reconstructed exactly and one haplotype with two mismatches (see

Table 5.5: **454 reads:** Overview of the performance of the haplotype reconstruction programs across all test data sets. The first two columns specify the data set and the mutual sequence divergences of the haplotypes. For each program we specify the results of the best run in three columns. This includes the range of Levenshtein distances between each of the reconstructed haplotypes and the corresponding closest "true" haplotype, the number of reconstructed haplotypes in the population and their lengths. ShoRAH has an additional column where we specify the number of runs (with different parameters) and in brackets the number of runs that did not complete.

		ShoRAH				PredictHaplo			QuRe		
DS I	divergence	#Ana(failed)	best Ldist	# rec_hap	length	best Ldist	# rec_hap	length	best Ldist	# rec_hap	length
a.)	19bp	14 (5)	7-172	328	8040	9-267	8	8162	7-382	110	7590
b.)	19bp	10 (4)	17-160	295	8040	16-374	6	8162	17-216	256	7595
c.)	91bp	10 (4)	24-161	259	8040	1, 3-25	10	8162	25-326	114	7582
d.)	91bp	10 (4)	21-98	551	8107	(3x)0, 2-34	10	8162	7-96	210	7589
e.)	325bp	10 (4)	1-209	300	8040	(3x)0, 1-2	10	8162	x	x	x
f.)	325bp	10 (3)	3-231	252	8040	(6x)0, 1-3	10	8162	x	x	x
DS II											
a.)	2-328bp	14 (0)	(2x)1, 2-82	586	2546	0, 8-9	5	2580	1, 4-173	84	2280
b.)	2-328bp	13 (1)	0, 26-302	41	2541	0, 14-422	19	2580	0, 8-989 *	272	2254
DS III											
a.)	1-41bp	11 (0)	0-672 *	53	8085	1, 195-1591	7	8162	x	x	x

[x] analysis did not complete

* the reference sequence was reconstructed

Table 5.5). However, the reconstructed population also contains haplotypes with a Levenshtein distance of up to 34 where the mutual distance between the "true" haplotypes is on average 91bp. None of the other programs were able to reconstruct any haplotypes for these populations. For the much more diverse data sets DSIe and DSIf ($\approx 4\%$), PredictHaplo achieved good results: three haplotypes were exactly reconstructed for the uniformly distributed data set DSIe and six haplotypes where exactly reconstructed for the log-normal distributed data set DSIf. The rest of the reconstructed haplotypes have $\leq$3 mismatches. ShoRAH was able to reconstructed one haplotype with one mismatch and seven haplotypes with two mismatches, but the population also contains haplotypes with up to 209 mismatches (compared to their closest true haplotype). QuRe aborted the calculations for DSIe and DSIf with an error message.

For data sets DSIIa and DSIIb we concentrated on the reconstruction of a single gene. PredictHaplo was able to reconstruct one haplotype exactly for each of the two data sets. But in the case of DSIIb we also find haplotypes with up to 422 mismatches in the reconstructed population. ShoRAH reconstructed two haplotypes with one mismatch and one haplotype with two mismatches. However, the reconstructed population contains a total of 586 haplotypes compared to 44 "true" haplotypes. In the case of DSIIb ShoRAH reconstructed one of the haplotypes exactly. QuRe achieved its best result across all data sets for DSIIb where the dominant haplotype was reconstructed exactly. But the reconstructed population contains also haplotypes with up to 989 mismatches. For DSIIa QuRe reconstructed one of the haplotypes with one mismatch.

DSIII was the most complex *in silico* population with a large number of haplotypes. ShoRAH was able to reconstruct the reference haplotype exactly but the rest of the population shows between 208 and 672 mismatches compared to their closest true haplotype. PredictHaplo found one "true" haplotype with one mismatch and the Levenshtein distance for the rest of the population is between 195 and 1,591. PredictHaplo seems to have problems in the presence of many low abundant haplotypes and seems to incorporate SNPs from many different haplotypes into very few reconstructed haplotypes. QuRe aborted the calculations for this data set with an error message.

The similarity measure for the Illumina read data sets

In the documentation of ShoRAH the usage of Illumina reads is only described for the reconstruction over a local window. We tested ShoRAH on Illumina reads as a matter of completeness on data sets where ShoRAH has been able to reconstruct haplotypes with 454 reads. We did two runs with different parameters for each of those data sets. ShoRAH only produced results for one run on DSIIa where 102 haplotypes were reconstructed with 424-467 mismatches and one run for DSIIb where 987 haplotypes were reconstructed with 588-681 mismatches. We ran QuRe on all data sets with Illumina

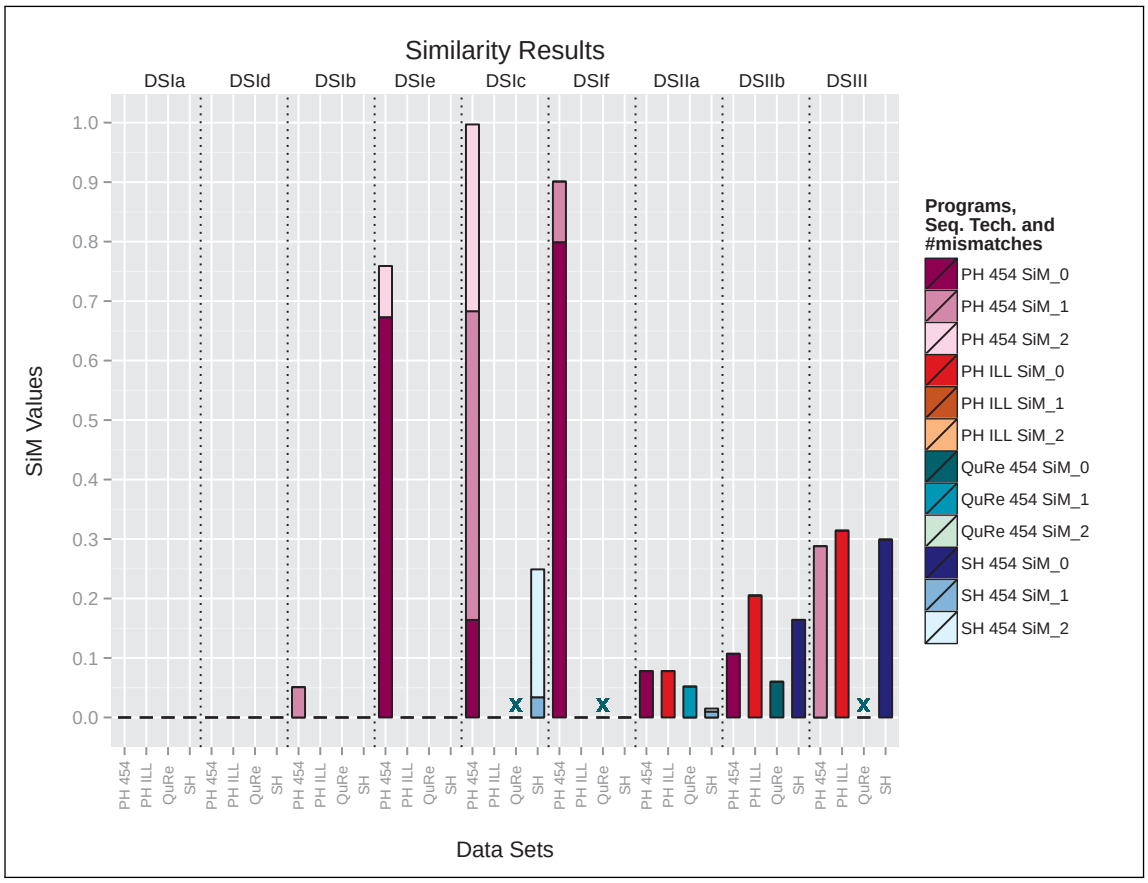


Figure 5.6: Similarity results for the haplotype reconstruction programs: The test data sets are indicated on the upper x-axis. For each data set we have four bars and the respective programs and sequencing technologies are indicated on the lower x-axis. The bars correspond to the result for PredictHaplo on 454 reads, PredictHaplo on Illumina reads, QuRe on 454 reads and ShoRAH on 454 reads. The programs are indicated with different colours and the shade of the colour signals how much of the population was reconstructed with zero, one or two mismatches in the sequence. The higher the value the better the reconstruction, where a perfect reconstruction corresponds to a value of one.

specific error rates of 0.0012 [31]. Unlike 454, Illumina has the same error probability in homopolymeric and non-homopolymeric regions. QuRe aborted computations with an error message for all Illumina data sets. We ran PredictHaplo three times for all of the data sets with the same parameters as for the 454 data sets. PredictHaplo returned only one or two haplotypes in each case. For DSIa - DSIf none of the "true" haplotypes were found and the reconstructed haplotype(s) showed an increasing number of mismatches compared to their closest true haplotype as the sequence divergence in the data sets increased. It seems that PredictHaplo recognised the SNPs occurring on various haplotypes but was not able to assign them to different reconstructed haplotypes. As Illumina reads are very short compared to 454 reads, there seems to be insufficient information to link the SNPs to their respective haplotype. In the case of DSIIa and DSIIb one haplotypes was successfully reconstructed. Only one haplotype was returned for DSIIa thus PredictHaplo assessed all of the SNPs occurring on other haplotypes as

Table 5.6: **Illumina reads:** The table summarises the results for PredictHaplo and ShoRAH for the Illumina read data sets. Analogously to Table 5.5 the Levenshtein distance, the number of reconstructed haplotypes and the length of the reconstructed haplotypes is displayed.

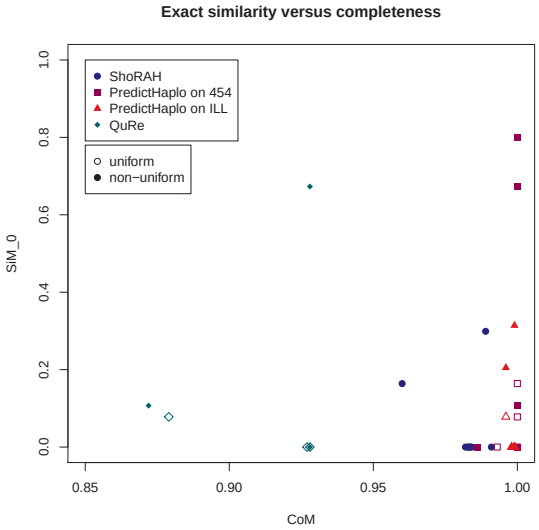
		PredictHaplo			ShoRAH		
DS I	divergence	best Ldist	# rec_hap	length	best Ldist	# rec_hap	length
a.)	19bp	9,134	2	8152	-	-	-
b.)	19bp	8	1	8152	-	-	-
c.)	91bp	45	1	8151	-	-	-
d.)	91bp	45	1	8152	-	-	-
e.)	325bp	185	1	8149	-	-	-
f.)	325bp	184	1	8147	-	-	-
DS II							
a.)	2-328bp	0	1	2570	424-467	102	2540
b.)	2-328bp	0, 57	2	2569	588-681	987	2580
DS III							
a.)	1-41bp	0 *	1	8152	x	x	x

* the reference sequence was reconstructed
- did not run any analyses as the reconstruction was not successful with the much longer 454 reads
x analysis did not complete

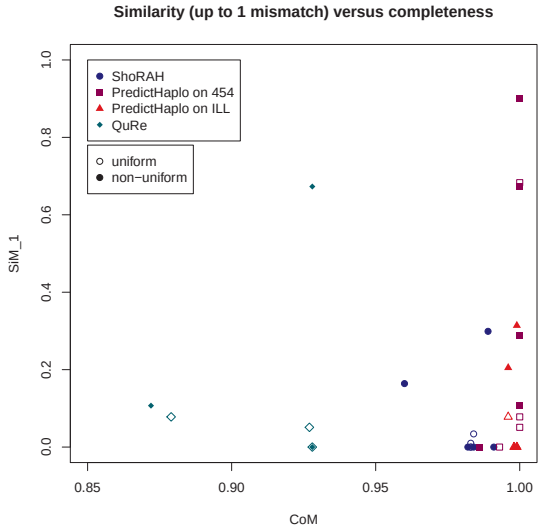
errors. In the case of DSIII the wildtype was successfully reconstructed. The results for PredictHaplo and ShoRAH are summarised in Table 5.6 and the PredictHaplo results are included in Figure 5.6.

Completeness of the reconstruction for the 454 and Illumina read data sets

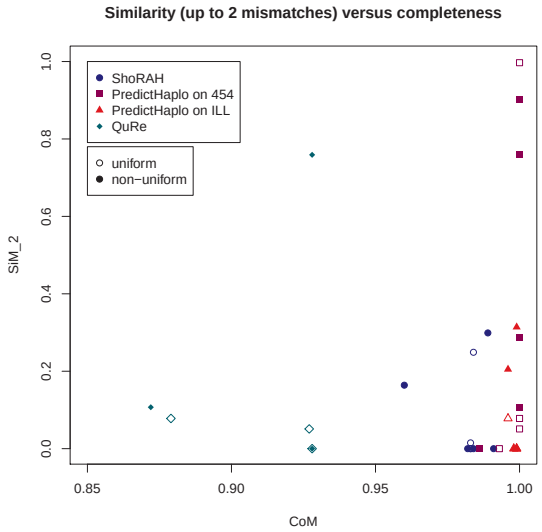
In Figure 5.7 we plotted the similarity versus completeness measure. The completeness measure reflects how much of the haplotype was reconstructed on average (in percent). For those parts of the haplotypes that were reconstructed the similarity reveals the quality of the reconstruction; a value of zero means that none of the "true" haplotypes was reconstructed and a value of one corresponds to perfect reconstruction. Across all of the data sets the reconstructed haplotypes covered between 87% and 100% of the length of the "true" haplotypes. PredictHaplo produced the best results in terms of completeness with values of $\geq$99%. ShoRAH produced very good results as well with values above 98% for all data sets besides DSIIb (96%). The QuRe results were $\approx$93% for DSIa, DSIb, DSIc and DSId and $\approx$87% for DSIIa and DSIIb.



(a) SiM_0 versus CoM



(b) SiM_1 versus CoM



(c) SiM_2 versus CoM

Figure 5.7: Similarity versus completeness for ShoRAH, PredictHaplo and QuRe: The similarity results with zero, one and two mismatches are displayed on the y-axis of the respective graph. The completeness (which is independent of the number of mismatches) is displayed on the x-axis. All completeness values were between 85 and 100%.

5.7 Limitations of the Read-Graph Approach for Haplotype Reconstruction

Eriksson et al. presented an algorithm in 2008 [60] using a mathematical and statistical approach for analysing the diversity of viral populations. Their graph approach is the basis for the program ShoRAH which has been widely used in the past. Newer programs have became available recently that are based on probabilistic clustering al-

gorithms. These programs returned significantly better results compared to ShoRAH. In the following we will discuss the possible underlying causes for limitations associated with the read-graph approach.

At the time that Eriksson et al. presented their algorithm pyrosequencing was still a novel sequencing technique and not yet well established. Pyrosequencing produces much shorter reads compared to traditional Sanger sequencing and the reads are more error-prone. But high-throughput, cost-effectiveness and speed constitute major advantages. Their algorithm starts with an error correction procedure for the obtained reads. The starting error rate is approximately 5-10 errors per kb. In the next step all possible haplotypes are inferred by overlapping the reads. A minimum subset of those haplotypes is then identified that explains the observed reads. The process is completed by calculating a frequency estimate for the inferred haplotypes to give an overall estimate of the population structure of the sample. Their method "works best for populations that are suitably diverse" [60]. The lower the diversity the harder it is to link two reads together on the same haplotype since there are less identifying features. Though the benefit of higher diversity is partially reduced at some point by the increasing difficulty of the alignment problem.

When sequencing a virus population we obtain many reads. The difficulty is to determine the haplotype that gives rise to a specific read. The algorithm detects a set of haplotypes that *explains* the observed set of reads. However, this set of haplotypes is not unique, though the cardinality of this set is a lower bound for the number of haplotypes present in the sample.

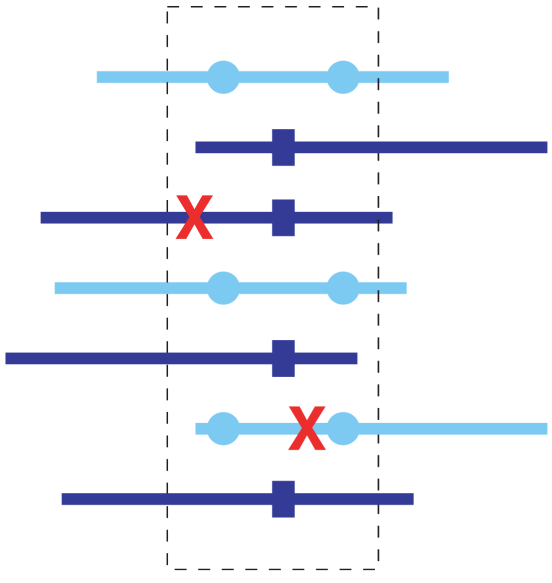


Figure 5.8: Error correction over sliding windows: errors are corrected by comparing mutations occurring in the same aligned region over a small window (indicated by the dashed line). (This figure is taken from [60].)

The input for their algorithm is a set of reads and a reference genome and the first step is aligning the reads to the reference genome. Then the error correction process starts with discarding $\approx 10\%$ of the reads that show an atypical length or contain ambiguous bases. Assuming that there are no true insertions, one can correct all occurring insertions by comparison with the reference genome.

Next, a multiple alignment is computed and the reads are compared in small "windows" as shown in Figure 5.8, where the current

window is indicated by the dashed line. The significance of a single mutation is evaluated with a binomial test as well as the occurrence of two mutations simultaneously. Taking this into account the reads are clustered into groups. The mutations are marked as circles and squares for the two groups, respectively (the groups are indicated by the different colours in Figure 5.8). The crosses mark changes that are not significant and thus detected as errors and corrected by the algorithm according to the consensus within the group. This corrects substitutions and deletions. Note that the procedure can mistake substitutions on extremely rare haplotypes (occurring with a frequency of less then 1%) for sequencing errors and eliminates them.

The next step is the haplotype reconstruction. The goal is to find a *minimum set of haplotypes* that accounts for as many of the observed (and corrected) reads as possible. This is based on the assumption that every haplotype can be reconstructed by a subset of overlapping reads. To find a minimum set of haplotypes a read-graph is constructed:

Let R be the set of aligned reads after the above described error correction procedure. Every read consists of a start position in the reference genome and its base pair sequence. R_{irred} denotes the set of *irredundant* reads, where a read is said to be *redundant* if there is another read that completely overlaps it. Let s be the source and t be the sink. Then the corresponding read graph G_R is an acyclic directed graph with vertices $\{R_{irred}, s, t\}$ and includes an edge from $r_1 \in R_{irred}$ to $r_2 \in R_{irred}$ if all of the following conditions are satisfied:

- r_1 and r_2 overlap and have the same base pair sequence in this overlapping region ($\Rightarrow$ the two reads can be combined into a contig)
- the starting position of r_1 is before the starting position of r_2 ($\Rightarrow$ implies the direction of the edge)
- without this edge there is no path in the graph from r_1 to r_2 ($\Rightarrow$ avoids cycles)

In addition we add edges from s to all reads with starting position 1 as well as edges from all reads ending at the final base of the reference genome to t. Every path from s to t corresponds to a possible haplotype. Finding a minimum set of haplotypes that accounts for all observed reads is equivalent to finding a minimal cover of the corresponding read graph or solving a maximum matching problem in the associated bipartite graph which can be done in at least $\mathcal{O}(|R_{irred}|^3)$. So the problem is in the worst case cubic in the number of irredundant reads.

Figure 5.9 clarifies the construction of the read-graph for a simple example of 20 reads of a sequence of length 8 over a binary alphabet $\{0, 1\}$. Each path from s to t is a possible haplotype that could give rise to a subset of the observed reads. For example the bottom path corresponds to the haplotype 00001111. A minimal cover of the graph corresponds

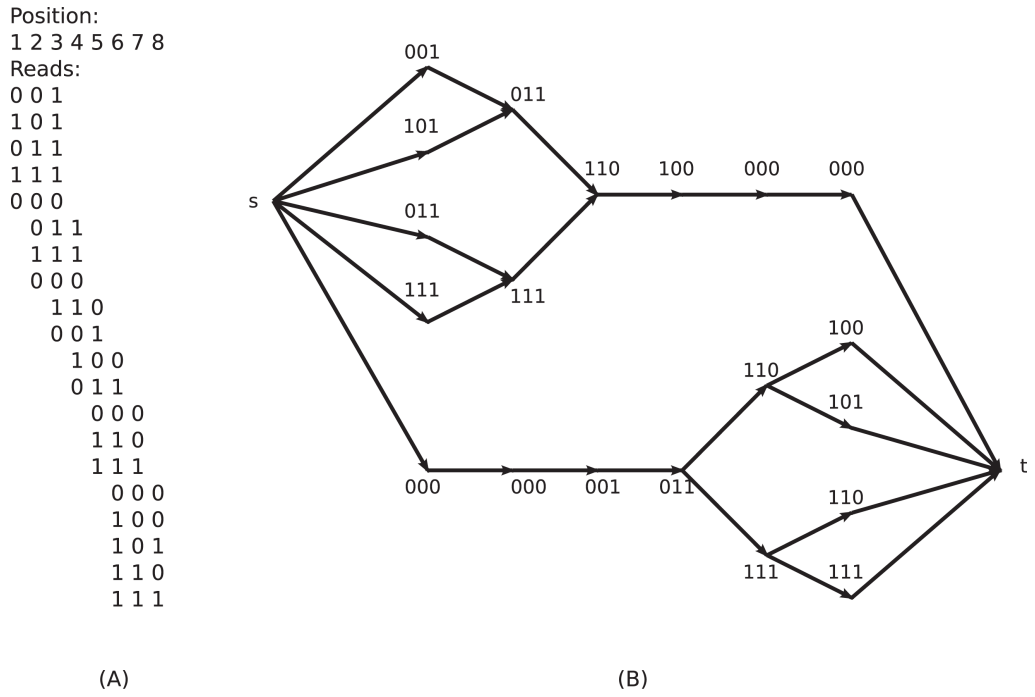


Figure 5.9: The read-graph (B) corresponding to the 20 reads shown on the left side (A) of a sequence of length 8 over a binary alphabet $\{0, 1\}$. (This figure is taken from [60].)

to a minimum set of haplotypes accounting for all observed reads. The minimal cover and thus the minimum set of haplotypes is in general not unique. But the cardinality of all minimal covers is the same and so is the number of haplotypes forming a minimum set. Thus we obtain a lower bound for the number of haplotypes present in the sample. Analogously, the number of possible paths could be regarded as an upper bound for the number of haplotypes. However, this number is in general very large.

Next, we address the question of how many reads are necessary to cover all bases of a genome of length n. Lander & Waterman [89] assume a uniform distribution of the reads. In practice though we often observe an uneven coverage. Reads are more likely to start at certain points than others depending on, e.g. GC content of the genome. In [60] a Poisson distribution is assumed for the probability that all bases of the genome are sequenced:

$$p = (1 - e^{-c})^n$$

Here c denotes the coverage which is the product of the total number of reads N with read length L divided by the genome length n:

$$c = \frac{NL}{n}$$

So the probability p, that all haplotypes occurring with a frequency of at least ρ are completely covered, is

$$p \geq (1 - e^{-c \cdot \rho})^n$$

Thus $p \geq (1 - e^{-\frac{NL\rho}{n}})^n \quad \Leftrightarrow \quad 1 - p^{\frac{1}{n}} \leq e^{-\frac{NL\rho}{n}} \quad \Leftrightarrow \quad -ln(1 - p^{\frac{1}{n}}) \geq \frac{NL\rho}{n} \quad \Leftrightarrow$

$$N \leq -\frac{n \cdot ln(1 - p^{\frac{1}{n}})}{L\rho} \tag{7}$$

This provides an upper bound for the number of reads required to cover all haplotypes occurring with a frequency of at least ρ. (Note, that there is a mistake in Equation (1) in [60].)

Suppose, we have haplotypes of length $8,000$bp. To cover all haplotypes occurring with a frequency of at least 3% with reads of 100bp with a probability of 99% we need at least:

$$N \leq -\frac{n \cdot ln(1 - p^{\frac{1}{n}})}{L\rho} = -\frac{8000 \cdot ln(1 - 0.99^{\frac{1}{8000}})}{100 \cdot 0.03} \approx 36,233$$

Therefore, at most $36,233$ reads are necessary to cover all haplotypes that occur with a frequency of at least 3%.

The last step of the haplotype reconstruction algorithms described by Eriksson et al. deals with the frequency estimation of the haplotypes. A virus population is regarded as a probability distribution on a set of haplotypes and the goal is to estimate this distribution given an observed set of reads. Denote with R the set of reads consistent with a minimal set of haplotypes H. The vector $u \in N^R$ describes the read data set where u_r gives the number of times a read r has been observed. The probability distribution of $h \in H$ is denoted by $p = (p_h)_{h \in H}$ and we assume a uniform distribution for the reads $r \in R$. So the probability of observing r is

$$P(R = r) = \sum_{h \in H} p_h \; P(R = r | H = h)$$

where $P(R = r|H = h) = \frac{1}{K}$ if h is consistent with r and 0 otherwise. With K we denote the number of reads consistent with a particular haplotype h. By maximising the log-likelihood function

$$l(p_1, \ldots, p_{|H|}) = \sum_{r \in R} u_r \; log(P(R = r))$$

we can obtain an estimate for the probability distribution p of the haplotypes using an EM algorithm.

The authors report that the reconstruction of the exact haplotypes for populations with low diversity is very difficult, but that the haplotypes that were found are close to the true ones. They were able to correctly reconstruct haplotypes with low frequencies ($\approx$3%) by repeatedly computing a minimum set of explaining haplotypes. They report

that their algorithm works well on "error-free reads that are diverse enough" and when "errors are introduced, performance decreases" but "the method still recovers much of the original population" [60]. One problem when dealing with a larger number of reads is that the number of paths in the graph increases as well and thus it is more likely that "haplotypes" containing errors are assigned positive probabilities.

The limit of irredundant reads that the algorithm can work with (as of May 2008) is approximately 13,000. This is probably due to the intense computations necessary for the construction of the read-graph. They also mention that in the case of low diversity the actual number of reads needed to ensure the complete coverage of all haplotypes can be much lower than the upper bound given in Equation (7). Also, instead of inferring a minimum cover of the read-graph, a maximum likelihood approach could be used to determine the most likely set of haplotypes explaining the reads.

The read-graph approach forms the basis for many of the currently available viral haplotype reconstruction programs. A critical factor for the read-graph approach is the read length as the reconstruction relies on the overlap of the reads. Next, we highlight the limitations of the read-graph approached based on the previously described FMV data set.

For simplicity, we assume that all reads are of the same length. Furthermore, we assume that mutations are independent and the mutation rate is 5.6×10^{-4} as computed in Equation (5). The probability that more than two mutations are captured by a read of length r is as follows:

We assume that the underlying distribution is binomial. The probability that a base is correct at a single position in the genome is $q = 1-(5.6419\times10^{-4}) \approx 0.99943581$ and the probability for a mutation is $p = 5.6419 \times 10^{-4}$. This yields the following probabilities:

$$P(0 \text{ mutations}) = q^r \tag{8}$$

$$P(1 \text{ mutations}) = \binom{r}{1} \cdot p^1 \cdot q^{r-1} = r \cdot p^1 \cdot q^{r-1} \tag{9}$$

$$P(2 \text{ mutations}) = \binom{r}{2} \cdot p^2 \cdot q^{r-2} = \frac{r \cdot (r-1)}{2} \cdot p^2 \cdot q^{r-2} \tag{10}$$

So suppose each read is 500bp long, then the probability of two mutations being captured on one read is:

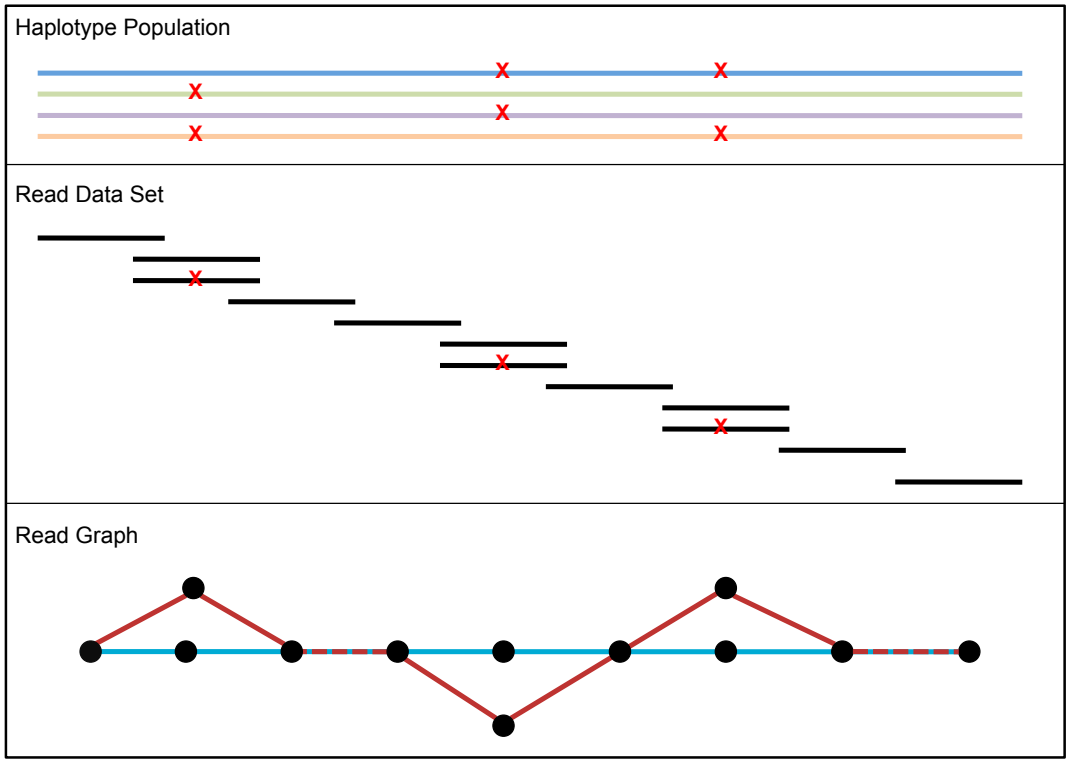


Figure 5.10: Limitations of the read-graph approach for haplotype reconstruction: The haplotype population is displayed in the upper plot. The population consists of four haplotypes and the SNPs on each haplotypes are marked with a red "x". The middle plot shows the unique reads of a NGS data set. These reads are used to construct the read-graph which is displayed in the lower plot. Each node of the read-graph corresponds to one of the unique reads. The minimal cover for this read-graph consists of two paths (marked in blue and red) and underrepresents the number of haplotypes in the population.

$$
\begin{aligned}
P(>2 \text{ mutations}) &= 1 - P(\leq 2 \text{ mutations}) && (11)\\
&\approx 1 - (0.754142 + 0.212860 + 0.029980) && (12)\\
&\approx 0.003018 && (13)
\end{aligned}
$$

Therefore, on average only about 0.3% of the reads cover ≥ 2 mutations. Thus, the large majority of reads will either lie on the "main path" of the read-graph that represents the reference sequence (if they do not cover a mutation) or they will add a single node that is directly connected to the main path by two edges. As a consequence a lot of base changes that do not necessarily occur on the same haplotypes can be explained by a single haplotype and only a small number of paths will be necessary to obtain a minimal cover of the corresponding read-graph (see Figure 5.10). Thus the lower bound will tend to greatly underestimate the real number of haplotypes present in the population.

5.8 Conclusion and Future Work

The programs that we tested were unable to cope with populations with low sequence divergences and low abundance levels. There are two problems: Firstly, sequencing errors can be mistaken for sequence divergence. Secondly, if SNPs occur with a distance exceeding the read length we cannot infer if these SNPs occurred on the same haplotype based on the reads. The reconstructed populations contained many false positive SNPs and false positive haplotypes and many true haplotypes were not recovered. This has serious consequences, for example in the development of drug therapies and vaccine design as the haplotypes that were not recovered cannot be targeted by the treatment. Haplotypes in a viral population can be resistant to a drug treatment due to mutations and, even if they occur at low frequencies, their presence has been shown to correlate with treatment failure [147][93]. Thus it is particularly important that reconstruction programs are also able to recover low frequency haplotypes in the population. We also need to bear in mind that for "real" data sets (in contrast to mock communities) we are not able to distinguish successfully reconstructed haplotypes from false positives. In many of the reconstructed populations the number of false positives far exceeds the number of successfully reconstructed haplotypes and false positives often showed a substantial number of mismatches compared to the "true" haplotypes. In addition, a high number of false positives needlessly complicates the development of effective treatments.

In general, the read-graph approach seems computationally expensive and not the optimal approach for the whole gene/genome haplotype reconstruction; the inferred set of haplotypes was, in most cases, much larger than the actual "true" population. The approach with a Dirichlet process mixture model, where errors are not corrected but incorporated as prior information, is computationally less expensive and seems to be more powerful. ShoRAH has been previously tested on single gene reconstruction rather than whole genome reconstruction. On our data sets ShoRAH achieved better results on data set DSIIb where the reconstruction concentrates on a single gene and haplotypes are distributed according to a log-normal distribution. For the 454 reads ShoRAH outperformed PredictHaplo on this data set. However, the DPMM approach of PredictHaplo seems to scale better to whole genome reconstruction than the read-graph approach. Also, as the read-graph approach reconstructs the haplotypes based on the overlap of the reads, the much shorter Illumina reads are not suitable for ShoRAH and QuRe.

None of the currently available programs for whole gene/genome reconstruction are designed to detect recombination. The program QuasiRecomb is accounting for recombination events but the currently published version only attempts local reconstruction. We created an *in silico* data set of ten haplotypes including two recombinants, to test the ability of PredictHaplo, ShoRAH and QuRe to detect recombination events. The

data set consists of the first eight haplotypes from data set DSIe ($\approx 4\%$ sequence divergence). We created two recombinants with breaking point at position 2,685 and 4,450 respectively and added them to the population. The ten haplotypes were mixed according to a uniform distribution and we simulated 120,000 454 reads with FlowSim. QuRe did not yield any results on this data set. For PredictHaplo we ran three analyses (with parameters 2,000, 3,000 and 4,000). On the best run PredictHaplo returned all eight non-recombinants with 0-2 mismatches, but was not able to identify any of the recombinants. For ShoRAH we used the parameters of its three best runs for data set DSIe. The best reconstructed population consisted of 271 haplotypes with a Levenshtein distance between 1 and 157. The closest reconstruct haplotype to the first recombinant showed 58 mismatches and the closest haplotype to the second recombinant showed 4 mismatches. This shows that the approach that ShoRAH takes is in principal able to identify recombination events but the high number of mismatches and false positives remains problematic.

All of the haplotype reconstruction programs have been tested previously but mostly on data sets with a much larger sequence divergence. The highest sequence divergence in our data sets is $\approx 4\%$. For bacteria the threshold to distinguish between different species is 3%. Though there is no similar method for comparing viruses (as there is no gene or region that all viruses have in common) we expect that a large fraction of the haplotypes in a viral population show very low sequence divergence. Thus programs need to be able to find haplotypes that only differ by a few nucleotides.

Further advances in sequencing technologies are just a matter of time. This will be accompanied by reduced error rates, increased read length and higher coverage. Those advances will greatly improve our ability to reconstruct viral haplotype population from a set of observed reads. However, reducing the rate of false positives in a reconstructed population remains an imperative if we are to obtain reliable results from the reconstruction that will facilitate the development of effective vaccines and treatments.

Future work will also involve the design of an *in vivo* viral haplotype community. This will allow us to test the impact of other biases (e.g. uneven coverage of the genome and environmental factors such as library preparation method, primer design and number of PCR cycles).

6 Error Profiles for Amplicon Sequencing Data Sets for the Illumina MiSeq Platform

6.1 Abstract

With read lengths of currently up to 2x300bp, high throughput and low sequencing costs Illumina's MiSeq is becoming one of the most utilised sequencing platforms worldwide. The platform is manageable and affordable even for smaller labs. This enables quick turnaround on a broad range of applications such as targeted gene sequencing, metagenomics, small genome sequencing and clinical molecular diagnostics. However, current knowledge on systematic errors in Illumina data is insufficient and error correction programs are not designed to address systematic errors in Illumina data. The identification and removal of these errors is essential for sequence analysis and vital if we are to draw valid conclusions. Studying true genetic variation in a population sample is fundamental for understanding diseases, evolution and origin and mistaking sequencing errors for diversity can have disastrous effects on any conclusions. We conducted a large study on the error patterns for the MiSeq based on complex *in vitro* mock communities. We tested state-of-the-art library preparation methods for amplicon sequencing and showed that the library preparation method and the choice of primers are the most significant sources of bias and cause distinct error patterns. Furthermore we tested the efficiency of various error correction strategies and identified quality trimming (sickle) combined with error correction (BayesHammer) followed by read overlapping (PANDAseq) as the most successful approach, reducing substitution error rates on average by 93%.

This work was also presented in the following publications:

> Melanie Schirmer, Umer Z. Ijaz, Linda D'Amore, Neil Hall, William T. Sloan, and Christopher Quince. **Insight into Biases and Sequencing Errors for Amplicon Sequencing with the Illumina MiSeq Platform.** (In review: Nucleic Acids Research)
>
> Linda D'Amore, Umer Z. Ijaz, Melanie Schirmer, Neil Hall, and Christopher Quince. **A comprehensive benchmarking study of next-generation sequencing platforms for 16S rRNA community profiling.** (In preparation: Genome Biology)

Original Contributions

To my knowledge, this is the first study on error profiles of the MiSeq platform and the most comprehensive study of Illumina error profiles up-to-date. This is also the first time that amplicon data sets were considered and a context of experimental design and

error patters was established. I identified distinct motifs for all three types of errors and showed that the motifs vary depending on the library preparation method and the primers. Additionally, I present the first independent comparison of currently available error removal strategies for Illumina data sets.

My contributions involved the design of the unbalanced mock community as well as extensive work on the reference databases which formed the basis for the subsequent analysis. The experimental work including the library preparation and sequencing was done in Liverpool. I conducted the entire study of the error profiles presented in this chapter including the development and implementation of the algorithms as well as the analysis of all data sets.

6.2 Introduction

The announcement by Roche to withdraw its GS FLX 454 pyrosequencing platform emphasises the need for a better understanding of Illumina errors. 454 and Illumina sequencing errors are fundamentally different and require different strategies with regards to the downstream analysis. The majority of errors in 454 data are related to homopolymers [76][31]. For Illumina, on the other hand, substitution type miscalls are the dominant source of errors. Illumina's sequencing technology is based on array formation. The sequencing templates are immobilised on a flow cell and a subsequent solid-phase bridge amplification generates up to 1,000 copies in close proximity (cluster generation). The sequencing-by-synthesis (SBS) technology uses fluorescently labeled reversible terminator-bound dNTPs (A,C,G,T) for the polymerisation. Only one base is added in each step due to the 3' termination of the incorporated nucleotide. The fluorophores are illuminated by a red laser for A and C and a green laser for G and T and imaged through different filters to identify the four different nucleotides. The fluorescent labels and the 3' terminators are then removed in order for the next cycle to commence. Challenges arise due to a strong correlation of A and C as well as G and T intensities as a result of similar emission spectra of the fluorophores and limitations of the filters. Furthermore, problems known as phasing and pre-phasing can cause noise in the cluster signal. Phasing can occur due to problems with the enzyme kinetics such as incomplete removal of the 3' terminators or the fluorophores which causes the synthesis of some molecules in a cluster to lag behind. During pre-phasing, on the other hand, the synthesis advances too fast which can be caused by inadequate flushing of the flowcell, by sequences in a cluster skipping an incorporation cycle or the incorporation of nucleotides without an effective 3' terminator. The number of affected sequences increases with each cycle and thus limits the read length. Overall, substitution type miscalls are the major source of errors for Illumina sequencing [84].

Most previous studies on Illumina specific errors have concentrated on the Genome Analyzer (GAII) and the HiSeq 2000 [157][111][110]. Significant improvements in the technology and software have generally improved error rates but we still face systematic errors in Illumina sequencing. Nakamura et al. [113] identified two sequence patterns in Illumina GAII data that trigger errors during the sequencing process - firstly, inverted repeats and secondly, GGC sequences. They suspect that the first pattern causes dephasing by inhibiting single-base elongation through folding of the single-stranded DNA and that the second pattern causes altered enzyme preference on the lagging-strand. They also report that mismatches are mainly observed in reads sequenced in the same direction and there is a strong correlation between average base call quality and mismatch rate. Their study also showed that the GGC pattern does not always trigger a sequencing error. The pattern "may occur once every 64 bases by chance" [113] but sequence specific errors (SSE) are less common. Furthermore a significant number of SSE positions was not associated with the identified sequence patterns suggesting that other factors may be a significant cause for sequencing errors.

For our experiments we used a variety of single species samples as well as a complex mock community consisting of 59 organisms. We developed a program that enables us to infer error profiles based on sequencing data from mock communities. Our software can identify mismatches and indels for any sequenced mock data set. This allowed us to study and compare the impact of: library preparation methods, run, input DNA amount, number of PCR cycles, Taq, DNA template and forward and reverse primer combinations. We provide an in-depth analysis of the errors occurring on both read directions for all types of errors and tested the reliability of quality scores. It has been reported previously that the per-base quality scores can be inaccurate and co-variation has been observed with attributes like sequencing technology, machine cycle and sequence context [55]. We will show that the accuracy of the quality scores varies depending on which library preparation method was used. The differentiation of true variation and context-specific sequencing errors is a major challenge in NGS analysis. Being able to infer error profiles for individual sequencing runs has the potential to greatly improve our ability to correct errors and thus enhance further sequencing analysis.

6.3 Materials and Methods

Mock Community & Sequencing Data

We sequenced a variety of samples ranging from single species to diverse mock communities with different abundance distributions. The single organisms included *Anaerocellum thermophilum Z-1320 DSM6725* (AT), *Bacteroides thetaiotaomicron VPI-5482* (BT), *Bacteroides vulgatus ATCC 8482* (BV), *Herpetosiphon aurantiacus ATCC 23779*

Table 6.1: Overview of organisms in the mock community.

Bacteria	
Acidobacterium capsulatum ATCC 51196	*Ruegeria pomeroyi DSS-3*
Akkermansia muciniphila ATCC BAA-835	*Salinispora arenicola CNS-205*
Anaerocellum thermophilum Z-1320, DSM 6725	*Salinispora tropica CNB-440*
Bacteroides thetaiotaomicron VPI-5482	*Shewanella baltica OS185*
Bacteroides vulgatus ATCC 8482	*Shewanella baltica OS223*
Bordetella bronchiseptica RB50	*Sulfitobacter sp. EE-36*
Burkholderia xenovorans LB400	*Sulfitobacter sp. NAS-14.1*
Caldicellulosiruptor saccharolyticus DSM 8903	*Sulfurihydrogenibium sp. YO3AOP1*
Chlorobaculum tepidum TLS	*Sulfurihydrogenibium yellowstonense SS-5*
Chlorobium limicola DSM 245	*Thermoanaerobacter pseudethanolicus ATCC 33223*
Chlorobium phaeobacteroides DSM 266	*Thermotoga neapolitana DSM 4359*
Chlorobium phaeovibrioides DSM 265	*Thermotoga petrophila RKU-1*
Chloroflexus aurantiacus J-10-fl	*Thermotoga sp. RQ2*
Clostridium thermocellum ATCC 27405	*Thermus thermophilus HB8*
Deinococcus radiodurans R1	*Treponema denticola ATCC 35405*
Desulfovibrio desulfuricans desulfuricans ATCC 27774	*Treponema vincentii I*
Desulfovibrio piger ATCC 29098	*Zymomonas mobilis mobilis ZM4*
Dictyoglomus turgidum DSM 6724	
Erwinia chrysanthemi	
Enterococcus faecalis V583	
Fusobacterium nucleatum nucleatum ATCC 25586	
Gemmatimonas aurantiaca T-27T	**Archaea**
Herpetosiphon aurantiacus ATCC 23779	*Archaeoglobus fulgidus DSM 4304*
Hydrogenobaculum sp. Y04AAS1	*Ignicoccus hospitalis KIN4/I*
Leptothrix cholodnii SP-6	*Methanocaldococcus jannaschii DSM 2661*
Nitrosomonas europaea ATCC 19718	*Methanococcus maripaludis C5*
Nostoc sp. PCC 7120	*Methanococcus maripaludis S2*
Pelodictyon phaeoclathratiforme BU-1	*Nanoarchaeum equitans Kin4-M*
Persephonella marina EX-H1	*Pyrobaculum aerophilum IM2*
Porphyromonas gingivalis ATCC 33277	*Pyrobaculum calidifontis JCM 11548*
Rhodopirellula baltica SH 1	*Pyrococcus horikoshii OT3*
Rhodospirillum rubrum ATCC 11170	*Sulfolobus tokodaii 7(S311)*

(HA), *Rhodopirellula baltica SH 1* (RBS), *Leptothrix cholodnii SP-6* (LC) and *Caldicellulosiruptor saccharolyticus DSM 8903* (CS). For the first mock community we combined even amounts of purified genomic DNA [145] from 49 bacteria and 10 archaea (see Table 6.1 for details). We used the same genomes to construct an uneven mock community with a complex community structure. The bacteria were split into 13 groups where the number of organisms in each group ranged from one to eight. Each group was assigned a weight drawn from a log-normal distribution and the organisms within each group follow again a log-normal distribution. Analogously, the archaea were split into three groups according to their respective phylum. Each group was assigned a weight drawn from a log-normal distribution and the abundance levels within each group were in turn modelled according to a log-normal distribution. Overall the bacteria make up 90% of the uneven mock community and the archaea make up 10%.

We sequenced the V4 and the V3/V4 region of the samples and also included two samples where the whole 16S gene was sequenced. Five different library preparation methods

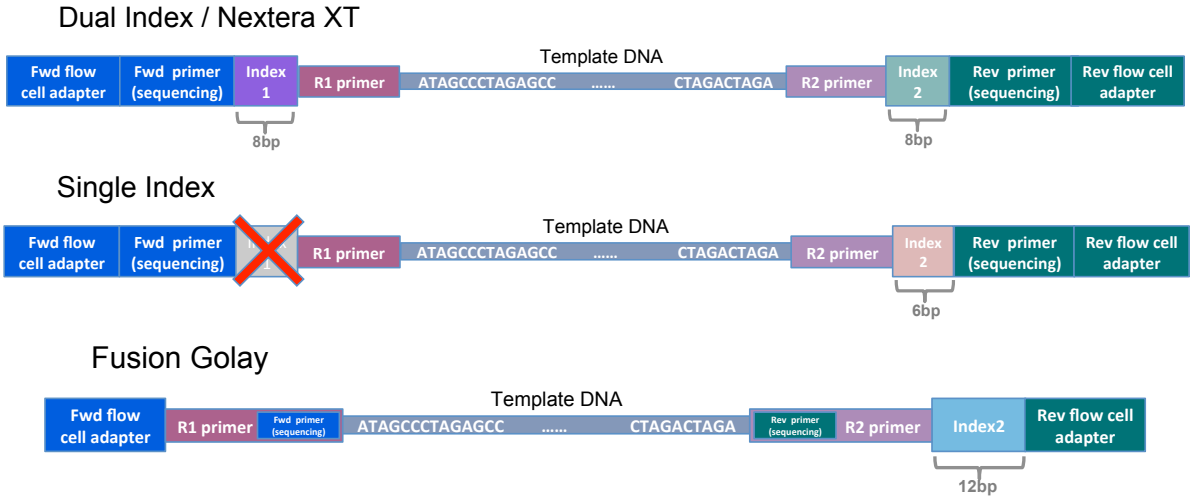


Figure 6.1: Overview of the different amplicon design methods.

were used including nested single index (SI), nested dual index (DI or 5NDI with five random nucleotides before the primer), NexteraXT (XT) and Fusion Golay (FG). The samples were distributed across seven runs and two MiSeq sequencing machines. We tested a range of different input quantities and tested two DNA polymerases (Kapa HiFi & NEB Q5). In addition we studied the impact of different forward and reverse primer combinations. A detailed list of all data sets including their parameters can be found in Tables 6.2+6.3. The data sets are available on the European Nucleotide Archive under the study accession number: PRJEB6244 (http://www.ebi.ac.uk/ena/data/view/PRJEB6244).

Each library preparation method represents a different amplicon design. The Fusion method is simple and cost effective for small amplicon numbers. It requires the design and synthesis of barcoded primers for each sample. In contrast to the other methods, here, the primers do not get sequenced during the run (see Figure 6.1). We tested this approach together with the Golay barcodes, which are customised 12bp error-correcting barcodes first described by Fierer et al. [61].

The tailed amplicon design was used for the DI and SI libraries. This a two-step amplification process. During the first round of PCR a universal primer is attached to both ends of the amplicon. The indices are then added to the universal primers on one or both sides during a second round of PCR to enable multiplexing. Overall, this type of library preparation is more time consuming but also more economical for larger amplicon numbers. We used the standard Illumina indices (I5 and I7) for the DI and SI data sets.

The NexteraXT library preparation method uses an engineered transposome to simultaneously fragment and tagment the DNA. Illumina flow cell adapter plus sequencing primer and optional indices are added during 10 cycles of PCR. This facilitates fast

Table 6.2: Overview of the experimental design for the data sets. Library preparation methods: nested single index (SI), NexteraXT (XT), nested dual index (DI), nested dual index with 5 random nucleotides before primer (5NDI), Fusion Golay (FG); Taq (DNA polymerase): HiFi Kapa (HF), Q5 neb (Q5); Template: *Anaerocellum thermophilum Z-1320 DSM 6725* (AT), *Bacteroides thetaiotaomicron VPI-5482* (BT), *Bacteroides vulgatus ATCC 8482* (BV), *Caldicellulosiruptor saccharolyticus DSM 8903* (CS), *Herpetosiphon aurantiacus ATCC 23779* (HA), *Rhodopirellula baltica SH 1* (RBS), *Leptothrix cholodnii SP-6* (LC), balanced mock community (MB), unbalanced mock community (MUB); Primers: see reference [54] for sequences

Meta ID	Lib. Prep.	Run	Region	Machine	input ng	PCR cycle (R1+R2)	Taq	Template	F & R primer
19	SI	1	V4	Miseq2	4	12+15	Q5	AT	515 & 805RA
20	SI	1	V4	Miseq2	4	12+15	Q5	BT	515 & 805RA
21	SI	1	V4	Miseq2	4	12+15	Q5	BV	515 & 805RA
22	SI	1	V4	Miseq2	4	12+15	Q5	CS	515 & 805RA
23	SI	1	V4	Miseq2	4	12+15	HF	AT	515 & 805RA
24	SI	1	V4	Miseq2	4	12+15	HF	BT	515 & 805RA
25	SI	1	V4	Miseq2	4	12+15	HF	BV	515 & 805RA
26	SI	1	V4	Miseq2	4	12+15	HF	CS	515 & 805RA
27	XT	2	V3/V4	Miseq1	2	15+12	Q5	AT	341f & 806rcb
28	XT	2	V3/V4	Miseq1	2	15+12	Q5	BT	341f & 806rcb
29	XT	2	V3/V4	Miseq1	2	15+12	Q5	BV	341f & 806rcb
30	XT	2	V3/V4	Miseq1	2	15+12	Q5	CS	341f & 806rcb
31	XT	2	V3/V4	Miseq1	2	12+12	HF	AT	341f & 806rcb
32	XT	2	V3/V4	Miseq1	2	12+13	HF	BT	341f & 806rcb
33	XT	2	V3/V4	Miseq1	2	12+14	HF	BV	341f & 806rcb
34	XT	2	V3/V4	Miseq1	2	12+15	HF	CS	341f & 806rcb
35	DI	2	V4	Miseq1	2	12+18	HF	MB	515 & 805RA
36	5NDI	2	V4	Miseq1	2	12+18	HF	MB	515 & 805RA
37	DI	2	V4	Miseq1	2	12+18	HF	MB	515 & 806rcb
38	5NDI	2	V4	Miseq1	2	12+18	HF	MB	515 & 806rcbc27
39	FG	3	V4	Miseq2	10	15	HF	MB	515 & 806rcbc27
40	FG	3	V4	Miseq2	10	15	HF	MB	515 & 806rcbc28
41	FG	3	V4	Miseq2	10	15	HF	MB	515 & 806rcbc29
42	FG	3	V4	Miseq2	1	25	HF	MB	515 & 806rcbc30
43	FG	3	V4	Miseq2	1	25	HF	MB	515 & 806rcbc31
44	FG	3	V4	Miseq2	1	25	HF	MB	515 & 806rcbc32
45	FG	3	V4	Miseq2	10	25	HF	MB	515 & 806rcbc33
46	FG	3	V4	Miseq2	10	25	HF	MB	515 & 806rcbc34
47	FG	3	V4	Miseq2	10	25	HF	MUB	515 & 806rcbc35
48	DI	4	V4	Miseq2	2	12+20	HF	MUB	F515A & 805RA
49	DI	4	V4	Miseq2	2	12+20	HF	MUB	F515A & 805RA
50	XT	4	16S	Miseq2	2	20	HF	MB	27YMF & 1492R
51	XT	4	16S	Miseq2	2	20	HF	MUB	27YMF & 1492R
52	DI	5	V4	Miseq2	2	5+15	HF	MB	F515A & 805RA

Table 6.3: Overview of experimental design for the amplicon data sets. (2)

Meta ID	Lib. Prep.	Run	Region	Machine	input ng	PCR cycle (R1+R2)	Taq	Template	F & R primer
53	DI	5	V4	Miseq2	2	8+15	HF	MB	F515A & 805RA
54	DI	5	V4	Miseq2	2	10+15	HF	MB	F515A & 805RA
59	DI	5	V4	Miseq2	2	8+15	HF	MB	F515A & 805RA
60	DI	5	V4	Miseq2	2	10+15	HF	MB	F515A & 805RA
61	DI	5	V4	Miseq2	2	10+15	HF	MB	F515A & 805RA
62	DI	5	V4	Miseq2	2	8+15	HF	MUB	F515A & 805RA
64	DI	5	V4	Miseq2	2	8+15	HF	MUB	F515A & 805RA
65	DI	5	V4	Miseq2	2	10+15	HF	MUB	F515A & 805RA
66	DI	5	V4	Miseq2	2	10+15	HF	MUB	F515A & 805RA
67	DI	5	V4	Miseq2	2	10+15	HF	MUB	F515A & 805RA
68	DI	5	V4	Miseq2	2	8+15	HF	HA	F515A & 805RA
69	DI	5	V4	Miseq2	2	8+15	HF	LC	F515A & 805RA
71	DI	5	V4	Miseq2	2	8+15	HF	RBS	F515A & 805RA
74	DI	5	V4	Miseq2	2	8+15	Q5	MB	F515A & 805RA
75	DI	5	V4	Miseq2	2	8+15	Q5	MB	F515A & 805RA
76	DI	5	V4	Miseq2	2	8+15	Q5	MB	F515A & 805RA
77	FG	6	V4	Miseq1	2	15	HF	MUB	515 & 806rcb
78	FG	6	V4	Miseq1	5	15	Q5	MB	515 & 806rcb
79	FG	6	V4	Miseq1	5	15	Q5	MB	515 & 806rcb
80	FG	6	V4	Miseq1	5	25	Q5	MB	515 & 806rcb
81	FG	6	V4	Miseq1	5	25	Q5	MB	515 & 806rcb
82	FG	6	V4	Miseq1	5	15	HF	MB	515 & 806rcb
83	FG	6	V4	Miseq1	5	15	HF	MB	515 & 806rcb
85	FG	6	V4	Miseq1	5	25	HF	MB	515 & 806rcb
86	DI	7	V4	Miseq2	2	10+15	HF	MB	515 & 806rcb
87	DI	7	V4	Miseq2	2	10+15	HF	MB	515 & 806rcb
88	DI	7	V4	Miseq2	2	10+15	HF	MB	515 & 806rcb
89	DI	7	V4	Miseq2	2	10+15	HF	MUB	515 & 806rcb
90	DI	7	V4	Miseq2	2	10+15	HF	MUB	515 & 806rcb
91	DI	7	V4	Miseq2	2	10+15	HF	MUB	515 & 806rcb
93	DI	7	V4	Miseq2	2	10+15	HF	AT	515 & 806rcb
94	DI	7	V4	Miseq2	2	10+15	HF	BT	515 & 806rcb
96	DI	7	V4	Miseq2	2	10+15	HF	CS	515 & 806rcb
97	DI	7	V3/V4	Miseq2	2	10+15	HF	MB	341f & 806rcb
98	DI	7	V3/V4	Miseq2	2	10+15	HF	MB	341f & 806rcb
99	DI	7	V3/V4	Miseq2	2	10+15	HF	MB	341f & 806rcb
100	DI	7	V3/V4	Miseq2	2	10+15	HF	MB	341f & 805RA
101	DI	7	V3/V4	Miseq2	2	10+15	HF	MB	341f & 805RA
102	DI	7	V3/V4	Miseq2	2	10+15	HF	MB	341f & 805RA

preparation times and only requires low amounts of input DNA (1ng). Here, we used the standard Illumina indices (I5 and I7). Figure 6.1 provides an overview of the different amplicon design methods. For further details see [54], [13].

Reference Database

For the construction of the reference database we first blasted an *E. coli* 16S rRNA gene against the full length genomes of all organisms in the mock community. If this resulted in less than four 16S rRNA variants for an organism, we searched for additional sequences directly on the NCBI database and added the relevant hits to our reference database. Subsequently we aligned all unique sequences (separately for each organism) to filter out redundant sequences including subsequences. This resulted in 116 database entries. To verify these sequences and in order to identify single nucleotide polymorphisms (SNPs) we used VarScan [85] in combination with a metagenomic data set and a full length 16S rRNA data set of the mock community. The metagenomic data set contained approximately 76 million reads after quality trimming and with a minimum read length of 60bp. With Burrows-Wheeler Aligner (BWA) we identified almost 128,000 16S rRNA reads. The full length 16S rRNA data set contained 2.2 million reads after quality trimming plus filtering and ≈1.9 million of these reads aligned to the *E. coli* reference 16S rRNA gene. In order to avoid mistaking library specific errors for SNPs we only incorporated SNPs that were detected for both data sets. Repeating the analysis three times revealed 33 SNPs. For the metagenomic data set VarScan identified no more SNPs after that. All SNPs that were simultaneously detected for an organism were added as a single variant. The final 16S rRNA reference database comprises 128 sequences.

6.4 Algorithm for Computing the Error Profiles

First, we aligned the reads with BWA (Version 0.7.3a-r367) [96] against the reference sequences. Then we converted the alignment to Sequence Alignment/Map (SAM) format using BWA and generated the MD tag with SAMtools [97]. Our program then infers position and nucleotide specific substitution, insertion and deletion rates. The Compact Idiosyncratic Gapped Alignment Report (CIGAR) string encodes matches and mismatches with "M", insertions with an "I" and deletions with "D". Based on the MDtag we then identified the nucleotides that were replaced during a substitution and the types of nucleotides affected during a deletion. From the extended CIGAR string we determined the substituting nucleotides and detected the nucleotides involved in an insertion. In addition we recorded position specific quality scores for all error types and the 3mers preceding errors (motifs) not containing the erroneous base itself.

Our program outputs 4xL matrices for each error type (where L is the read length) for the set of R1 and R2 reads, respectively. The number of rows corresponds to the read

length and each row specifies the nucleotide specific error rates for a certain position on the read. We then normalised these matrices by counting the number of occurrences of each nucleotide on the read for each position, accounting for errors where the nucleotide should have been observed. We then added the number of detected substitutions for this nucleotide, and finally subtracted the number of substitutions where the nucleotide was the substituting nucleotide, i.e. was mistaken for another nucleotide. This reflects the true number of occurrences of A, C, G and T.

To verify our algorithm we extended our read simulation program (presented in Chapter 3) to generate reads based on error profiles of the above described format. The error profiles, inferred from the simulated reads, concurred with the original error profiles used to simulate the reads. In addition we used mock error profiles with a simple stepwise increase of the error rates along the read for the read simulation. Again, the reconstructed error profiles concurred with the mock profiles used for the simulation and thus validate the algorithm. The algorithms are implemented in Perl and Bash and are available on https:/bitbucket.org/ms_research/ep.

Metric for Overall Comparison: Hellinger Distance

We measured the similarity between the error distributions using the Hellinger distance. The error rates across the read length can be interpreted as probability distributions. We considered substitutions, insertions and deletions separately for R1 and R2 reads, respectively, and summed over the different types of errors in each case.

Definition: Let $P = (p_1, \ldots, p_L)$ and $Q = (q_1, \ldots, q_L)$ denote two discrete probability distributions. Then the Hellinger distance H is defined as

$$H(P, Q) = \sqrt{\frac{1}{2} \sum_{i=1}^{L} (\sqrt{p_i} - \sqrt{q_i})^2}$$

A value between zero and one is returned. The closer this value is to zero the more similar the two distributions.

6.5 Results

We only present the detailed results for data set *DS35*. However, the same detailed analysis was conducted for all data sets listed in Tables 6.2+6.3 and two additional data sets can be found in Appendix A. To compare the individual profiles and to identify any patterns associated with particular parameters we then used the Hellinger distance to contrast the error and quality profiles of the different data sets. Subsequently we studied the overall error rates across all library preparation methods and identified associated

Table 6.4: A selection of substitutions that occurred at a very high rate in data set *DS35*. Column 1-3 specify the type of substitution, its position and the substitution rate for the R1 reads. Column 4-6 detail the respective information for the R2 reads.

R1:		rate	R2:		rate
A -> G	pos 226	25%	A -> G	pos 57	3%
T -> G	pos 162	2%	T -> C	pos 136	2%
T -> G	pos 179	1%	G -> A	pos 57	3%
C -> G	pos 118	18%	G -> C	pos 174	14%

biases and motifs. We conclude this section by testing the efficiency of several currently available error removal techniques.

Detailed Error and Quality Profiles for Data Set DS35

For data set *DS35* the V4 region of the balanced mock community was amplified and the nested dual index library preparation method was used. Figure 6.2 displays the position and nucleotide specific substitution rates for the R1 and R2 reads, respectively. A small number of errors can result in a high error rate if a nucleotide has very few occurrences at a certain position. In order to avoid overemphasis of these rare errors we smoothed the error profiles for the visualisation as follows: For the substitutions we computed the expected minimum number of errors, averaging over all positions. In the case of *DS35* T shows the smallest average error rate (0.000262). There are 593,868 R1 reads, so assuming a uniform distribution over nucleotides we would expect approximately 148,467 occurrences of each nucleotide at each position and thus approximately 38 errors (rounded down). Analogously, we set the minimum threshold for the R2 reads to 144 (smallest nucleotide specific error rate: 0.000967). The insertion and deletion rates as well as the rates of unknown nucleotides (Ns) were calculated relative to the total coverage of each position. This avoids the problem of overemphasis and hence we do not need to apply a minimum threshold.

Substitution Error Profiles

For all types of substitutions we observed an accumulation of errors across the first 10 bp of the reads. The error rates also increased towards the end of the read in particular for the R2 reads and we can see a clear preference for the substituting nucleotide for some types of substitutions. We compared the substitution preference for each original nucleotide across the last 50bp. For the R1 reads we detected the following rates: In 66% of the cases A got substituted by C. For C we observed a substitution with A in 58% of the cases. G got substituted by T in 45% of all cases and T got substituted by

Substitutions:

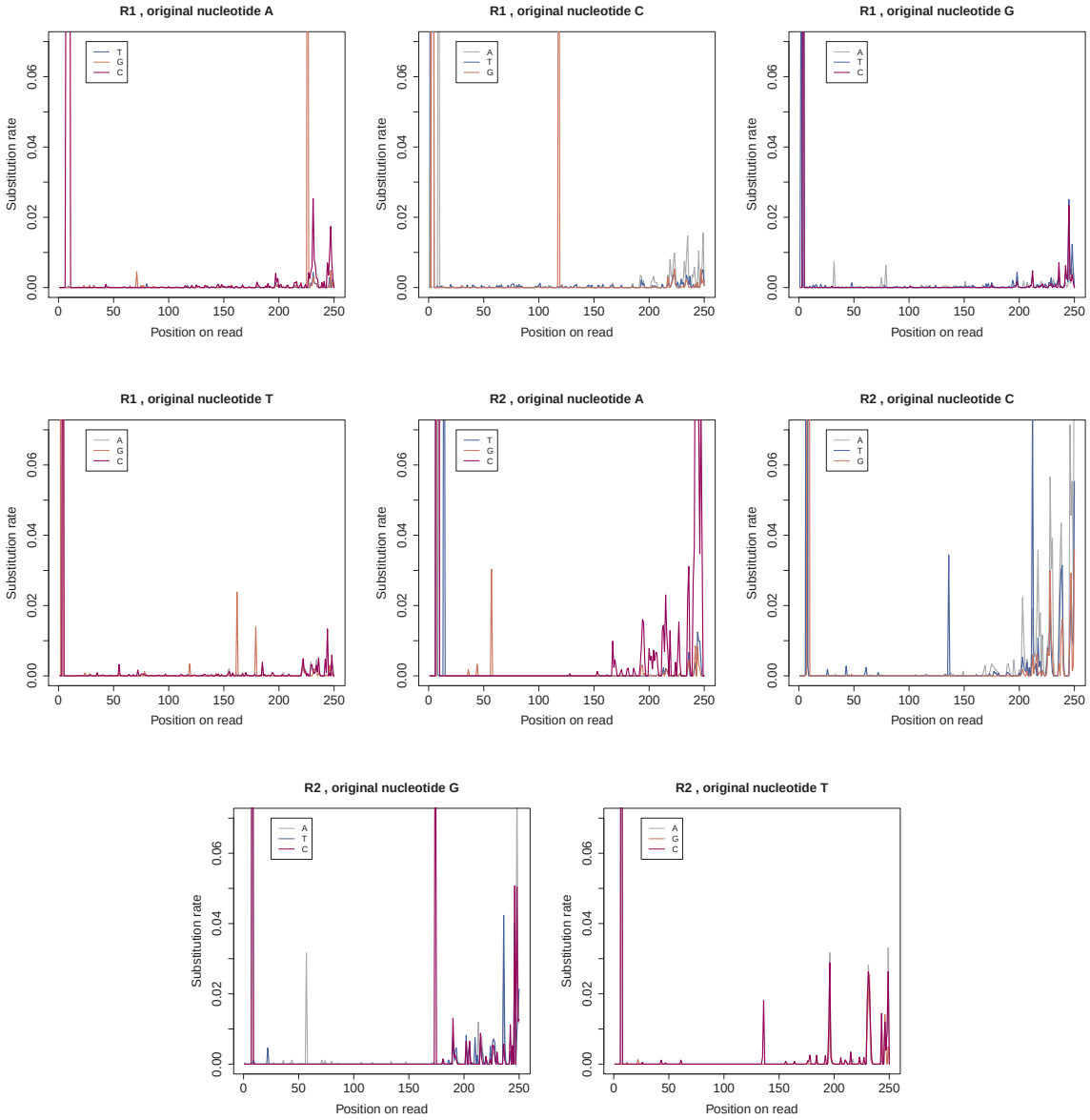
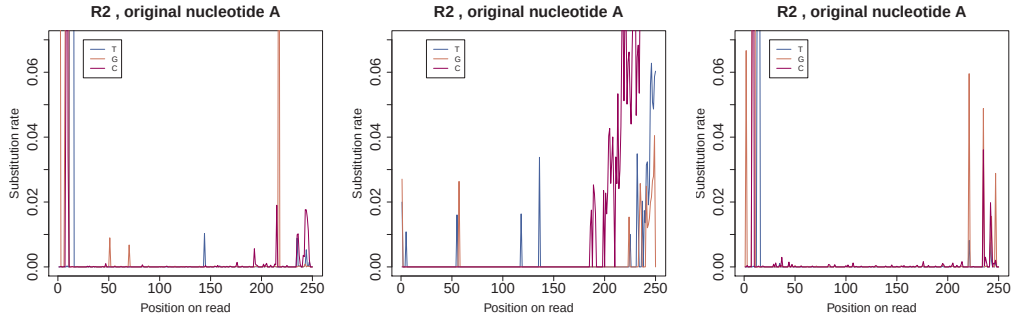


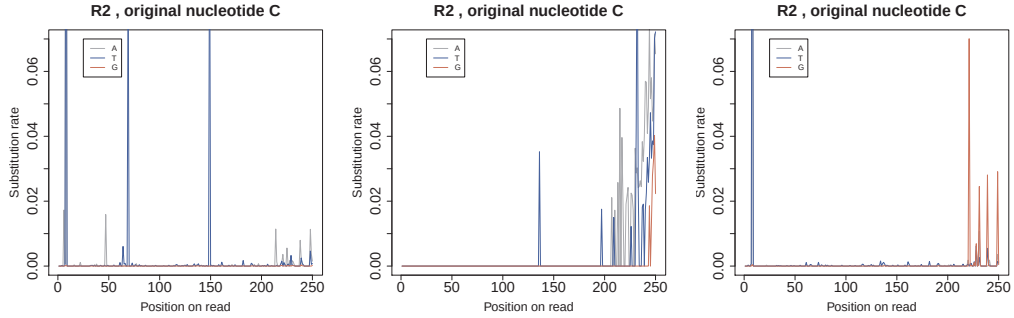
Figure 6.2: Nucleotide specific substitution error profiles for amplicon data set *DS35*: each graph shows the substitution rates for a specific original nucleotide and the colours indicate the substituting nucleotide. The first four graphs show the R1 profiles and the last four graphs show the R2 profiles.

C in 45% of all cases. For the R2 reads we detected a similar bias: We observed A to C substitutions in 85% of all cases, C to A in 61%, G to T in 40% and T to C in 40% of all cases. We also found that the overall error rate is significantly higher in the R2 reads with ≈0.0107 compared to only ≈0.0064 for the R1 reads.

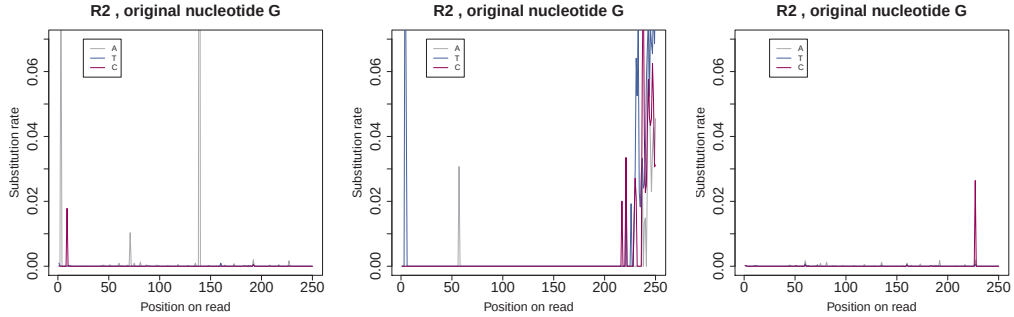
Another noticeable characteristic were the spikes occurring at certain positions across the reads with error rates much higher than the average error rate. There are several possible underlying reasons for the accumulation of errors at those positions. We first



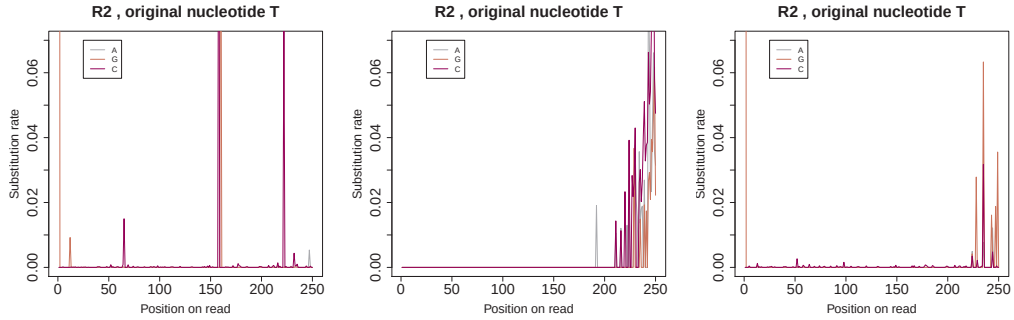
(a) Sub. profiles for R2 (orig. nucleotide A): *Bacteroides thetaiotaomicron VPI-5482*



(b) Sub. profiles for R2 (orig. nucleotide C): *Bacteroides thetaiotaomicron VPI-5482*



(c) Sub. profiles for R2 (orig. nucleotide G): *Bacteroides thetaiotaomicron VPI-5482*



(d) Sub. profiles for R2 (orig. nucleotide T): *Bacteroides thetaiotaomicron VPI-5482*

Figure 6.3: Error profiles for three *Bacteroides thetaiotaomicron VPI-5482* data sets: Each library was constructed with a different method. For the data set displayed in the first column the nested single index was used, for the data set displayed in the second column NexteraXT was used and the last library was constructed with the nested dual index.

Table 6.5: Examples of indels occurring at rates considerably higher than the average insertion and deletion rates.

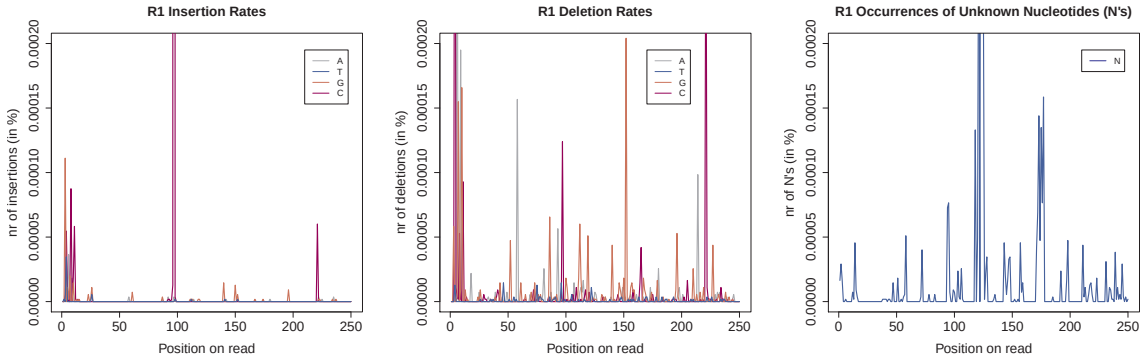
Insertions:		rate	Deletions:		rate
R1	pos 97	0.9%	R1	pos 221	0.03%
R2	pos 72	0.007%	R2	pos 32	0.02%
R2	pos 195	0.8%	R2	pos 70	0.04%

checked if the spikes are likely to be caused by errors in the database. We compared the R2 substitution profiles for three different data sets (see Figure 6.3). In all three cases the organism *Bacteroides thetaiotaomicron VPI-5482* was sequenced. For the first data set (column one) the V4 region of the sample was amplified and prepared with the nested single index, for the second data set (column two) the V3/V4 region was amplified and the library was constructed with the NexteraXT kit and for the third data set (column three) the V4 region was amplified and the nested dual index was used. For the visualisation we smoothed the error profiles accordingly. As all R2 reads cover the V4 region, any issues with the reference database should be visible in all three error profiles. The graphs clearly illustrate that the spikes are not concurrent and thus indicate that it is unlikely that the cause of the spikes are errors in the database. Another indication that the spikes are not a problem with the reference sequences is the rate at which those substitutions occurred. Table 6.4 gives a selection of spikes that were encountered in *DS35* specifying the type of substitution, the position and the rate at which the substitution was observed. The organisms in this mock community were initially uniformly distributed. However, PCR amplification introduces a bias as not all sequences are amplified in equal measure. Therefore, we re-calculated the abundance distribution of the 16S rRNA reference genes based on the read alignments. On average each reference sequence accounted for 0.86% of the population with a maximum of 2.8%. Thus the frequency of each 16S rRNA sequence is in most cases significantly lower than the error rate of the respective spike and errors would need to occur simultaneously in multiple sequences to account for the observed rates.

Insertion and Deletion Error Profiles

Figure 6.4 displays the position specific insertion and deletion profiles as well as the distribution of unknown nucleotides (Ns) across all reads. Insertion and deletion (indel) rates are ≈100x lower than the substitution rates. We also observed that insertions with rates of 0.000040 and 0.000043 for R1 and R2 reads, respectively, are twice as likely as deletions for which we observed rates of 0.000017 and 0.000027 for R1 and R2 reads, respectively. Again, the majority of indels seem to concentrate around certain positions with rates up to 225x higher than the average indel rate (see Table 6.5). The

R1 Profiles for Insertions, Deletions and Ns:



R2 Profiles for Insertions, Deletions and Ns:

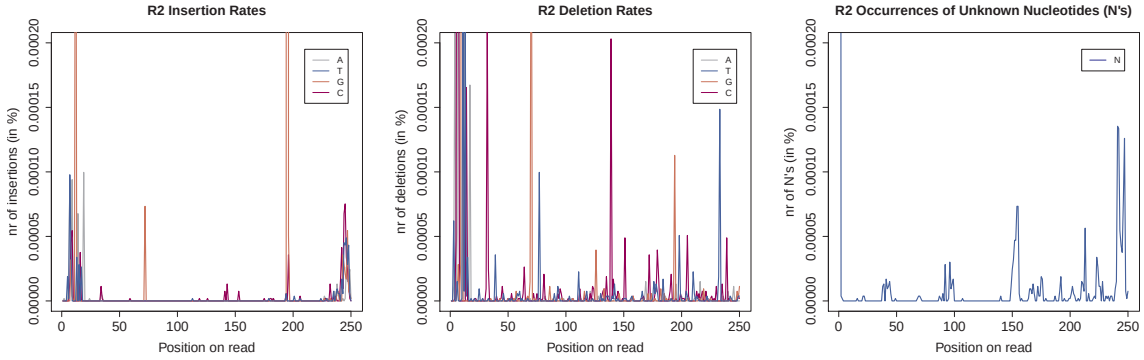
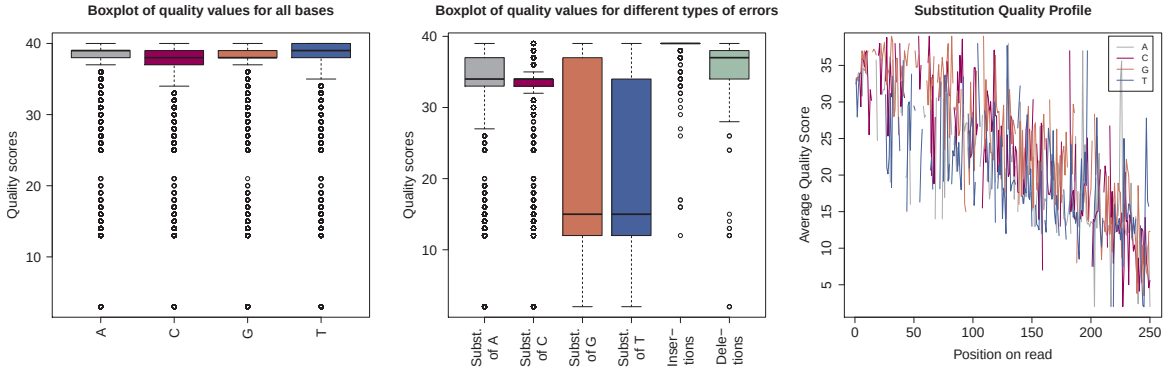


Figure 6.4: Error Profiles for insertions, deletions and unknown nucleotides (Ns): The first three graphs show the R1 error profiles. For insertions the colour identifies the inserted nucleotide and for deletions the colour refers to the type of nucleotide that was deleted. The lower three graphs display the error profiles for the R2 reads, respectively.

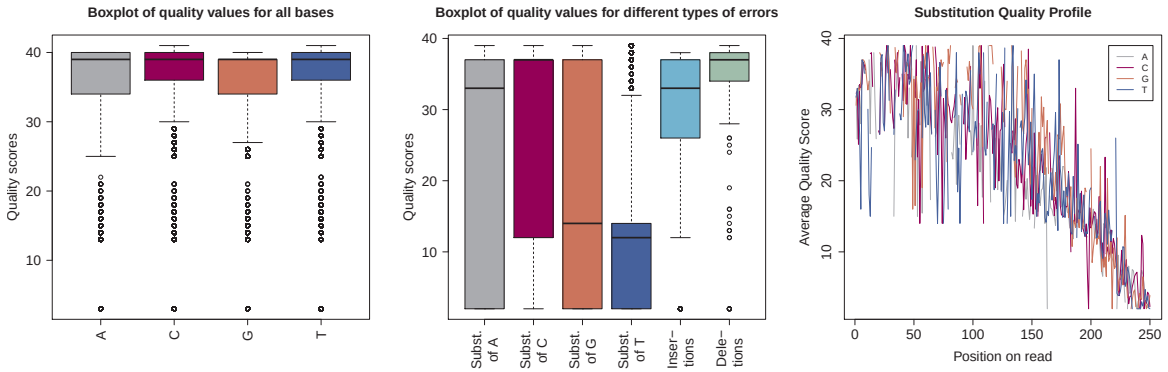
non-uniform distributions of unknown nucleotides (N) indicate that Ns as well do not occur randomly.

Correlation of Quality Scores and Errors

The first column of Figure 6.5 displays the observed quality scores for all reads. For this data set we generally encountered very high quality scores for the R1 reads and only slightly lower values for the R2 reads. In the second column we constrained the boxplot to quality scores associated with substitution errors. Most noticeable is the range of quality scores for substitutions of As and Cs. The average quality score for those types of errors was only slightly lower than the average quality score observed for the respective base in general. Furthermore almost all of the quality scores associated with substitutions of C are between 32 and 35 and 75% of the quality scores associated with substitutions of A are above 32 for the R1 reads. The R2 reads showed a larger range for those error types, though a significant number of errors was also associated with very high quality scores. Erroneous Gs and Ts show on average much lower quality



(a) R1 quality profiles

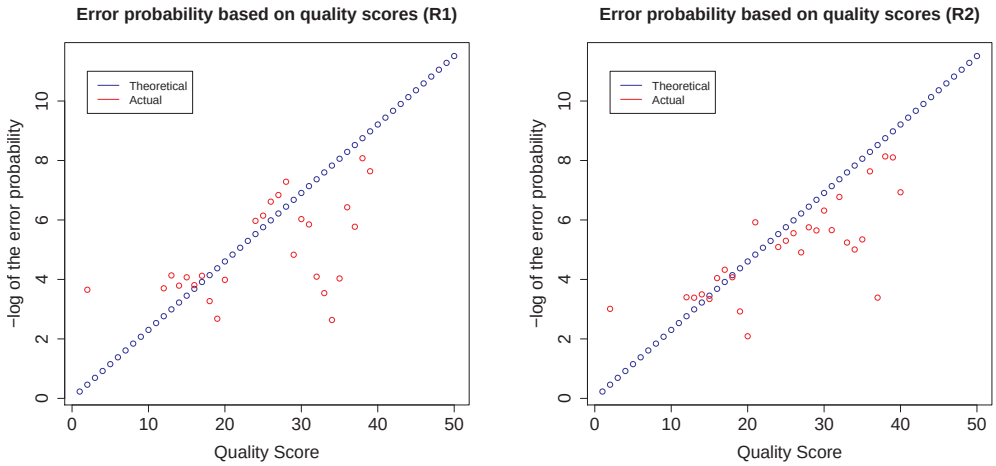


(b) R2 quality profiles

Figure 6.5: Quality profiles for R1 and R2 reads: The boxplots in the first column display the distribution of quality scores for all reads. The second column shows the distribution of quality scores associated with errors and the last column shows the average quality score of substitution errors for each position across the read.

value. G and T are read by the same laser (green channel). Erroneous bases sequenced on the red channel have on average very high quality values and cannot be detected based on the reported quality score. We observed the same issue for insertions and deletions. In R1 reads 75% of the indels showed quality scores of 35 and above. In R2 reads the same was true for deletions, for insertions the average quality score dropped just below 35. The last column of Figure 6.5 shows the position specific substitution quality profiles and suggests that there is a correlation between position of the error and its quality value. Errors occurring at the start and middle of the read had in general much higher quality scores and the quality value decreased towards the end of the reads.

In order to evaluate the suitability of quality scores to identify errors, we compared the theoretical accuracy to the actual accuracy in Figure 6.6. All displayed quality values of the actual accuracy were observed at least 2,000 times. The theoretical accuracy was higher than the actual accuracy for many of the high quality scores (in particular for the R1 reads) whereas the actual accuracy of the lower quality scores was much higher than the the theoretical accuracy. The figure highlights that the quality scores is not



(a) Accuracy of quality scores in R1 reads (b) Accuracy of quality scores in R2 reads

Figure 6.6: Comparison of theoretical accuracy (blue) of the quality scores and actual accuracy (red) for data set *DS35*.

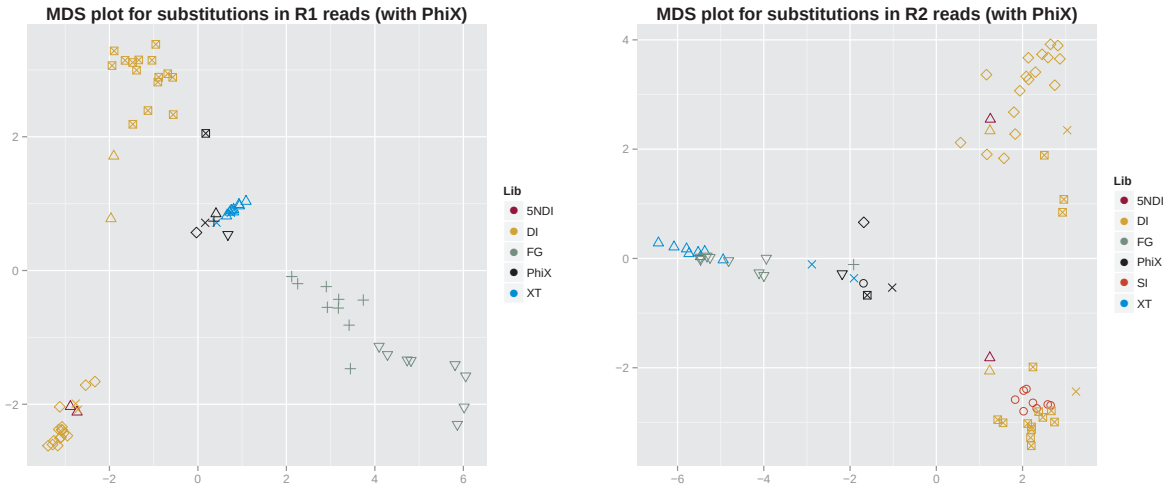
significant in most cases. It is possible that some of the high quality scores associated with errors refer to PCR errors introduced before the actual sequencing step. Nevertheless these results indicate that quality scores are of limited use for the identification of errors in this amplicon data sets as low quality values do not reliable reflect the error potential of the respective base.

Overall Comparison of Error and Quality Profiles

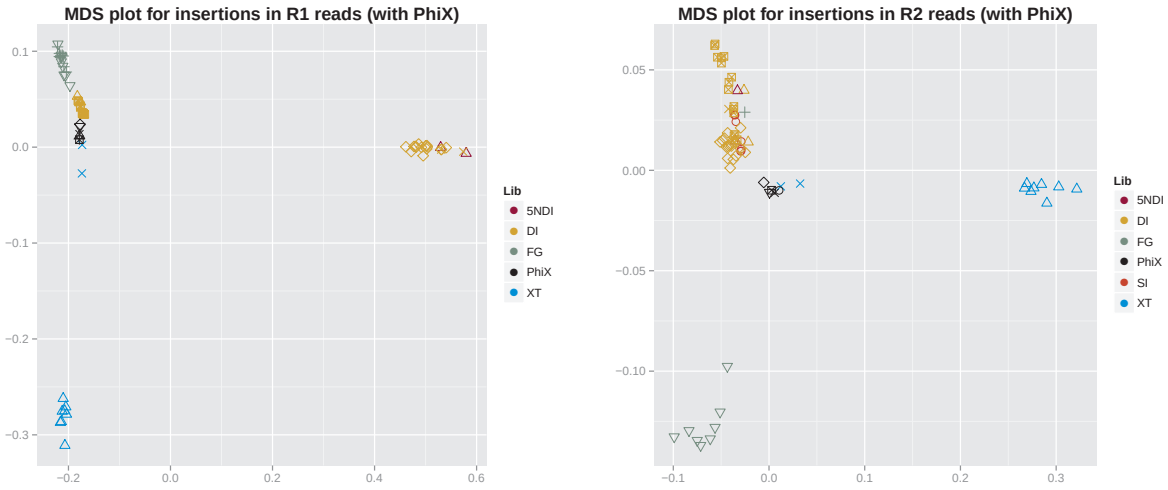
We tested a range of factors across 73 data sets including five different library preparation methods, amount of input DNA, number of PCR cycles, two different Taqs, sample/template impact, region (V3 versus V3/V4 and 16S rRNA), machine (two different MiSeqs were used), different forward and reverse primers as well as run specificity of the errors. We analysed each data set as described above and computed the corresponding error and quality distributions. We then compared those distributions using the Hellinger distance in order to identify patterns and to determine the experimental factors associated with those patterns. As the Hellinger distance places less emphasis on spikes, there was no need to smooth the distributions prior to computing the distance matrices.

Error Profiles

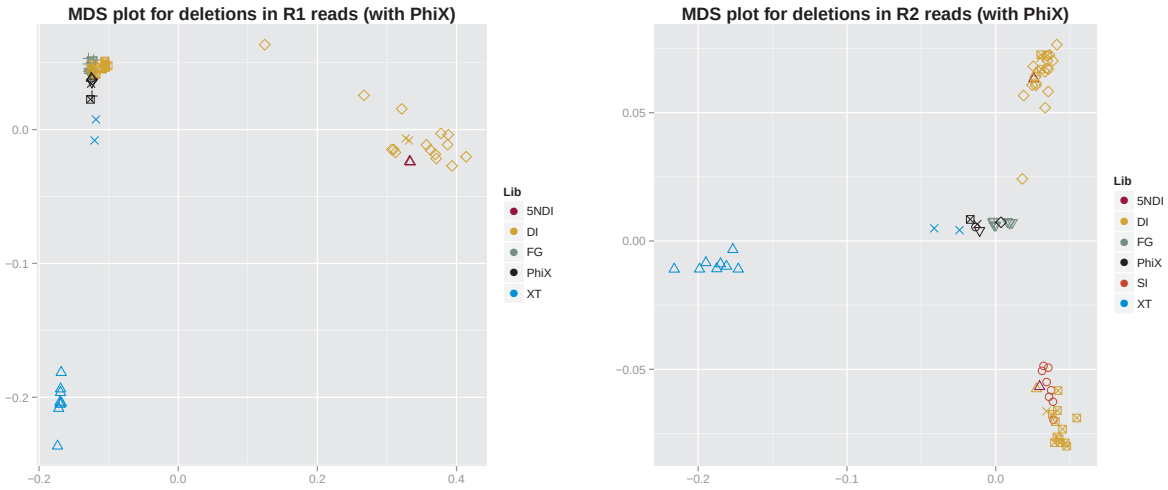
We visualised the level of similarity of the position specific error distributions by means of multidimensional scaling (MDS) (see Figure 6.7). In order to derive meaningful error distributions for a data set, we required at least 1,000 aligned reads per data set. (Note that none of the SI data sets held $\geq$1,000 aligned R1 reads. The SI data sets were thus



(a) Substitutions



(b) Insertions



(c) Deletions

Figure 6.7: Comparison of error distributions across all amplicon data sets. We used the Hellinger distance to construct similarity matrices. The colours indicate the library preparation method and the shapes indicate different runs.

Table 6.6: Results of permutation ANOVA for R1 and R2 substitutions.

R1 reads	Df	SumsOfSqs	MeanSqs	F.Model	R2	Pr(>F)
Library Preparation	3	632.26	210.753	24.1463	0.36841	0.001
Run	1	110.02	110.018	12.6049	0.06411	0.001
input ng	1	19.29	19.286	2.2096	0.01124	0.040
PCR Cycle R1+R2	9	234.16	26.018	2.9809	0.13644	0.001
Taq	1	31.02	31.025	3.5546	0.01808	0.004
Template	8	122.45	15.307	1.7537	0.07135	0.002
F R Primer	13	340.06	26.158	2.9970	0.19815	0.001
Residuals	26	226.93	8.728		0.13223	
Total	62	1716.19			1.00000	
R2 reads	Df	SumsOfSqs	MeanSqs	F.Model	R2	Pr(>F)
Library Preparation	4	1017.07	254.267	20.5682	0.44486	0.001
Run	1	112.37	112.374	9.0902	0.04915	0.001
input ng	1	60.47	60.469	4.8915	0.02645	0.001
PCR Cycle R1 + R2	10	310.93	31.093	2.5151	0.13600	0.001
Taq	1	18.37	18.374	1.4863	0.00804	0.129
Template	8	180.77	22.597	1.8279	0.07907	0.001
F R PRIMER	5	165.95	33.190	2.6848	0.07259	0.001
Residuals	34	420.31	12.362		0.18384	
Total	64	2286.24			1.00000	

not included in the R1 figures.) Across all types of errors there was a distinct tendency to cluster according to library preparation (indicated by colour) and run (indicated by shape). The R1 substitution profiles for the Fusion Golay, for example, formed a distinct cluster. This cluster consists in turn of two subclusters reflecting that the samples were sequencing on two different runs. The dual index and 5N dual index data sets clustered as well though we observed a higher degree of variability between different sequencing runs. The NexteraXT data sets clustered tightly aside from two data points representing the full length 16S rRNA samples. The PhiX data sets from each run formed their own distinct cluster. This is in accordance to the assumption that the library preparation has a major impact on the error distribution as the adapters used for PhiX are the same as for the TruSeq library preparation method and would thus show a distinct pattern. This also implies that PhiX is not suitable to identify error rates or patterns if the actual sample was prepared with a different library preparation method.

We used the Hellinger distance to create distance matrices followed by a permutation ANOVA to determine how much of the variation can be explained by the experimental

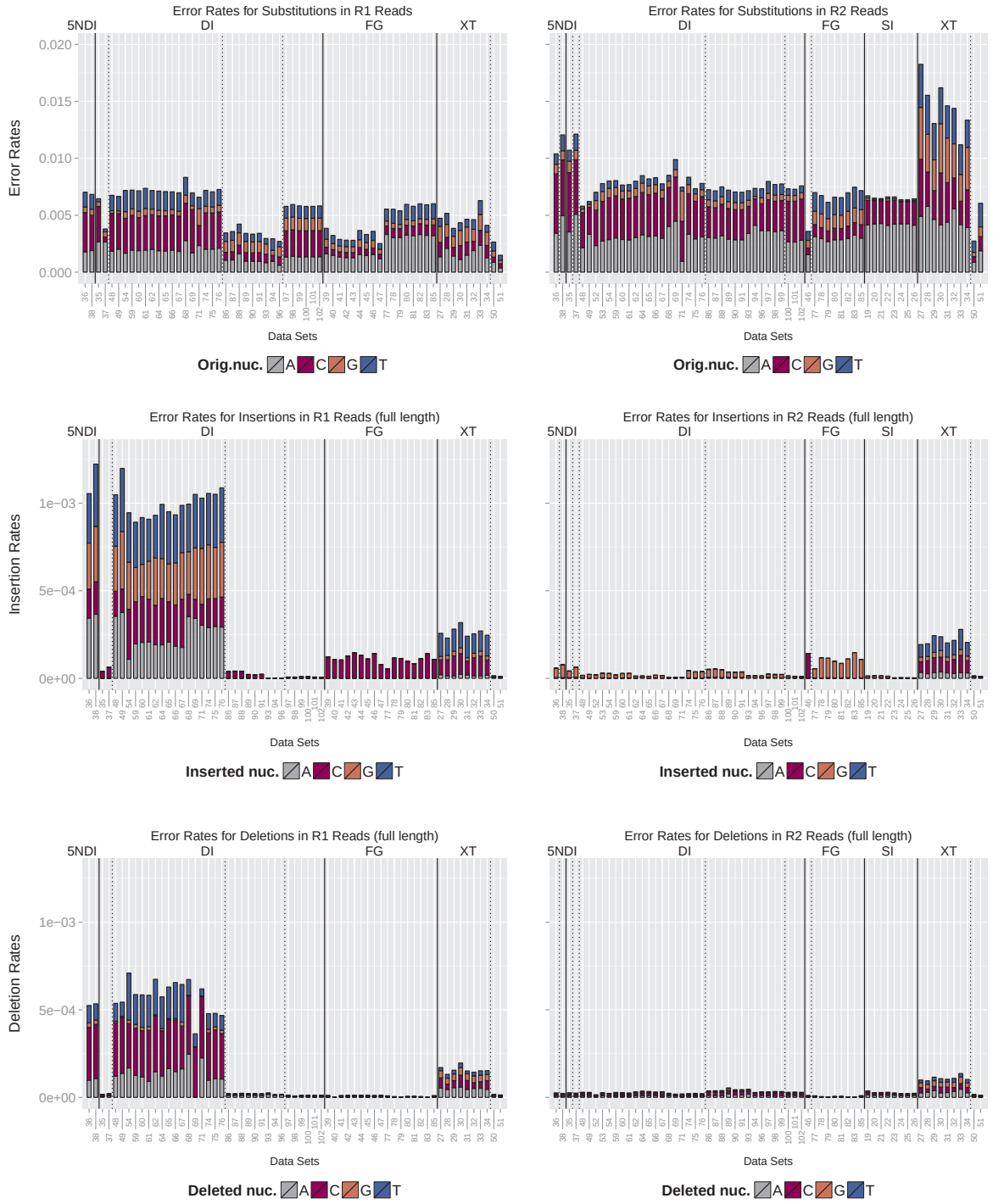


Figure 6.8: Comparison of the overall error rates for each amplicon data sets. The lower x-axis indicates the name of the data set and the upper x-axis specifies the library preparation method. The dashed lines further distinguish between different forward and reverse primers. The error bars show the extent that each original nucleotide contributed to the error rate.

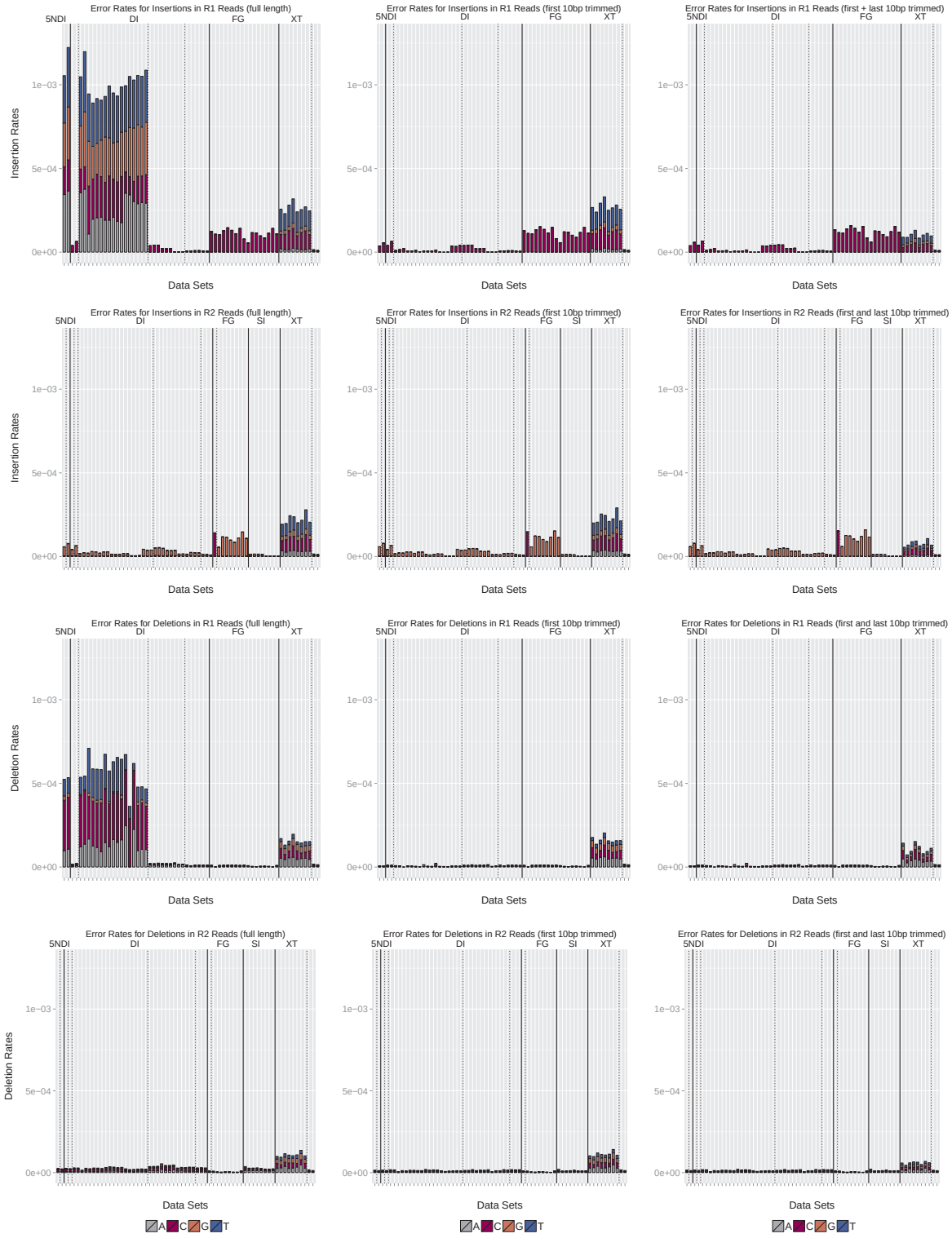


Figure 6.9: Trimming the start and end of the read to remove indels: The first column shows the R1 and R2 indel rates for the raw reads (full length). The second column shows the error rates after trimming the first 10bp and the last column shows the error rates after additionally trimming the last 10bp. Data sets indicated on the x-axis are grouped by library preparation method (solid line) and primers (dashed line) (from left to right): 36, 38, 35, 37, 48, 49, 54, 59, 60, 61, 62, 64, 65, 66, 67, 68, 69, 71, 74, 75, 76, 86, 87, 88, 89, 90, 91, 93, 94, 96, 97, 98, 99, 100, 101, 102, 39, 40, 41, 42, 43, 44, 45, 46, 47, 77, 78, 79, 80, 81, 82, 83, 85, 27, 28, 29, 30, 31, 32, 33, 34, 50, 51

factors and to identify the factors driving the clustering that we observed in Figure 6.7. The experimental factors included library preparation method, run, region, machine, input ng, PCR Cycle (R1+R2), Taq, template and forward plus reverse primer. We used stepwise regression with RDA to determine which factors to include and their order based on the R1 data and used the same model for the R2 reads. The resulting model (library preparation method + run + input ng + PCR cycle R1&R2 + Taq + template + forward & reverse primer) was used for the adonis function of the vegan R package. For both R1 and R2 substitutions the library preparation method was identified as the major factor, explaining 37% and 44% of the variability, respectively. (The details of the ANOVA results can be found in Table 6.6.)

Comparison of Error Rates for Different Library Preparation Methods

In Figure 6.8 we compared the overall error rates of the data sets grouped by library preparation method and forward and reverse primer. Note that we only considered aligned reads here. An overview of the percentage of aligned reads is given in Figure 6.16 (rates for raw reads are marked in grey). For all of the data sets the error rates increased for the R2 reads. We noted the most dramatic increase for some of the NexteraXT data sets where the error rate for the R2 reads was more than double the rate of the R1 reads. We also noticed a certain amount of variation for each library preparation method. In the case of the FG data sets, for example, *DS39-DS47* were on the same sequencing run and showed a lower rate compared to the other FG data sets which were sequenced on a different run. For the DI data sets four different forward primers were used. The first two data sets, the following 17 data sets, the following nine data sets and the last six data sets have the same forward primer, respectively. There was also a clear bias of A and C which, in particular for the R2 reads, accounted for a large fraction of the overall error rate. This could indicate a general problem with the red laser as both A and C are read by the same laser.

Indel rates are in general almost two orders of magnitude smaller than the substitution rates. Though for 17 of the DI data sets and the two 5NDI data sets we recorded a huge increase in insertions and deletions on the R1 reads. For all of these data sets the 515 or F515A forward primer was used. For the data sets where the same forward primer was used in connection with the Fusion Golay we did not record the same build-up of indels. The majority of those errors occurred at the start of the reads. By trimming the first 10bp of the reads we were able to remove 95%-100% of all insertion errors for those data sets and 96%-100% of all deletion errors (see Figure 6.9).

We also detected a preference for the substituting nucleotide for the different library preparation methods. This bias seems to be mostly run specific, though we recorded a high preference for G as the substituting nucleotide in R1 reads and T in R2 reads for

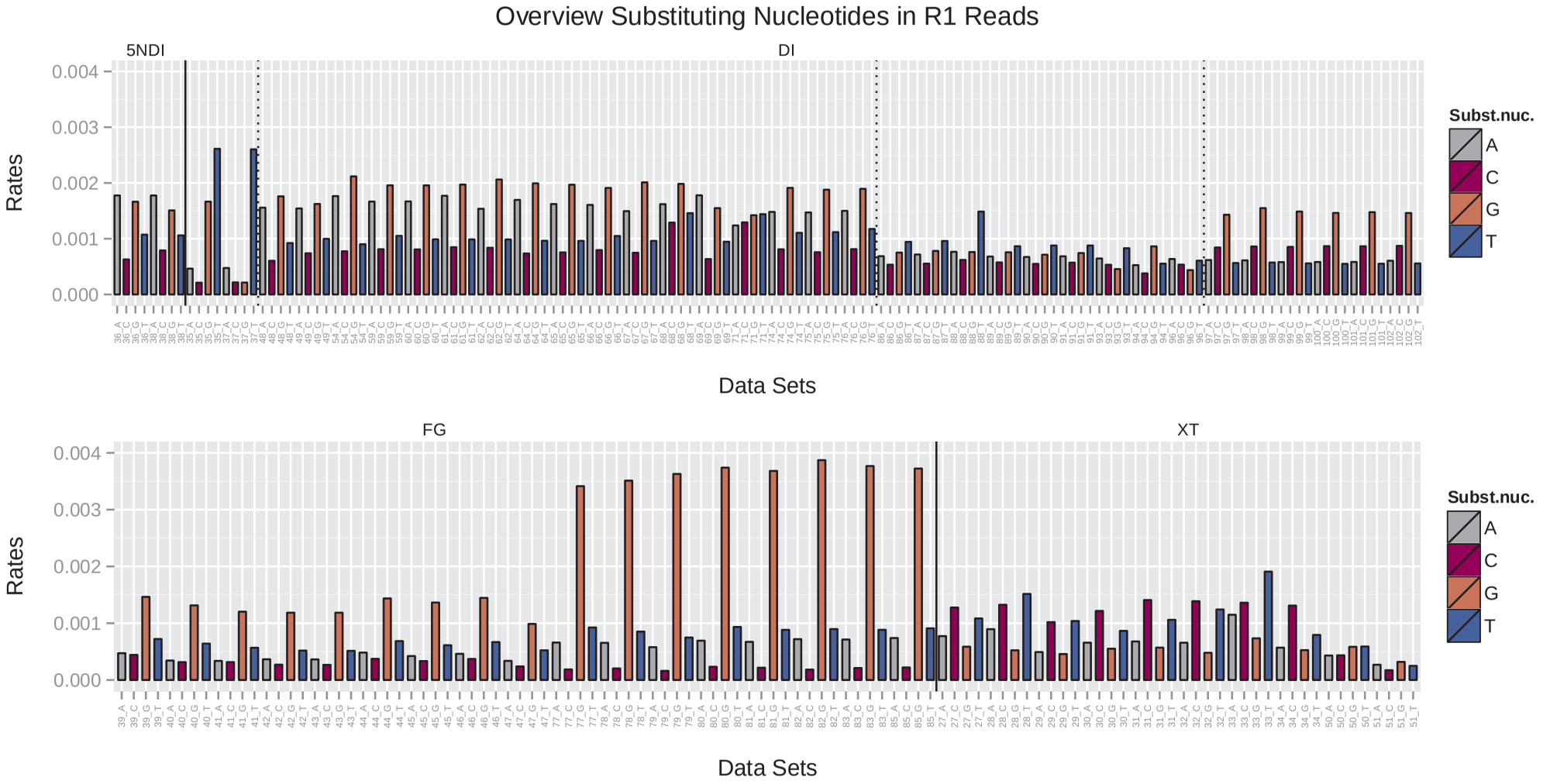


Figure 6.10: Rate of each substituting nucleotide in R1 reads for the 5N dual index and dual index data sets (upper plot) and the Fusion Golay and NexteraXT data sets (lower plot). Data sets are separated according to library preparation method (solid line) and primers (dashed line).

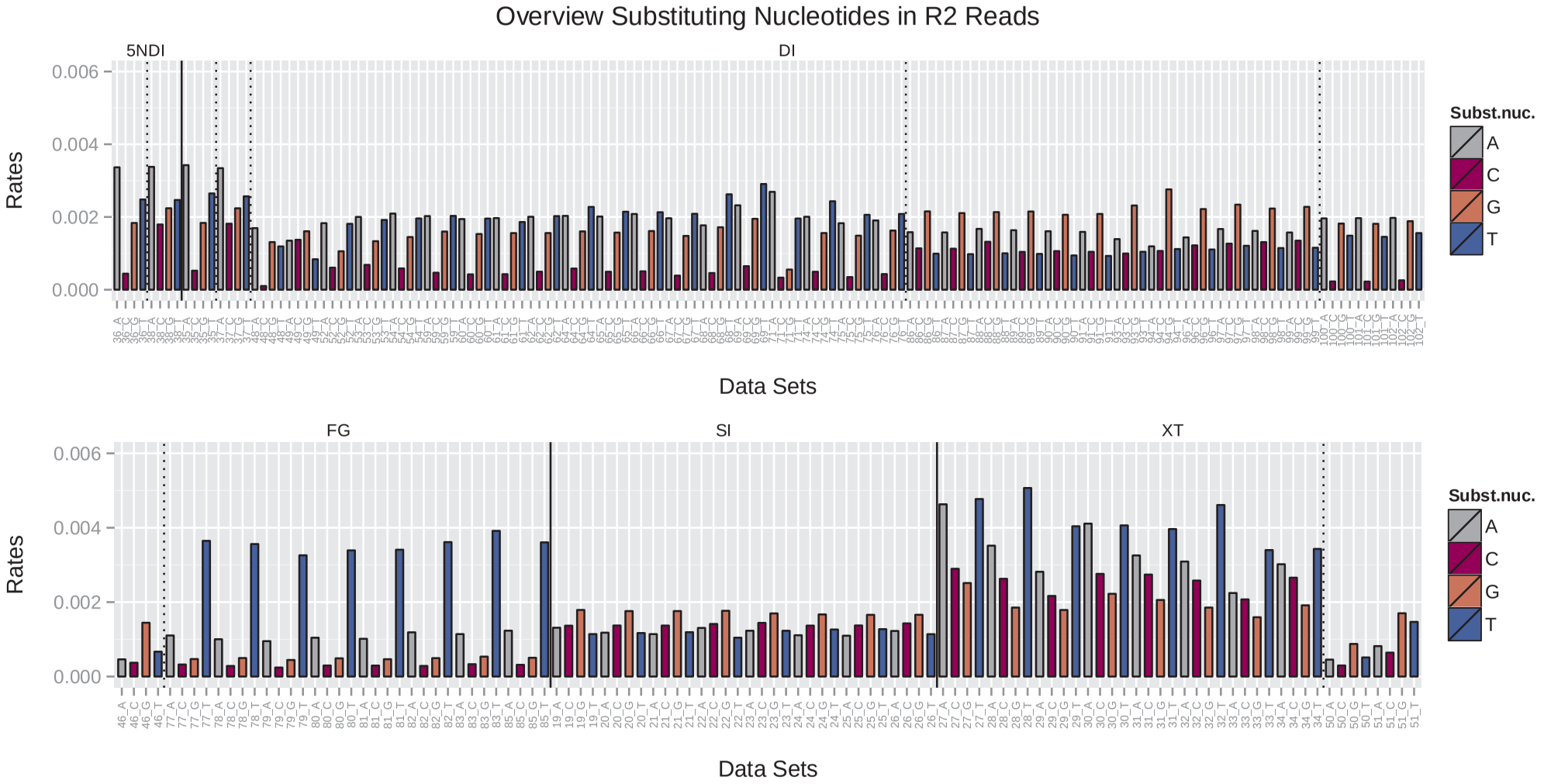


Figure 6.11: Rate of each substituting nucleotide in R2 reads for the 5N dual index and dual index data sets (upper plot) and the Fusion Golay, single index and NexteraXT data sets (lower plot). Data sets are separated according to library preparation method (solid line) and primers (dashed line).

the Fusion Golay. For *DS82*, for example, G was the substituting nucleotide in 68% of all substitutions that occurred on R1 reads and T in 65% of all substitutions that occurred on R2 reads. (For more details see Figure 6.10+6.11.)

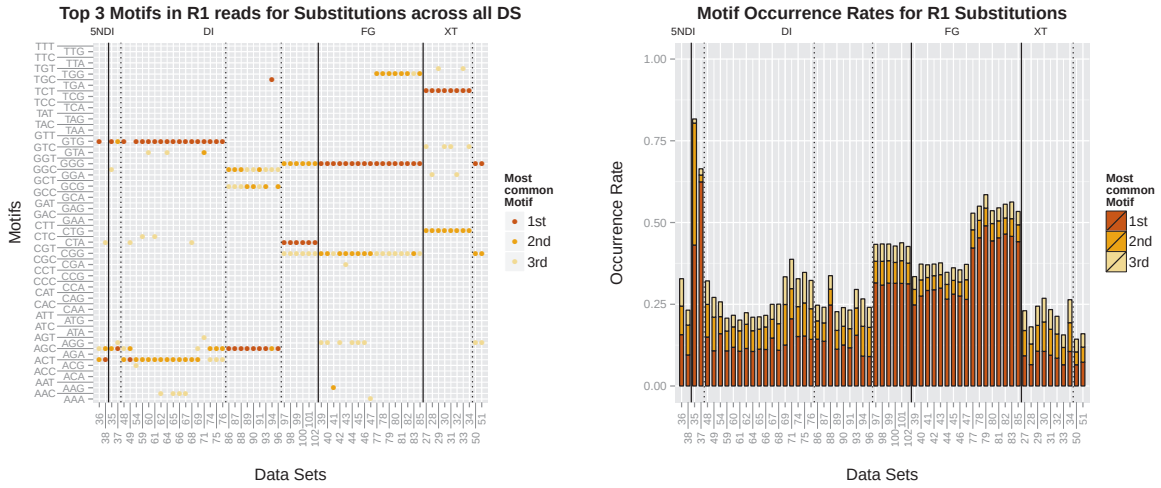
Motifs

Nakamura et al. [113] previously reported "sequence patterns that trigger sequence-specific errors" for the Illumina Genome Analyzer (GAII) in 2011. Since then there have been major developments with another four Illumina platforms entering the market and improvements regarding the chemistry providing much longer reads with lower error rates. Nakamura's findings are based on a single library preparation method and the read length was limited to 36bp. We tested if a similar bias prevails for the MiSeq platform with read lengths of 2x250bp testing five different library preparation methods. We additionally assessed the impact of different experimental factors.

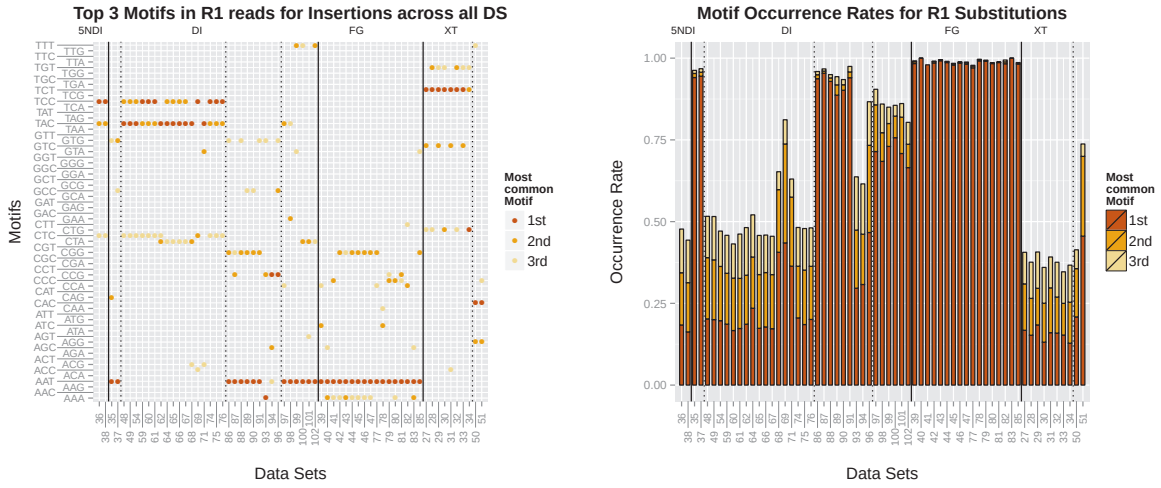
We recorded all 3mers preceding errors (in the following referred to as *motifs*) for substitutions, insertions and deletions and measured the percentage of errors that is explained by the top three motifs. Figure 6.12 shows the results for the R1 reads. (The analogous results for the R2 reads can be found in Figure 6.13.)

In particular for the substitutions, Figures 6.12a and 6.13a show that the three most common motifs are very similar for data sets with the same library preparation (separated by the solid lines) with additional subclusters based on the forward and reverse primer (indicated by the dashed line), respectively. The plots on the right side show in each case the percentage of substitution errors that follow motif1, motif2 and motif3, respectively. In the case of *DS35* more than 80% of all errors succeed "GTG" (motif1) or "AGC" (motif2). And for half of the Fusion Golay data sets only three motifs (out of 64 possible ones) account for more than 50% of all substitution errors. On average the three most common motifs accounted for 34% of all substitution errors. This bias is even more pronounced for insertions, where more than 95% of all errors are preceded by the motif "AAT" for all Fusion Golay data sets. And on average 72% of all insertion errors follow the three most common motifs. For deletions we were able to connect on average 48% of all errors to three motifs. For the R2 reads the motifs account for an even larger fraction of errors in the case of substitutions and insertions: on average 44% of all substitution errors, 78% of all insertion errors and 46% of all deletion errors can be connected to three motifs. And more than 95% of all insertion errors in the Fusion Golay data sets were related to a single motif.

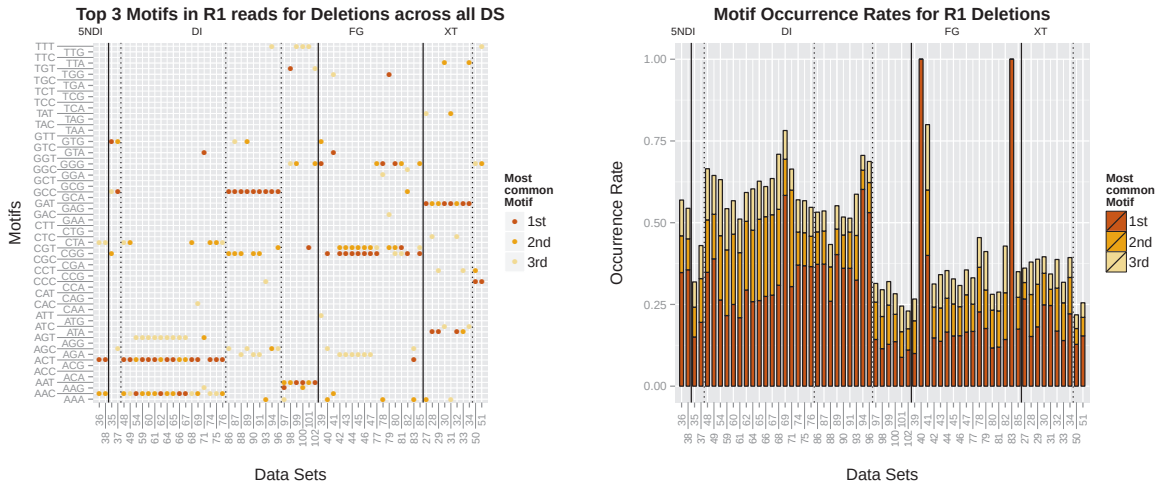
To determine the driving factors for the formation of these motifs, we used a permutation ANOVA with the Bray-Curtis distance analogously to the analysis of the error profiles. For the R1 and R2 substitutions the forward and reverse primer combination explains the largest fraction of the variance with 60% and 55%, respectively. The library preparation



(a) R1 Substitutions

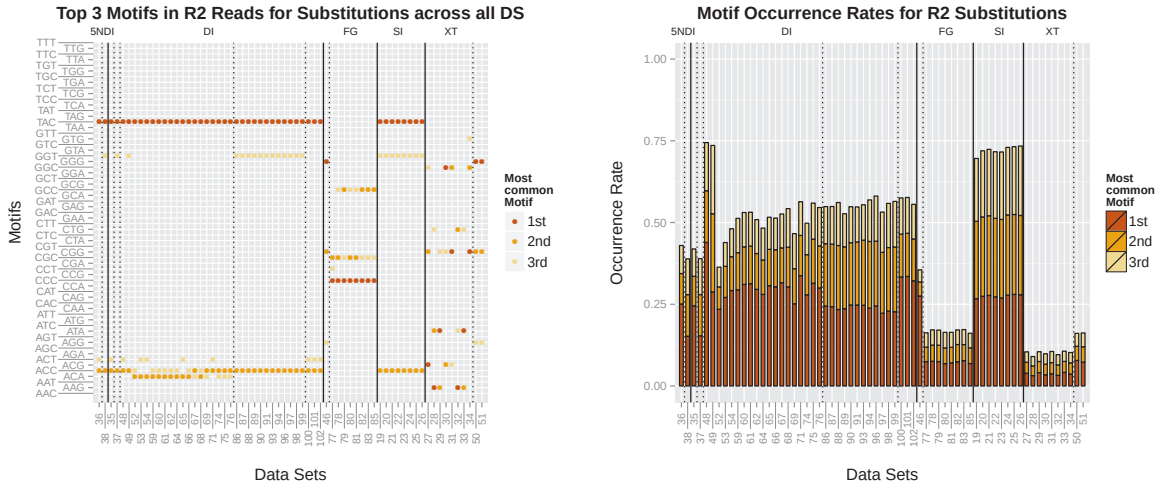


(b) R1 Insertions

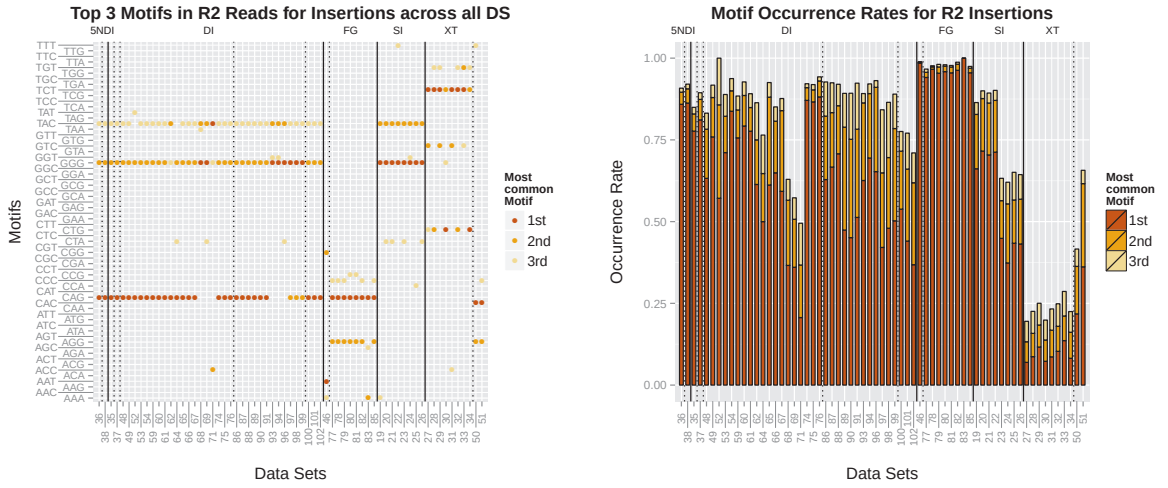


(c) R1 Deletions

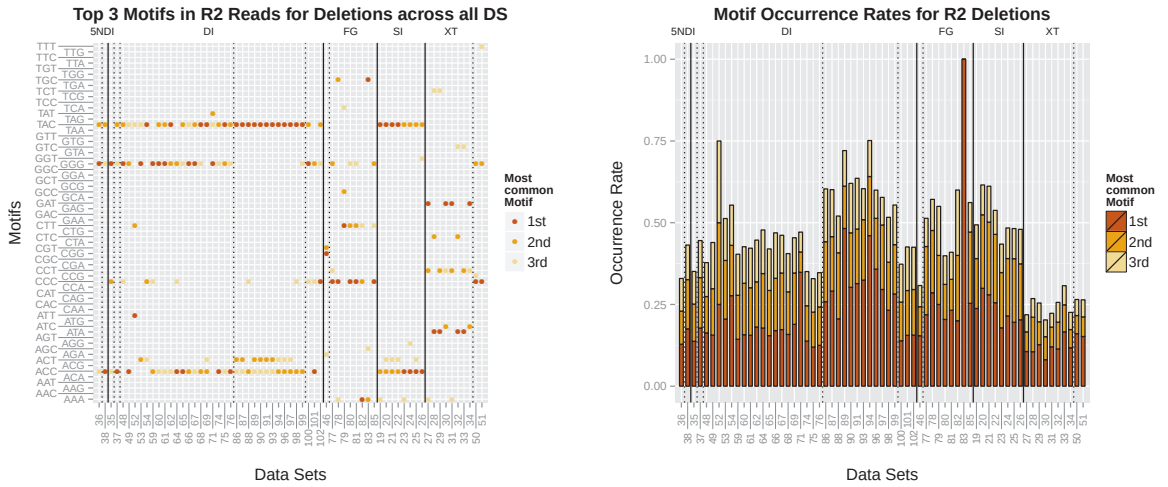
Figure 6.12: Motifs and motif occurrence rates for substitution, insertion and deletion errors in R1 reads: All 3mers preceding substitutions, insertions or deletions were recorded. For each data set the 3 most common motifs and the percentage of errors associated with those motifs are displayed. Solid lines separate the data sets according to library preparation methods and dashed lines further divide them according to different forward primers.



(a) R2 substitutions



(b) R2 insertions



(c) R2 deletions

Figure 6.13: Motifs and motif occurrence rates for substitution, insertion and deletion errors in R2 reads: We recorded all 3mers preceding a substitutions, insertions or deletions in R2 reads. For each data set the three most common motifs and the percentage of errors associated with those motifs are displayed. Solid lines separate the data sets according to library preparation methods and dashed lines further divide them according to different forward primers.

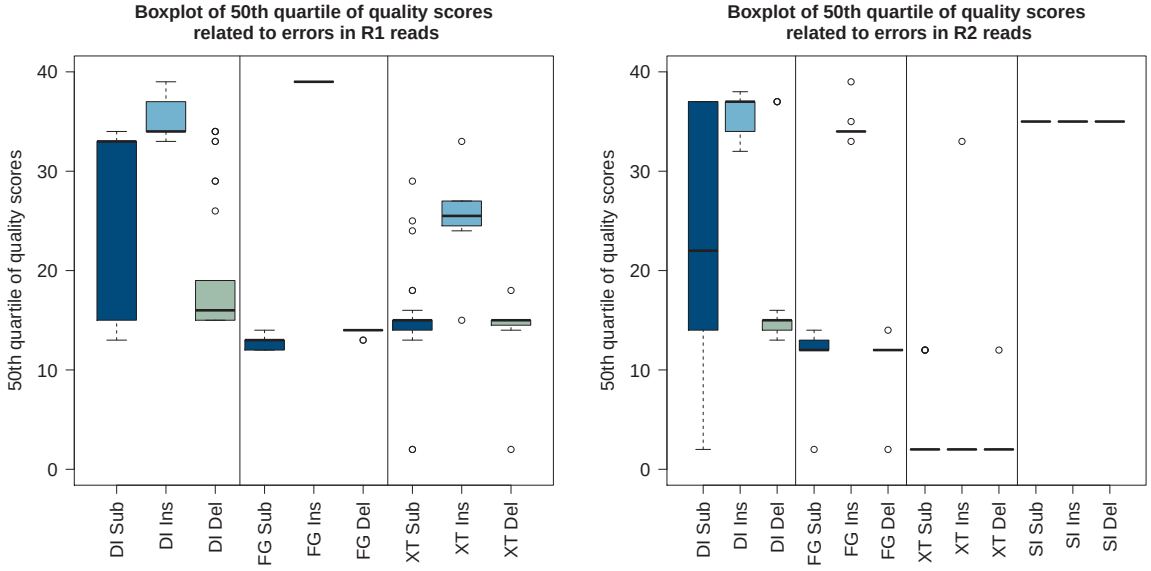
Table 6.7: Insertion and deletion rates of raw reads, after trimming the first 10bp and after additionally trimming the last 10bp. (Note: None of the R1 single index data sets contained ≥1,000 reads after alignment.)

	R1 Ins	R1 Del		R2 Ins	R2 Del
NexteraXT			**NexteraXT**		
RAW	0.000213	0.000128	RAW	0.000180	0.000090
Trim start	0.000221	0.000133	S10	0.000187	0.000093
Trim start & end	0.000083	0.000090	S10E10	0.000064	0.000050
Dual Index			**Dual Index**		
RAW	0.000509	0.000297	RAW	0.000026	0.000031
Trim start	0.000016	0.000009	Trim start	0.000024	0.000014
Trim start & end	0.000017	0.000009	Trim start & end	0.000024	0.000015
5N Dual Index			**5N Dual Index**		
RAW	0.001139	0.000529	RAW	0.000069	0.000024
Trim start	0.000046	0.000007	Trim start	0.000069	0.000014
Trim start & end	0.000048	0.000007	Trim start & end	0.000069	0.000014
Fusion Golay			**Fusion Golay**		
RAW	0.000112	0.000008	RAW	0.000109	0.000006
Trim start	0.000116	0.000008	Trim start	0.000114	0.000006
Trim start & end	0.000121	0.000008	Trim start & end	0.000117	0.000006
			Single Index		
			RAW	0.000009	0.000027
			Trim start	0.000007	0.000013
			Trim start & end	0.000007	0.000013

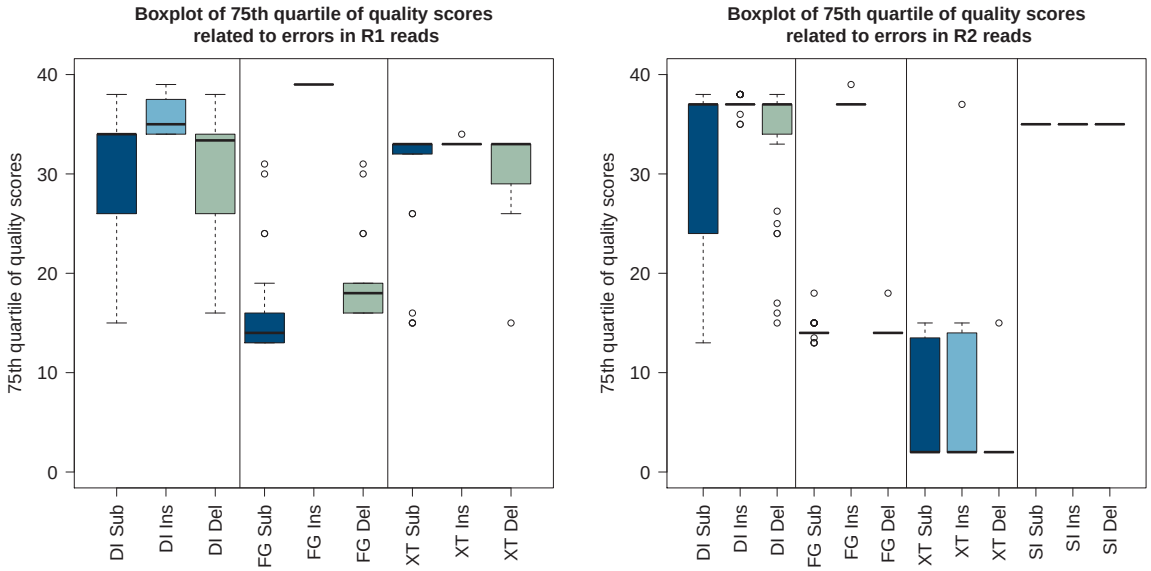
method explains an additional 16% and 26%, respectively. For insertions a total variance of 78% and 80%, respectively for R1 and R2 reads, can be explained by primers together with the library design and 78% for R1 deletions. The significant factors for R2 deletions are the run and the library design explaining a total of 25% of the variance.

Quality Scores

We investigated the quality scores associated with errors across all data sets. Figure 6.14a & b display the 50th and 75th quartile for all data sets, meaning 50% and 25%, respectively, of all quality scores associated with errors were above these values. The data sets are grouped by library preparation and the quality scores associated with substitutions, insertions and deletions are displayed separately. For the dual index data sets a large fraction of errors showed high quality scores. In particular, 50% of all R1 and R2 insertions were connected with quality scores of 32 and above for all data sets.



(a) 50th quartile



(b) 75th quartile

Figure 6.14: Overview of 50th and 75th quartile of quality scores associated with errors across all data sets. The results for the R1 reads are displayed on the left and the results for the R2 reads are on the right. Data sets were grouped by library preparation method (DI = dual index, SI = single index, FG = fusion golay, XT = NexteraXT) and substitution, insertion an deletion errors are displayed separately. Note, that for none of the single index data sets enough R1 reads aligned to construct meaningful quality profiles (threshold = 1,000 reads).

For the Fusion Golay data sets substitutions and deletions were well characterised by their quality scores but insertions showed very high quality scores. For the NexteraXT data sets we recorded high quality scores for $\geq 25\%$ of all errors. However, the 50th quartile was overall lower than for the dual index data sets and errors on R2 reads were well characterised. The single index data sets showed very high quality scores across all types of errors.

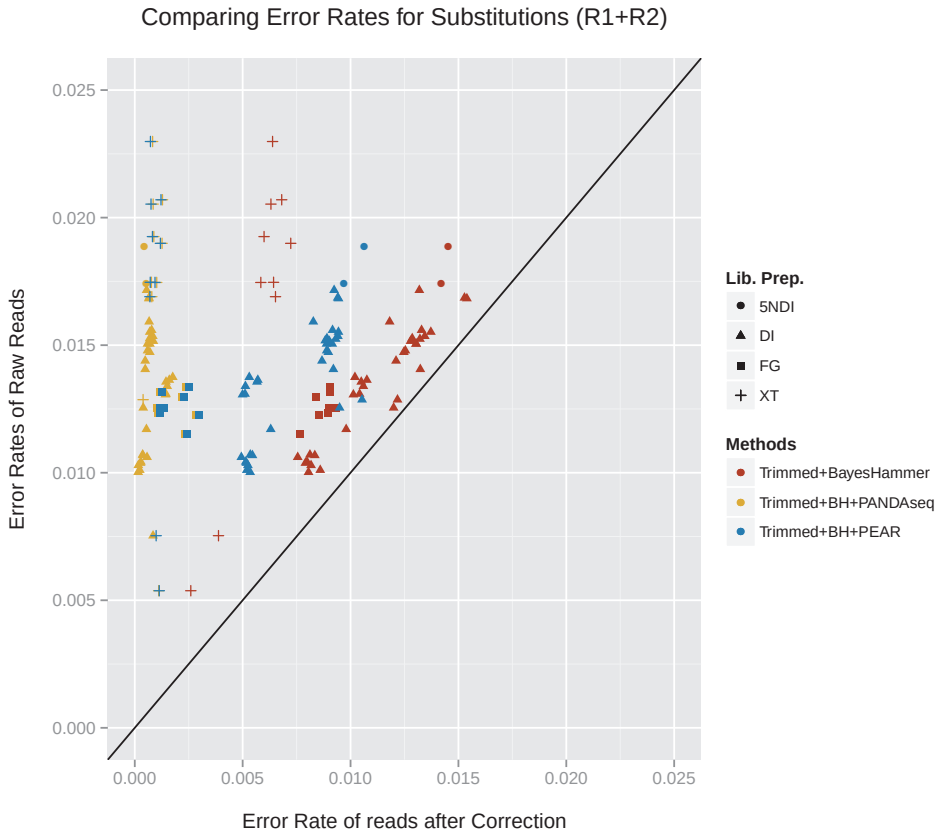


Figure 6.15: Comparison of error rates of the raw reads (R1+R2 rates) to different error corrections approaches including trimming+BayesHammer, overlapping reads with PANDAseq and overlapping reads with PEAR. Only data sets for which at least 1,000 reads aligned for all methods were included. Data sets not included: 19-26, 52+53 (not enough raw R1 reads aligned), 39-45+47 (not enough raw R2 reads aligned).

Error Correction

We compared different error removal techniques including trimming the start and end of the reads, trimming based on quality scores with sickle [11], error correction with BayesHammer [115], overlapping reads with PEAR (v0.9.1) [165] and PandaSeq (version 2.4, with a minimum overlap of 50bp for V4 data sets and 10bp for V3/V4 data sets) [106], and combinations of the different strategies.

Insertions and Deletions

Trimming the start and/or end of the reads proved to be an important step in removing indel errors. The average error rates are shown in Table 6.7 for raw reads, after trimming the first 10bp and after additionally trimming the last 10bp. For NexteraXT on average 61% of the insertion and 29% of the deletions can be removed from the R1 reads by trimming the last 10bp and 64% and 44%, respectively, for the R2 reads. For the DI data sets trimming the start of the read removed most indel errors. On average 96% of the R1 insertion and 97% of the R1 deletions as well as 8% and 55%, respectively,

Table 6.8: ANOVA results for motifs of substitutions, insertions and deletions. Analogously to the permutation ANOVA for the error profiles, we determined the model (F R Primer + Library Preparation + Template + Run + Taq + PCR Cycle R1+R2) for the adonis() function and used the Bray-Curtis distance for the three most common motifs for each data set.

R1 substitutions	Df	SumsOfSqs	MeanSqs	F.Model	R2	Pr(>F)
F R Primer	14	2.0719	0.147993	11.5151	0.60910	0.001
Library Preparation	3	0.5470	0.182343	14.1879	0.16082	0.001
Template	8	0.1518	0.018980	1.4768	0.04464	0.144
Run	1	0.0219	0.021884	1.7028	0.00643	0.193
Taq	1	0.0110	0.011044	0.8593	0.00325	0.443
PCR Cycle R1+R2	8	0.2509	0.031359	2.4400	0.07375	0.016
Residuals	27	0.3470	0.012852		0.10201	
Total	62	3.4016			1.00000	
R2 substitutions	Df	SumsOfSqs	MeanSqs	F.Model	R2	Pr(>F)
F R PRIMER	7	1.9774	0.282485	54.078	0.54752	0.001
Library Preparation	4	0.9238	0.230938	44.210	0.25578	0.001
Template	8	0.1669	0.020867	3.995	0.04622	0.001
Run	1	0.0023	0.002270	0.435	0.00063	0.598
Taq	1	0.0485	0.048532	9.291	0.01344	0.002
PCR Cycle R1+ R2	8	0.3098	0.038728	7.414	0.08579	0.001
Residuals	35	0.1828	0.005224		0.05062	
Total	64	3.6115			1.00000	
R1 insertions	Df	SumsOfSqs	MeanSqs	F.Model	R2	Pr(>F)
F R Primer	14	3.9771	0.284081	10.4352	0.66016	0.001
Library Preparation	3	0.7015	0.233835	8.5895	0.11644	0.001
Template	8	0.3420	0.042749	1.5703	0.05677	0.066
Run	1	0.0212	0.021200	0.7787	0.00352	0.495
Taq	1	0.0156	0.015576	0.5722	0.00259	0.598
PCR Cycle R1 + R2	8	0.2321	0.029010	1.0656	0.03852	0.397
Residuals	27	0.7350	0.027223		0.12201	
Total	62	6.0245			1.00000	
R2 insertions	Df	SumsOfSqs	MeanSqs	F.Model	R2	Pr(>F)
F R PRIMER	7	1.20091	0.171559	38.343	0.46427	0.001
Library Preparation	4	0.87932	0.219831	49.131	0.33994	0.001
Template	8	0.31297	0.039121	8.744	0.12099	0.001
Run	1	0.00000	-0.000003	-0.001	0.00000	0.981
Taq	1	0.00087	0.000872	0.195	0.00034	0.882
PCR Cycle R1 + R2	8	0.03602	0.004502	1.006	0.01392	0.440
Residuals	35	0.15660	0.004474			0.06054
Total	64	2.58670			1.00000	
R1 deletions	Df	SumsOfSqs	MeanSqs	F.Model	R2	Pr(>F)
F R Primer	14	3.9771	0.284081	10.4352	0.66016	0.001
Library Preparation	3	0.7015	0.233835	8.5895	0.11644	0.001
Template	8	0.3420	0.042749	1.5703	0.05677	0.099
Run	1	0.0212	0.021200	0.7787	0.00352	0.490
Taq	1	0.0156	0.015576	0.5722	0.00259	0.626
PCR Cycle R1+R2	8	0.2321	0.029010	1.0656	0.03852	0.423
Residuals	27	0.7350	0.027223		0.12201	
Total	62	6.0245			1.00000	
R2 deletions	Df	SumsOfSqs	MeanSqs	F.Model	R2	Pr(>F)
F R PRIMER	7	0.9838	0.14055	1.9134	0.16835	0.054
Library Preparation	4	0.9188	0.22971	3.1272	0.15724	0.009
Template	8	0.5735	0.07168	0.9759	0.09814	0.497
Run	1	0.4995	0.49952	6.8003	0.08548	0.001
Taq	1	0.0634	0.06340	0.8632	0.01085	0.412
PCR Cycle R1+R2	8	0.2337	0.02922	0.3978	0.04000	0.965
Residuals	35	2.5709	0.07345		0.43995	
Total	64	5.8437			1.00000	

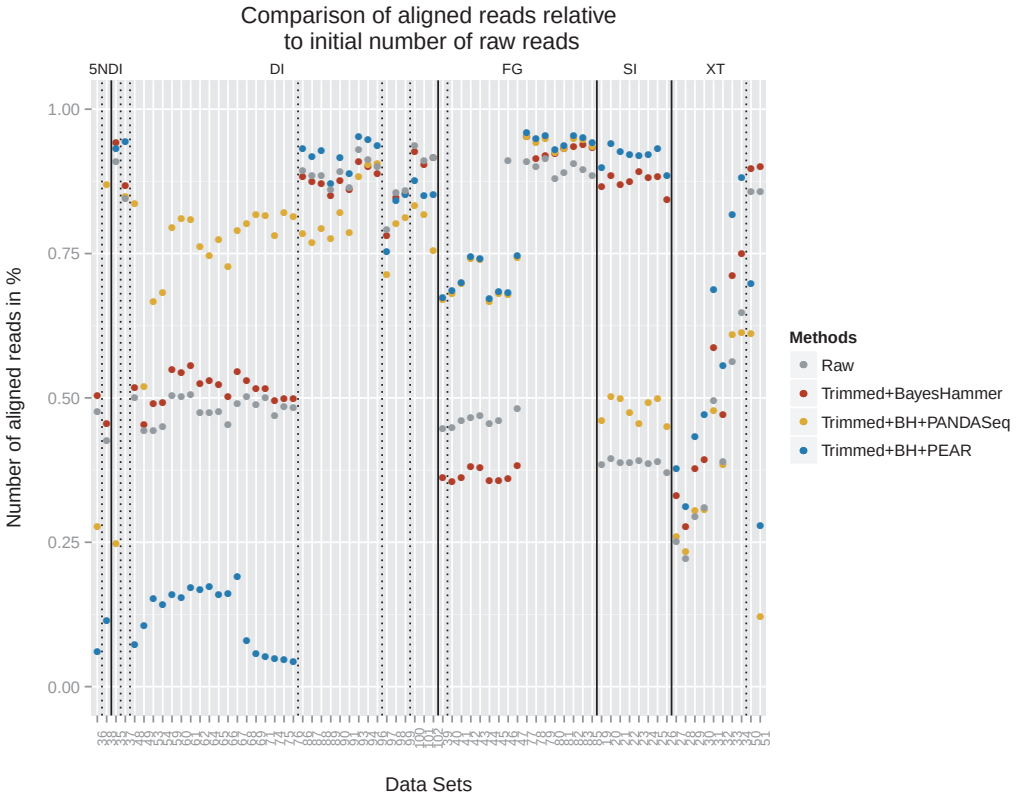


Figure 6.16: Comparison of the number of aligned reads relative to the initial number of raw reads. For the raw reads and reads processed with sickle plus BayesHammer, we summed the R1 and R2 rates. We also included trimming plus BayesHammer and overlapping with PANDAseq and PEAR, respectively, as those combination returned the lowest error rates. Data sets are separated according to library preparation methods (solid line) and primers (dashed line).

for the R2 reads. We observed similar results for the 5N Dual Index data sets. For the Fusion Golay the trimming showed almost no effect and for the SI data sets trimming the start of the R2 read removed about 50% of all deletions.

Substitutions

By overlapping the reads we were able to achieve further significant improvements with regards to the error rates. The best results in terms of error removal were achieved with a combination of quality trimming the reads with sickle, then applying BayesHammer for error correction and then overlapping the reads with PANDAseq. For the data sets displayed in Figure 6.15 the substitution error rates were reduced by 77%-98% with an average 93.2%. Figure 6.16 compares the percentage of aligned reads for the most successful approaches. After overlapping with PANDAseq between 12% and 95% of the reads aligned with an average of 69% across all data sets.

PEAR combined with read trimming and BayesHammer was able to reduce error rates by about 60% on average (range: 18%-97%). The number of aligned reads ranged from 4% to 96% and on average 61% of the read-pairs could be aligned after overlapping.

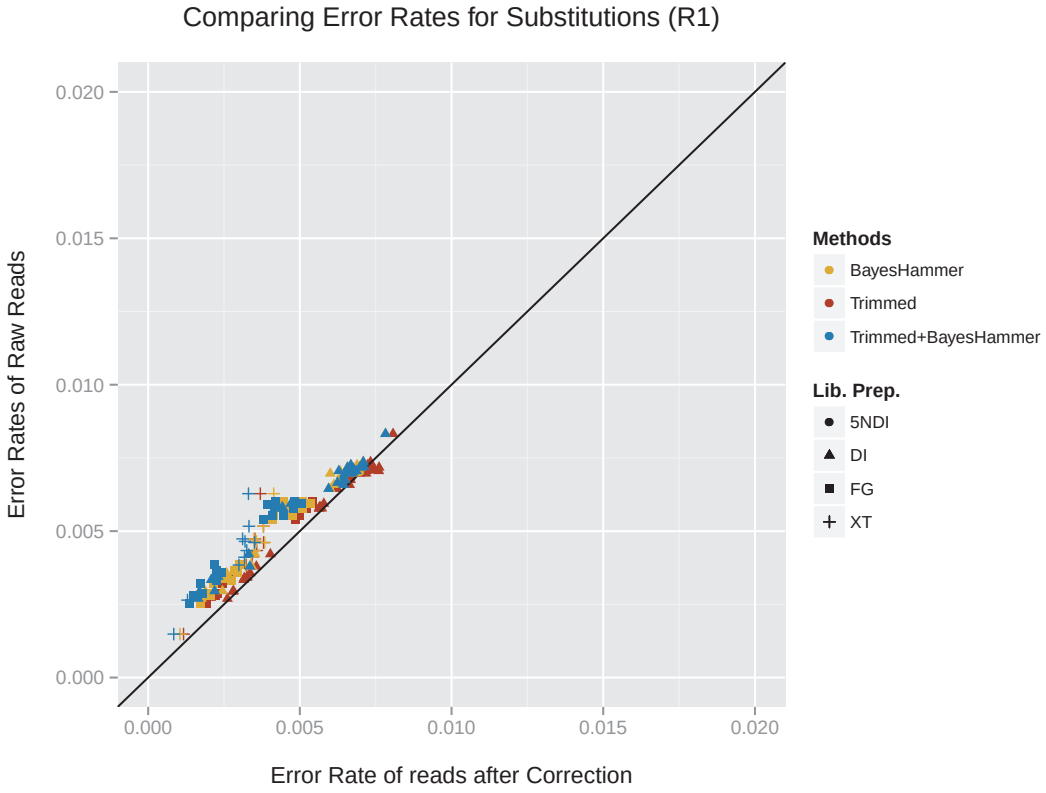


Figure 6.17: Comparison of error correction methods for R1 reads (keeping R1 and R2 reads separate). First trimming the reads and then applying BayesHammer yields slightly better results than each method on its own. We included only the data sets in the figure for which we had at least 1,000 aligned reads for all methods. Excluded data sets: 19-26 (no results on raw reads), 52 (no results across all methods), 53 (no results on raw reads).

PEAR encountered problems with the alignment of the DI and 5NDI data sets with high indels rates. For the Fusion Golay data sets PANDAseq and PEAR produced similar results in terms of aligned reads with lower rates for *DS 39-47* where the fraction of substitutions linked to the top three motifs was about 20% lower. It is also noticeable that for these data sets quality trimming combined with error correction lowered the number of aligned reads by about 15% on average. PANDAseq also encountered problems with the SI data set with lower rates than trimming+BayesHammer and trimming+BayesHammer+PEAR. The NexteraXT data sets produced very mixed results with regards to the percentage of aligned reads. The best results for the NexteraXT amplicon data sets were achieved by PEAR which aligned between 31% - 88%. Note that *DS50* and *DS51* were the full length 16S rRNA data sets (displayed in the last two columns of Figure 6.16). For *DS50* fragments between 500bp and 1,000bp were selected (average 590bp) and for *DS51* fragment size selection included sequences between 600 and 1,500bp (average 767bp). Although smaller fragments are preferentially sequenced we would expect (in particular for *DS51*) that only a small fraction of the reads can be overlapped.

If read-overlapping is not a possibility (i.e. if the average fragment size was larger than two times the read length) the best strategy for error removal was quality trimming followed by error correction with BayesHammer (see Figure 6.17+6.18). We recorded the most substantial improvement for the R2 reads of the NexteraXT data sets. The error rates slightly increased for some of the data sets after quality trimming. This is due to an increase in the number of aligned reads. When restricting the data sets to the reads that aligned prior to trimming/correcting the reads, the rates very slightly decreased.

6.6 Conclusion and Future Work

We have shown that the experimental design has a major impact on the error patterns of the sequencing data. To our knowledge, this was the first study on error profiles for the MiSeq and also the first time that a large range of experimental factors was tested in connection with error patterns. We used a complex mock community, to reflect the conditions encountered in real samples, as well as single species. A total of 73 data sets was used to show that the library preparation method together with the choice of primers causes an extensive bias towards certain motifs causing substitutions, insertions and deletions, respectively. This provides strong evidence that Illumina errors do not occur randomly.

The increased error rates that we observed towards the end of the reads are assumed to be due to accumulation of phasing and pre-phasing events throughout the sequencing process. Every time a molecule fails to elongate properly or advances too fast, the overall signal for the cluster suffers from interference. As the read length increases the cluster signal can get weaker due to an accumulation of these events resulting in higher error rates towards the end of the read [57]. This explains the gradual increase of errors that we observed in the position and nucleotide specific distributions in addition to the spikes caused by the motifs.

We demonstrated that A and C are more prone to substitution errors compared to G and T. Both A and C are identified through the red channel. This indicates a problem with either the red laser or the filter that is used to distinguish between the nucleotides. Also, the fluorescence emitted by A and C have the highest intensities. So any interference with the signal would result in an erroneous base call. G on the other side shows the lowest initial emission intensity. In particular for the Fusion Golay, the most common substituting nucleotide was a G which could also indicate signal disturbances.

Besides the library preparation method, we identified the forward and reverse primer as one of the major driving factors for the error profiles. The sources of errors described

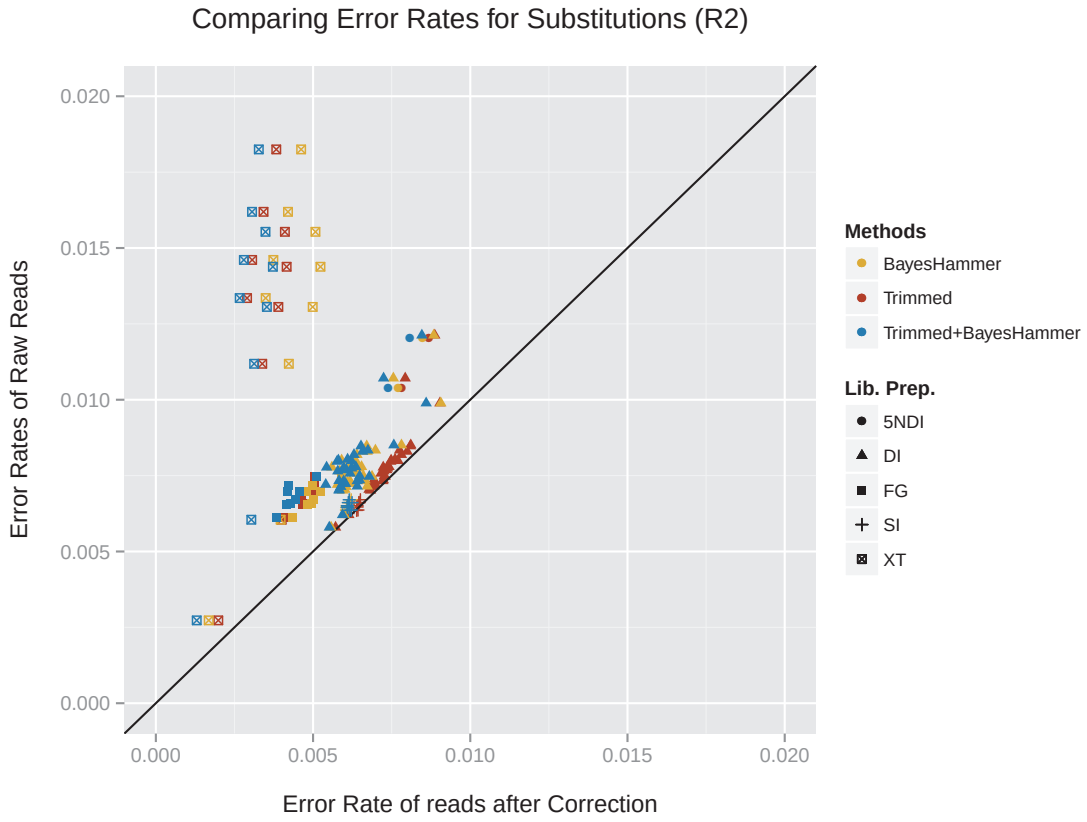


Figure 6.18: Comparison of error correction methods for R2 reads analogously (see Figure 6.17). Excluded data sets: 39-41+43 (no results across all methods), 42+44+45+47 (no results for raw reads, BayesHammer, trimming+BayesHammer), 46+52 (no results for trimming+BayesHammer).

above (i.e. phasing and pre-phasing, problems with red laser/filter) can be attributed to the actual sequencing process. In contrast to this, the library preparation method and the choice of primers are biases introduced prior to the sequencing process.

Figure 6.19 summarises the error rates for each library preparation method with regards to different error removal techniques. For none of the SI data sets could we align enough of the raw R1 reads. Overall the highest error rates were encountered for the NexteraXT data sets. At the same time trimming plus error correction achieved the best results on these data sets as well as additionally overlapping reads. PANDAseq achieved the best results across all library preparation methods. Overall the figure shows that error rates can be significantly reduced by combining various strategies for error removal.

Our quality score analysis showed that quality scores are of limited use for the identification of errors in amplicon sequencing data. Results differed for the various library preparation methods. Only substitutions and deletions in Fusion Golay data sets and errors in the R2 reads for the NexteraXT library preparation method were well characterised, while the majority of errors for all other library preparation methods was

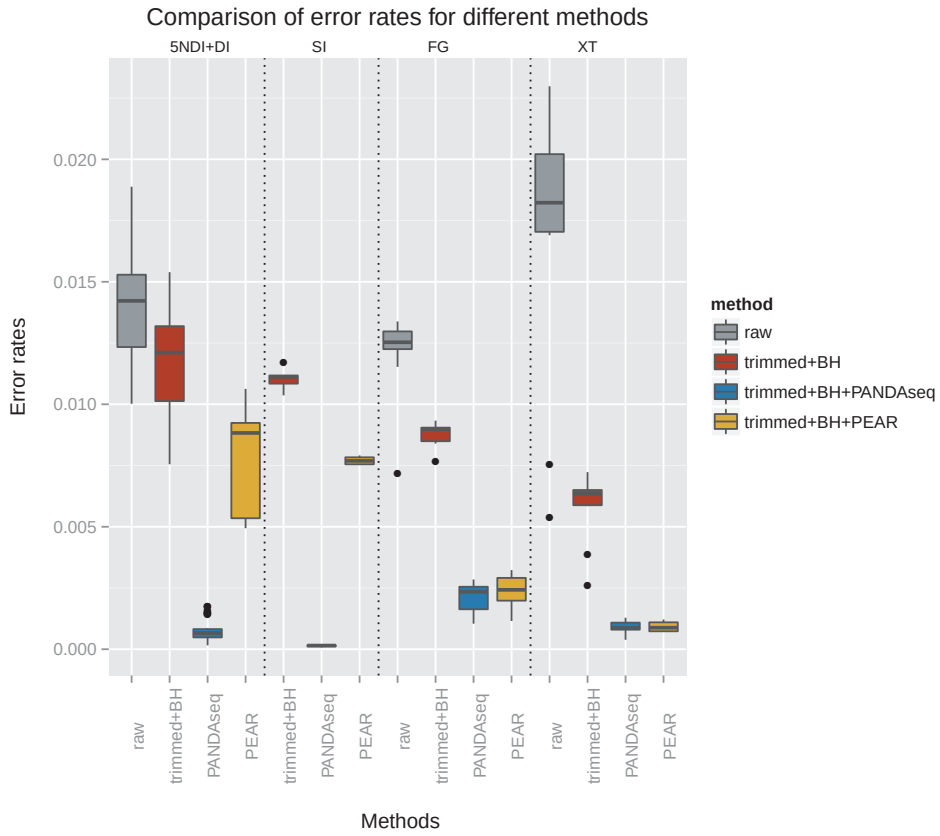


Figure 6.19: Range of average error rates (R1&R2) for the different library preparation methods (indicated on the upper x-axis). The grey bar plots shows the error rates for the raw reads, in red are the error rates after trimming and error correction, in blue and yellow are the error rates after additionally overlapping reads with PANDAseq and PEAR, respectively.

associated with high quality scores.

Choosing the most appropriate approach for a particular data set, depends on the individual hypothesis. Error rate reduction needs to be balanced with maximisation of aligned reads. For this we need to take the experimental factors into account. The number of aligned reads increased after trimming plus error correction for most data sets. Error rates were slightly reduced for the R1 reads and significantly reduced for the R2 reads. Thus quality trimming and error correcting reads is sensible for any kind of data sets. Additional trimming of the start of the reads seems advisable for the dual index data sets as they showed a huge increase of indels over the first 10bp in some cases. Overlapping reads with PANDAseq has proven most effective in removing errors but might reduce the number of aligned reads depending on the library preparation method and primers that were used. PEAR achieved higher numbers of aligned reads for some data sets but was not able to reduce errors to the same extent.

We observed similar results in terms of errors, motifs, read alignment and error removal potential for data sets with similar experimental design, i.e. same library preparation

method, forward and reverse primer and sequenced on the same run. Including a small mock community in a sequencing run could thus be used to determine the best strategy for removing errors from the sequencing data. We showed that PhiX is not suitable for this as the adapters used for PhiX represent a specific library preparation method that can differ from the one used for the actual sample. The purpose of PhiX is often to increase the data quality of low diversity samples and to optimise the cluster map generation. The same can be achieved by including a mock community with the added benefit of detailed information on the error patterns.

Sequencing error caused by motifs are more noticeable in amplicon data sets because of a higher degree of similarity between the sequences. They are represented by spikes in the position specific error distributions. We will subsequently extend our study to metagenomic data sets. This will allow us to separate sequencing errors from PCR errors and give further insight into the sources for different types of miscalls.

Systematic errors can cause major problems during the analysis of the sequencing data if programs assume that errors occur randomly. In particular for the identification of SNPs systematic errors will result in a high false positive rate and for diversity estimates systematic errors might result in a significant overestimation of the diversity in the sample. In order to identify these systematic errors it is important to infer individual error profiles for different sequencers, library preparation methods and sequencing types to handle miscalls. Illumina error rates are currently based on errors detected for the PhiX genome during the sequencing process. We showed that these error rates can greatly differ from the actual sample. Our approach offers the possibility to infer detailed error profiles for individual sequencing runs.

7 Illumina Error Profiles for Metagenomic Data Sets

7.1 Abstract

Metagenomics has emerged as a powerful approach for the analysis of microbial communities and has facilitated fundamental advances in microbial ecology. Entire microbial communities can be sequenced regardless of the ability to culture the organisms in the laboratory. The DNA of environmental samples is directly sequenced and therefore permits the characterisation of populations in specific environments. Metagenomics became widely available with the advent of next generation sequencing technologies and revealed the extensive diversity of microbial populations in different environments. Here, we extend our work on amplicon error profiles, presented in Chapter 6, to metagenomic data sets and included additional Illumina platforms as well as more low-input library preparation methods. We studied 41 metagenomic data sets sequenced on the MiSeq, HiSeq or Genome Analyzer II in combination with state-of-the-art library preparation methods including Nextera, NexteraXT, Parkinson and the standard library preparation method (TruSeq). The sequenced samples consist of diverse mock communities with different abundance distributions, as well as several single species samples. The reads from each sequencing run were aligned with Burrows-Wheeler Aligner (BWA) in paired-end mode. We extended our error profile software to process metagenomic data sets, computed the position and nucleotide specific error profiles for the individual data sets and recorded the occurrence of motifs (3mers preceding errors) for all types of errors. This revealed various biases associated with limitations due to the sequencing chemistry as well as biases associated with the Nextera technology. Furthermore, we tested different error removal methods to identify the best strategy for Illumina data sets. We showed that error trimming plus correction is capable of removing 66% of the substitution errors on average.

Original Contributions

Here, I present the first comprehensive study on error patterns in metagenomic data sets involving several Illumina sequencing platforms and I investigate the impact of experimental factors on error patterns. I designed the unbalanced mock community. The sample preparation and sequencing was performed by our collaborators in Liverpool, Dr Linda D'Amore and Prof. Neil Hall, and the alignment of the sequenced genomes was done by Dr Umer Z. Ijaz. My work comprises the entire subsequent bioinformatic analyses including the alignment of the data sets and the design and implementation of the algorithm to analyse the error patterns and motifs. Furthermore, I tested the different error removal techniques, evaluated the results and established the connection between experimental factors and biases.

7.2 Introduction

With an estimated number of 826 HiSeq sequencers, 500 Genome Analyzer (GA) II platforms and 183 MiSeq instruments, Illumina represents the dominant technology in the sequencing market [9]. Therefore, a better knowledge of systematic errors in Illumina sequencing data is urgently required to derive accurate and meaningful results. Here, we tested and compared these well established Illumina platforms. The Genome Analyzer was the first Solexa/Illumina sequencing platform and launched in 2006. Although popular for many years, it has now been outperformed by newer sequencers that offer longer reads and higher throughput. We included the platform to determine to which extent biases in the GA persist in the newer Illumina technologies. The HiSeq 2500 is currently one of the most popular platforms and can produce up to 8 billion paired-end reads of 2x125bp within 6 days (sequencing only) in high output mode. In rapid run mode the platform can achieve up to 1.2 billion paired-end reads of 2x150bp in 40 hours (cluster generation + sequencing). Illumina's benchtop sequencer, the MiSeq, produces the longest reads with up to 2x300bp. Cluster generation, sequencing and base calling takes approximately 55 hours and results in up to 50 million paired-end reads.

Here, we conducted a large study on metagenomic sequencing data across different Illumina platforms and library preparation methods to determine error patterns directly related to the sequencing process. This study builds on the work presented in Chapter 6, where we explored error profiles in amplicon data sets in connection with the MiSeq platform. Amplicon sequencing is an important tool to study microbial diversity and to identify the bacteria present in samples, however, it cannot reveal the functional capacities of the organisms. Metagenomics reveals information about the complete genomes of the organisms and offers insight into their functional abilities resulting in a much broader picture of the community. In contrast to metagenomics, amplicon sequencing requires several cycles of polymerase chain reaction (PCR) prior to the sequencing process. Therefore, not only sequencing but also PCR errors were encountered in the data. Our previous study identified library preparation and forward and reverse primer combination as the driving factors leading to distinct error patterns reflecting the mixture of sequencing and PCR related errors in the data. For metagenomics, the DNA is extracted and directly sequenced omitting the initial PCR amplification step. Note however, that the Nextera library preparation method involves a limited cycle PCR amplification step for the tagmentation of the fragments.

Several library preparation methods are available nowadays. The standard Illumina method for preparing sequencing libraries starts with the fragmentation of the template DNA by either sonication, nebulisation or shearing. This is followed by DNA repair and end polishing, plus ligation of platform specific adaptors. These adapters comprise

flow cell adapters, that allow the fragments to bind to the flow cell surface, sequencing primers, required for the synthesis of the template during the sequencing, and optional indices for multiplexing. Illumina's standard TruSeq sample preparation kit supports this workflow and is available with either 24 single indices or, alternatively, dual indices for sequencing up to 96 libraries simultaneously on a single run. For the TruSeq method 1μg of input DNA is recommended [79]. However, most of the input material is lost during the library preparation and the method is time-consuming and labor intensive. Recently, a new technology was developed that combines these steps into one reaction. The Nextera transposome technology allows simultaneous fragmentation and tagmentation by using an adapted *in vitro* transposition. This method requires less input DNA and offers shorter preparation times [149]. The transposome consist of the transposase and a transposon complex with engineered transposon ends. The transposase catalyses the insertion of excised transposons into the template DNA resulting in random double stranded breaks. During this process the 3' end of the transposon strands, including a unique adapter sequence, gets attached to the 5' end of the target DNA. After the template DNA is labeled at the 5' end, a complementary tag is added to the 3' end using a polymerase extension. Platform specific sequencing adapters can additionally be added, and the sample can be enriched and bar-coded with the standard Illumina indices using limited-cycle PCR. Libraries can be prepared in 90 minutes and are optimised for 50ng of input DNA. Further, the low input NexteraXT kit enables libraries prepared with only 1ng of input DNA.

Parkinson et al. introduced another low input library preparation method promising libraries from picogram quantities [120] by using a modified transposome-mediated fragmentation technique. Their results indicate, that a similar coverage can be achieved with 20pg compared to the coverage obtained from a standard library prepared with 1ug of DNA.

Low input library preparation methods present a great advancement for DNA sequencing as large quantities of input material are not always available. These methods make sequencing accessible to a broader range of research areas, including clinical and environmental studies as well as forensics. We analysed the errors and biases associated with these methods to test their capabilities and compare them to the standard library preparation method.

7.3 Materials and Methods

Library preparation methods

The standard Illumina indices were used for all libraries. For the standard and Parkinson

libraries multiplexing was implemented with single indices. For Nextera and NexteraXT dual indexing was employed. DNA quantities of 250ng and 500ng were tested for the standard library. Most Nextera libraries were prepared with 50ng of starting DNA. In addition, two libraries with 0.5ng were tested. The starting amount for the NexteraXT libraries was 1ng for all data sets. For the Parkinson libraries 0.5 or 0.05ng were used. After fragmenting the DNA a range of 600 to 900bp was selected for all data sets with the Pippin Prep.

Samples

The samples for sequencing included a diverse mock community consisting of 49 bacterial and 10 archaeal genomes. For the first mock community even amounts of genomic DNA were combined (balanced mock), for the second community the genomic DNA was mixed according to a log-normal distribution (unbalanced mock). For further details see Chapter 6. We also sequenced several single species samples including *Burkholderia xenovorans (LB400)*, *Desulfovibrio desulfuricans subsp. desulfuricans str. ATCC 27774*, *Enterococcus faecalis V583*, *Nanoarchaeum equitans Kin4-M*, *Rhodospirillum rubrum ATCC 11170*, *Thermus thermophilus HB8* and *Treponema vincentii I.*

Platforms

For the metagenomic error profiles we tested the Genome Analyzer II, the HiSeq and the MiSeq. The samples sequenced on the MiSeq included three mock community samples and nine single species samples. The samples were prepared with Nextera, NexteraXT or the standard library preparation method. With higher throughput the HiSeq and GAII are more commonly used for diverse data sets where a high coverage is required. Therefore, we mainly tested the mock communities on these platforms. On the GAII we sequenced 13 mock community samples. These data sets covered the standard, Nextera and Parkinson library preparation method with varying amounts of starting DNA. The HiSeq data sets include 14 mock community samples and two single species samples and were prepared with the Nextera, NexteraXT and standard library preparation method. The samples were distributed across four runs on two MiSeq sequencers, two HiSeq runs and three GAII runs. Tables 7.1 and 7.2 provide an overview of the different parameters for each test data set. For very large data sets the reads were subsampled to one million for the MiSeq data sets, four million for the HiSeq data sets and three million for GAII data sets for the subsequent analysis.

Reference database

The mock community was part of a study by Shakya et al. [145] which provided the majority of the genome sequences for our reference database. However, four organisms exhibited poor coverage and were therefore resequenced: *Desulfovibrio desulfuricans*

Table 7.1: Overview of the experimental design for the metagenomic data sets (1). Library preparation methods: Nextera (N), NexteraXT (XT), Parkinson Low Input (P), Standard TruSeq (S); Templates: *Burkholderia xenovorans (LB400)* (BX), *Desulfovibrio desulfuricans subsp. desulfuricans str. ATCC 27774* (DSV), *Enterococcus faecalis V583* (EF), *Nanoarchaeum equitans Kin4-M* (NE), *Rhodospirillum rubrum ATCC 11170* (RHO), *Thermus thermophilus HB8* (TT), *Treponema vincentii I* (TV), balanced mock community (MB), unbalanced mock community (MUB);

Platform	Meta ID	Library	Run	Machine	Input ng	Template	Read length
MiSeq	55	N	2	Miseq2	50	MUB	2x250bp
	59	S	3	Miseq2	250	MB	2x250bp
	60	S	3	Miseq2	250	MUB	2x250bp
	76	XT	4	Miseq2	1	BX	2x250bp
	77	XT	4	Miseq2	1	DSV	2x250bp
	78	XT	4	Miseq2	1	EF	2x250bp
	80	XT	4	Miseq2	1	TT	2x250bp
	81	N	4	Miseq2	50	NE	2x250bp
	82	N	4	Miseq2	50	TV	2x250bp
	102	XT	5	Miseq2	1	BX	2x250bp
	103	XT	5	Miseq2	1	RHO	2x250bp
	104	XT	5	Miseq2	1	TT	2x250bp
GAII	4	S	1	GAII1	500	MB	2x100bp
	5	P	2	GAII1	0.5	MB	2x100bp
	6	P	2	GAII1	0.05	MB	2x100bp
	7	P	2	GAII1	0.05	MB	2x100bp
	8	S	2	GAII1	500	MUB	2x100bp
	10	N	3	GAII1	0.5	MUB	2x100bp
	11	N	3	GAII1	50	MUB	2x100bp
	12	N	3	GAII1	50	MB	2x100bp
	31	N	3	GAII1	0.5	MB	2x100bp
	32	P	2	GAII1	0.5	MB	2x100bp
	33	P	2	GAII1	0.5	MB	2x100bp
	34	P	2	GAII1	0.05	MB	2x100bp
	35	P	2	GAII1	0.05	MB	2x100bp

desulfuricans ATCC 27774, *Enterococcus faecalis V583*, *Nanoarchaeum equitans Kin4-M* and *Treponema vincentii I*. The respective reads were assembled with VelvetOptimiser [25] and SPAdes [38]. Contigs of at least 1,000bp were concatenated and included in the reference database.

Algorithm for computing the error profiles

The reads were aligned with the latest Burrows-Wheeler Aligner (BWA) algorithm in paired-end mode: BWA-MEM (version 0.7.9a) [96]. We used the -M option to mark shorter split hits as secondary alignments. All secondary alignments and unmapped

Table 7.2: Overview of the experimental design for the metagenomic data sets (2).

Platform	Meta ID	Library	Run	Machine	Input ng	Template	Read length
HiSeq	15	N	1	Hiseq1	50	MB	2x100bp
	16	N	1	Hiseq1	50	MUB	2x100bp
	21	XT	1	Hiseq1	1	MB	2x100bp
	22	XT	1	Hiseq1	1	MB	2x100bp
	23	XT	1	Hiseq1	1	MUB	2x100bp
	24	XT	1	Hiseq1	1	MUB	2x100bp
	25	XT	1	Hiseq1	1	DSV	2x100bp
	26	XT	1	Hiseq1	1	RHO	2x100bp
	63	XT	2	Hiseq1	1	MB	2x100bp
	64	XT	2	Hiseq1	1	MB	2x100bp
	65	XT	2	Hiseq1	1	MUB	2x100bp
	66	XT	2	Hiseq1	1	MUB	2x100bp
	70	N	2	Hiseq1	50	MB	2x100bp
	71	N	2	Hiseq1	50	MUB	2x100bp
	74	S	2	Hiseq1	250	MB	2x100bp
	75	S	2	Hiseq1	250	MUB	2x100bp

reads were discarded. Our previous study on amplicon error profiles showed that the R1 and R2 reads exhibited distinct error patterns. The paired-end alignment strategy was used as it offers higher accuracy, but for the subsequent analysis the aligned reads were again separated into R1 and R2 reads based on the FLAG field of the Sequence Alignment/Map (SAM) files. The FLAG field also specifies if the read originates from the plus or minus strand which was taken into consideration. Next, the MD tag was generated with SAMtools (version 0.1.18 and 0.1.19) [97]. We then applied the same algorithm as described in Chapter 6 to compute the position and nucleotide specific error profiles and motifs.

7.4 Results

We start this section by discussing the nucleotide and position specific error profiles for one HiSeq data set (*DS70*) in detail and we investigate the occurrence of nucleotides across all read positions. Next, we compare the quality score associated with the different types of errors for R1 and R2 reads. The detailed analysis of a GAII and MiSeq data set can be found in Appendix B. For the overall comparison of all 41 data sets, we analyse the error rates with regards to the original nucleotide and substituting nucleotide. This is followed by a comparison of the motifs identified for each of the data sets. Furthermore, we examine the ability of the quality scores to predict different types of errors. This section concludes with an outline of the capacities of different error removal approaches across platforms and library preparation methods.

Substitutions:

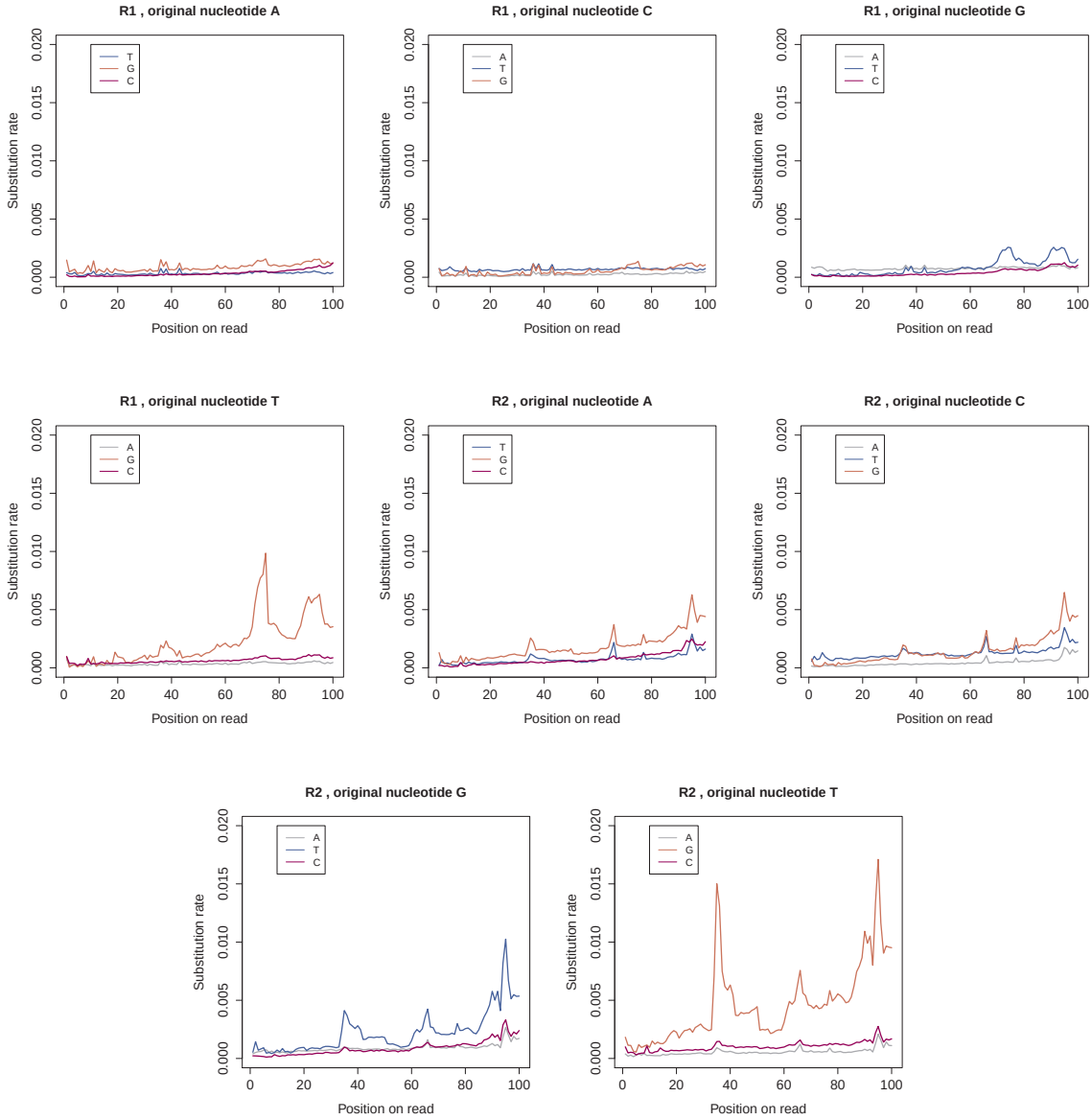
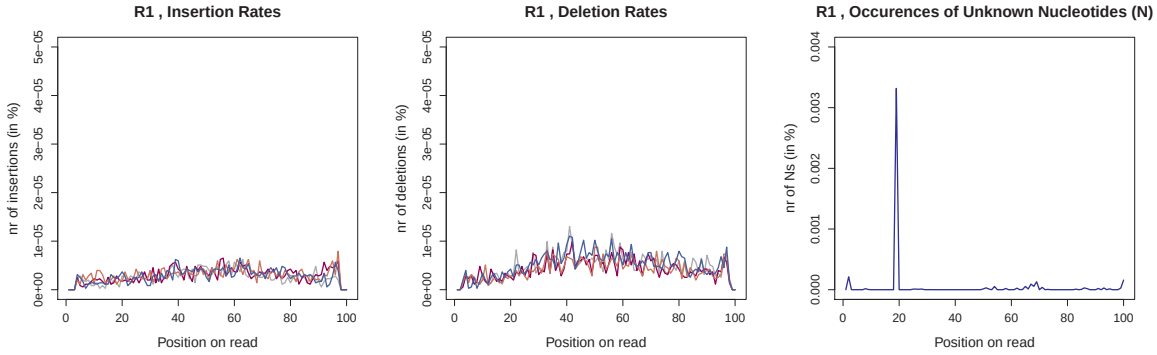


Figure 7.1: Nucleotide specific substitution error profiles for metagenomic data set *DS70*: Each graph shows the substitution rates for a specific original nucleotide and the colours indicate the substituting nucleotide. The first four graphs show the R1 profiles and the last four graphs show the R2 profiles.

Detailed error and quality profiles for data set *DS70*

Here, we present the detailed error profiles for one of the HiSeq data sets, where the library for the balanced mock community was prepared with the Nextera kit using 50ng of input DNA. The substitution profiles of the metagenomic data set are presented in Figure 7.1. The graphs highlight the tendency of substitutions to cluster together. We hypothesise that this effect is related to the polymerase and the nature of the ddNTPs, as will be further outlined in the discussion of this chapter. This effect was not visible in the

R1 Profiles for Insertions, Deletions and Unknown Nucleotides (Ns):



R2 Profiles for Insertions, Deletions and Unknown Nucleotides (Ns):

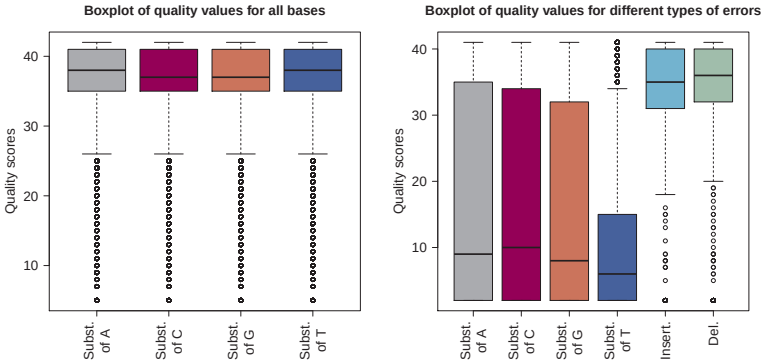
Figure 7.2: Error profiles for insertions, deletions and unknown nucleotides (Ns): The first three graphs show the R1 error profiles. For insertions, the colour identifies the inserted nucleotide and for deletions the colour refers to the type of nucleotide that was deleted. The lower three graphs display the error profiles for the R2 reads, respectively.

amplicon profiles as a non-uniform nucleotide distribution is encountered in amplicons, i.e. not every nucleotide occurs at every position.

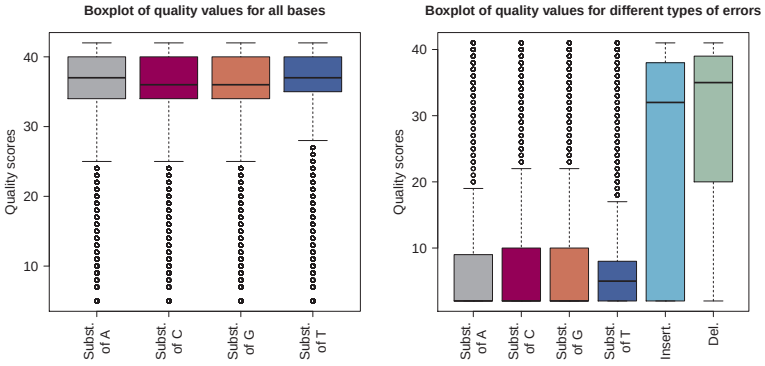
Error profiles

In contrast to the amplicon error profiles presented in Chapter 6, the metagenomic error profiles did not show sharp increases of error rates at individual positions. These spikes in the amplicon data sets are due to the motif-based nature of the errors. As amplicons cover the same region the effect of these motifs is visible as spikes in the position specific error profiles. Furthermore, the graphs in Figure 7.1 revealed a clear bias in terms of the substituting nucleotide. G seems to be preferentially incorporated if an A, C or T is sequenced and if G is sequenced a T is falsely incorporated for the majority of substitutions.

The overall error rates of this data set were very low: a rate of 0.0021 was recorded for the R1 reads and 0.0042 for the R2 reads. However, the errors are not evenly distributed



(a) R1 quality profiles

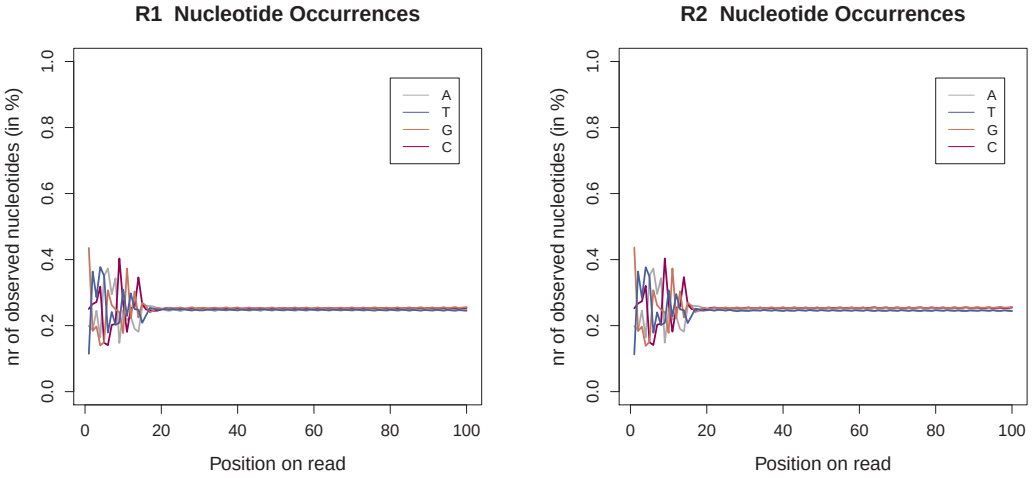


(b) R2 quality profiles

Figure 7.3: Quality profiles for R1 and R2 reads: The box plots in the first column display the distribution of quality scores for all bases. The second column shows the distribution of quality scores associated with errors.

across positions and nucleotides, creating a significant bias. The nucleotides A and C show the lowest error rates with 0.0004 in the R1 reads and 0.0008 in the R2 reads for both nucleotides. G shows a slightly higher average error rate of 0.0005 in the R1 reads and 0.0010 in the R2 reads. T exhibited the highest average error rate with 0.0008 and 0.0015, respectively for R1 and R2. Further, much higher error rates were observed at individual positions. For example, at read position 35 in the R2 reads substitutions of T were observed in 1.74% of all reads (rate 0.0174). Overall, error rates increased towards the end of the read and errors are twice as likely to occur in R2 reads.

The insertion and deletion profiles as well as the distribution of unknown nucleotides are displayed in Figure 7.2. Indel errors occur at a much lower rate compared to substitutions: Rates of $2.8 \cdot 10^{-6}$ for R1 insertions and $5.1 \cdot 10^{-6}$ for R1 deletions were observed. For R2 we observed rates of $3.5 \cdot 10^{-6}$ and $4.9 \cdot 10^{-6}$, respectively for insertions and deletions. Indel errors were more evenly distributed across the length of the read, with a small increase for the last 10bp. Deletions of all four nucleotides were observed at comparable rates and, similarly, insertions rates were comparable across all nucleotides



(a) Nucleotide distribution in R1 reads

(b) Nucleotide distribution in R2 reads

Figure 7.4: Comparison of occurrence rates of the four nucleotides across the reads for data set *DS70*. The library for this data set was prepared with the Nextera kit and sequenced on the HiSeq.

with the exception of G insertions, where marginally higher rates were recorded.

Quality scores

Next, we analysed the quality scores for the different error types. Overall, the data sets displayed very high quality scores with an average of 37 and 35 for R1 and R2, respectively. A large part of the substitution errors were well characterised: 69% of the R1 substitutions and 86% of the R2 substitutions showed quality scores below 20. For insertions and deletions, on the other hand, the quality scores were meaningless as the majority of indel errors were assigned a very high quality score. Only 19% of the R1 and 35% of the R2 indel errors showed quality scores below 20.

Nucleotide distribution

For all Nextera and NexteraXT libraries we observed uneven nucleotide distributions at the start of the reads. These library preparation methods rely on the transposome technology, where the transposase, is used to simultaneously fragment and tagment the template DNA. For most Nextera and NexteraXT data sets these fluctuations effected approximately the first 20bp of the R1 and R2 reads. Figure 7.4 displays the results for data set *DS70*. For the Parkinson libraries, which use an adapted version of the Nextera technology, similar fluctuations were observed. Here, these fluctuations seem to effect a larger part of the start of the read affecting the first 30bp for most of the data sets. The observed fluctuations were also more extreme.

Table 7.3: Average substitution rates for GAII, HiSeq and MiSeq for the metagenomic data sets, split according to the original nucleotide.

Platform	R1/R2	A	C	G	T
GAII	R1	0.0015	0.0010	0.0008	0.0018
GAII	R2	0.0035	0.0029	0.0019	0.0026
HiSeq	R1	0.0004	0.0004	0.0004	0.0008
HiSeq	R2	0.0007	0.0007	0.0007	0.0012
MiSeq	R1	0.0011	0.0009	0.0011	0.0011
MiSeq	R2	0.0023	0.0015	0.0016	0.0024

Overall comparison of error and quality profiles

In the following we compare the error rates as well as biases with regards to the substituting nucleotide across all data sets. Furthermore, we examine the motifs associated with substitution and indel errors and examine the reliability of quality scores.

Substitution rates

The overall error rates for all data sets are displayed in Figure 7.5. The upper two graphs compare the substitution rates between platforms and library preparation method and show the differences between the R1 and R2 reads. The Genome Analyzer II showed the highest error rates with average substitution rates of 0.0051 for R1 reads and 0.0109 for R2 reads. The HiSeq data sets showed the lowest substitution rates of all three platforms with average rates of 0.0021 for R1 and 0.0033 for R2. The MiSeq, Illumina's benchtop sequencer, showed lower rates compared to the GAII but higher rates than the HiSeq platform. Note however, that the MiSeq is able to provide the longest reads. We recorded average substitution rates in the MiSeq data sets of 0.0043 and 0.0077 for R1 and R2, respectively. The bar plots show the proportion of errors associated with the four different types of original nucleotides. For both the GAII and HiSeq, the highest substitution rates were observed for T and rates roughly doubled for the R2 reads. Additionally, the rates confirm that improvements for the HiSeq not only resulted in lower rates, but also in more similar substitution rates for A, C and G, however, the bias for T remains. For the MiSeq, R1 error rates were comparable for all for nucleotides. For R2 substitutions, higher rates were observed for A and T. (For further details see Table 7.3.) Overall, the largest fluctuation in substitution error rates were recorded for the MiSeq.

Indel rates

Insertion and deletion rates (displayed in Figure 7.5) are generally very low. A sharp increase in insertions was observed for HiSeq data set *DS26* as well as the MiSeq data

Figure 7.5: Comparison of error rates for all metagenomic data sets. The upper graphs indicate the proportion of substitutions of A, C, G and T for each data set, respectively. The two graphs in the middle show the proportion of inserted A, C, G and T nucleotides and the lower graphs show the proportion of deletions associated with each of the four nucleotides. Data sets are grouped by sequencing platform (solid lines) and library preparation method (dashed lines).

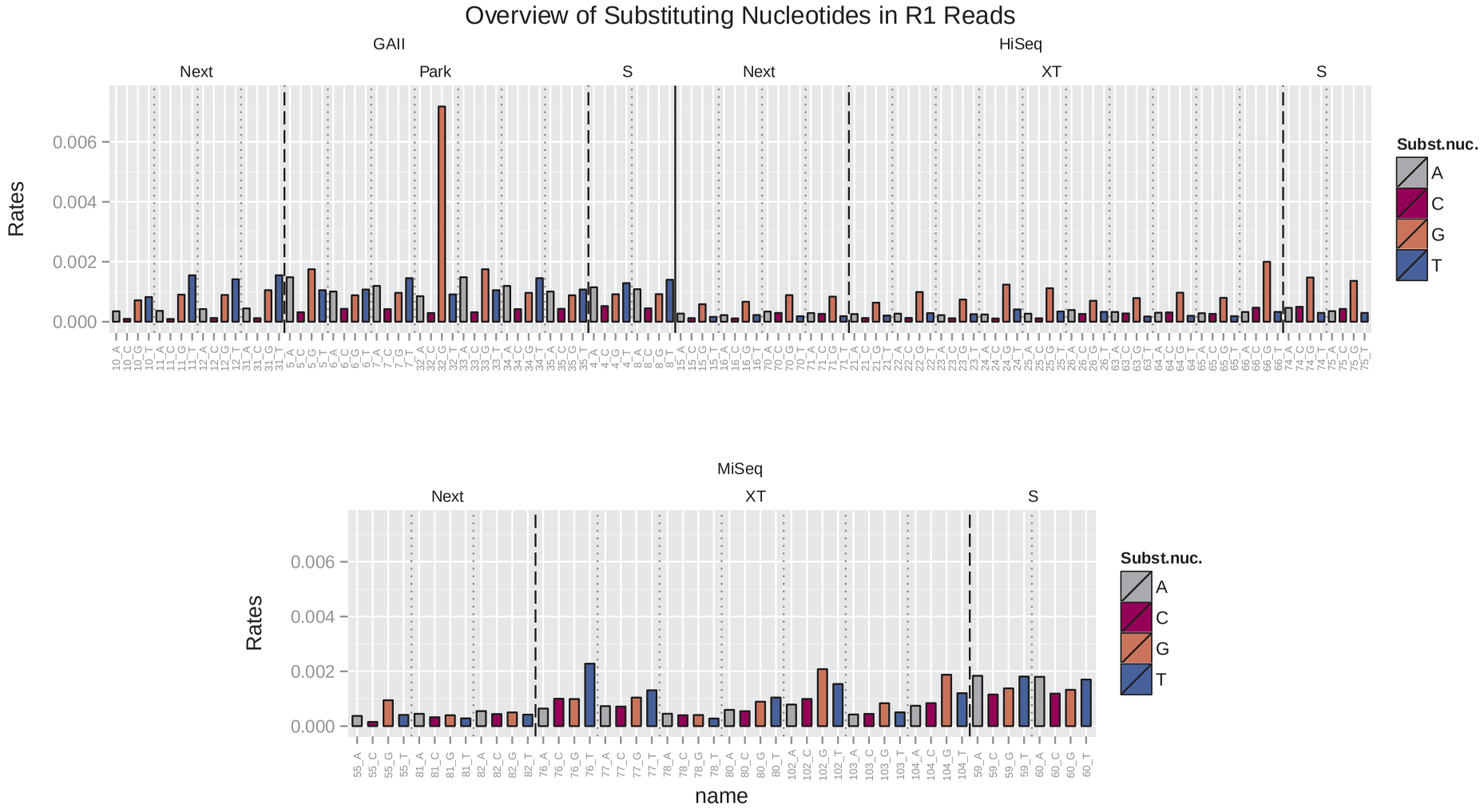


Figure 7.6: Comparison of substituting nucleotides in R1 reads: the upper plot shows the GAII and HiSeq, the lower plot shows the MiSeq data sets.

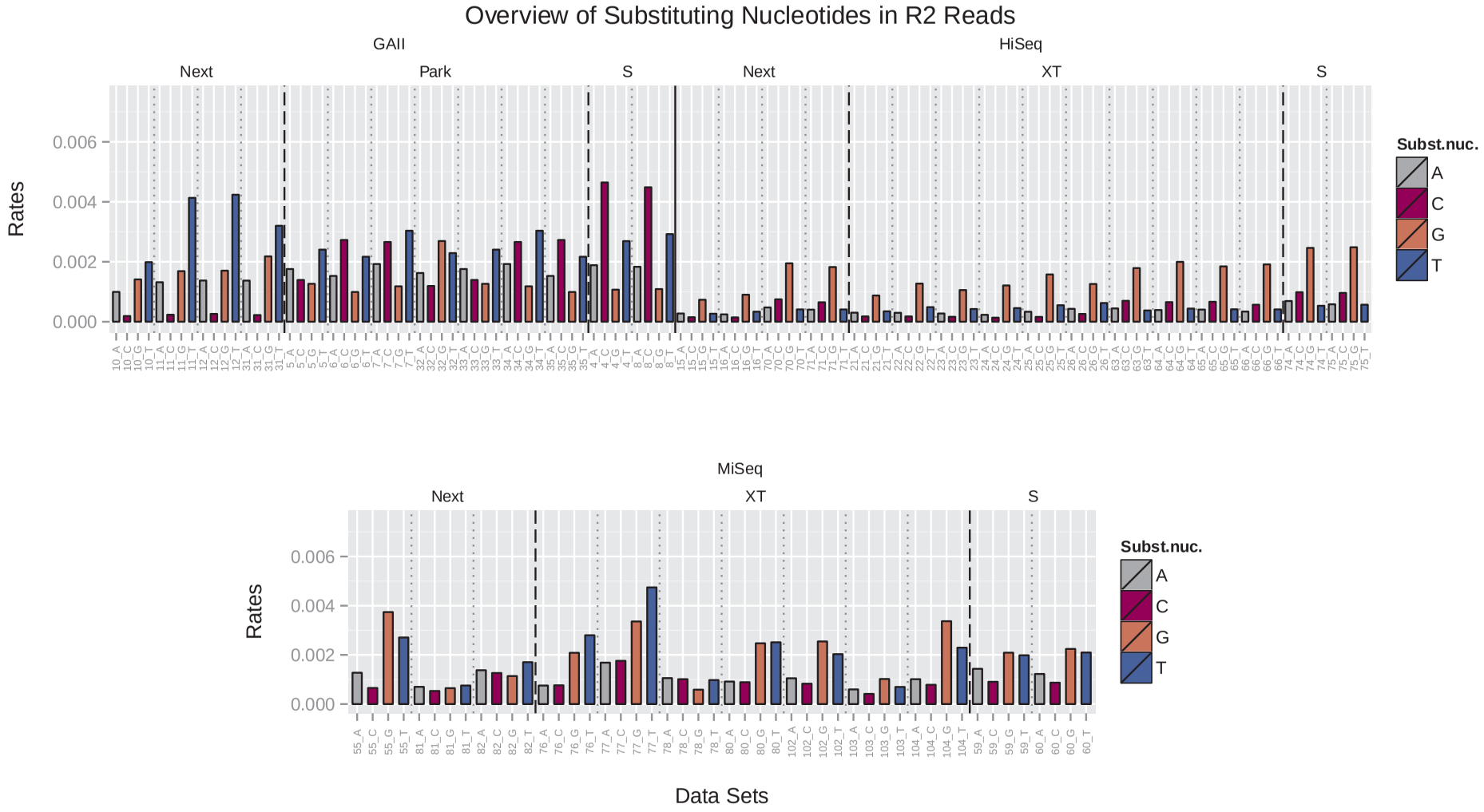


Figure 7.7: Comparison of substituting nucleotides in R2 reads: the upper plot shows the GAII and HiSeq, the lower plot shows the MiSeq data sets.

sets that were prepared with the standard library method. For this HiSeq data set, two tight peaks in the position specific insertion rates were recorded and for the two MiSeq data sets insertions accumulated over ≈25 bp in the centre of the reads.

Substituting nucleotide

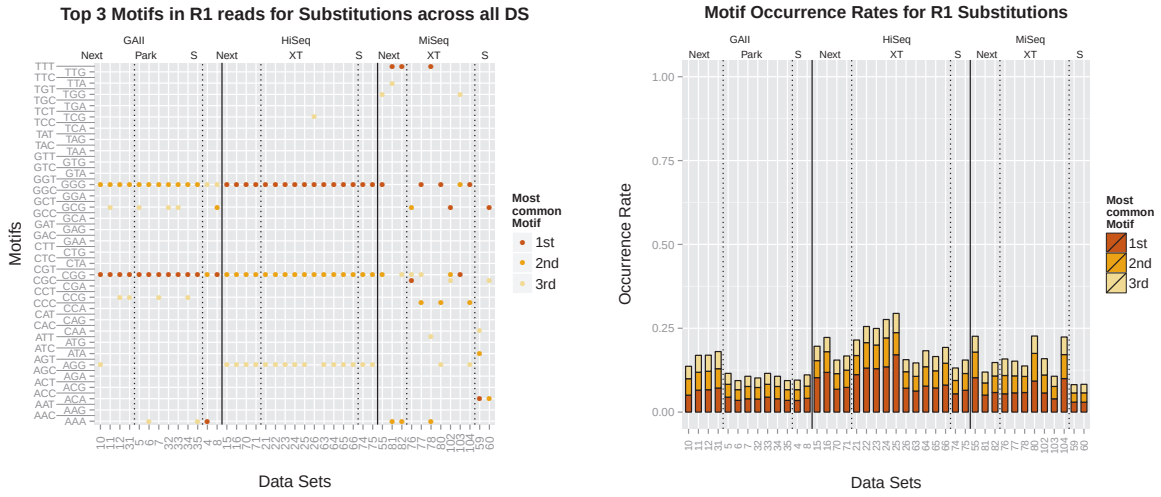
In addition to recording the substituted nucleotides, we also analysed the substituting nucleotides that were falsely incorporated. Figure 7.6 and 7.7 show the results for the R1 and R2 reads, respectively. For the GAII and HiSeq, C was rarely the substituting nucleotide in the R1 reads. A bias towards G was recorded for both R1 and R2 for all HiSeq data sets. One GAII data set (*DS32*) showed a high rate of G in the R1 reads. The nucleotides mainly affected by substitutions in this data set were T and A. For the MiSeq data sets no pronounced bias towards a certain nucleotide could be identified.

Motifs

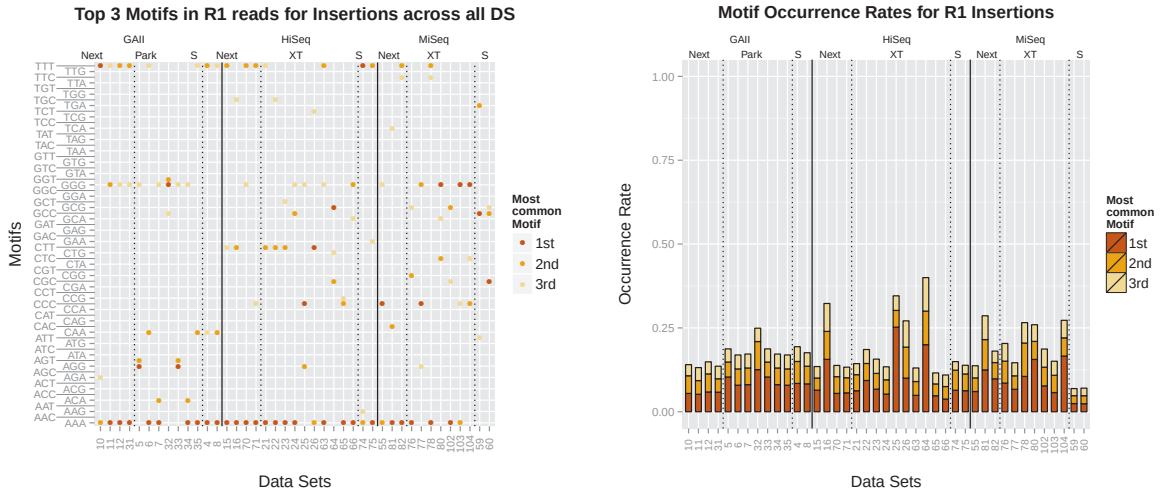
The motifs (3mers preceding errors) for all data sets were recorded. The results are displayed in Figure 7.8 and 7.9. We will first examine the motif-based nature of substitution errors for all three platforms. A coherent pattern for the substitution motifs was detected in the GAII and HiSeq data sets. The two most common motifs for both R1 and R2 reads for the GAII were "CGG" and "GGG". On average, the first motif accounted for 4.7% and 4.2% of all substitutions in R1 and R2 reads, respectively, and the second motif accounted for 3.1% in R1 and 3.9% in R2 reads. For the HiSeq data sets the same two motifs were identified. Here, "GGG" was the first motif and "CGG" the second most common motif. The bias is more pronounced with on average 9.5% and 10.0% of all R1 and R2 substitutions, associated with the first motif. For some data sets more than 17% of all R1 substitutions were associated with "GGG". For the second motif on average 5.8% of the R1 and 6.7% of the R2 substitutions were preceded by this motif. It is notable, that all first and second motifs for the HiSeq and GAII end in "GG". For the MiSeq data sets more variation among the top motifs was observed for the data sets presented in this chapter. The top three motifs accounted on average for

Table 7.4: Overview of the most common motifs for the GAII, HiSeq and MiSeq.

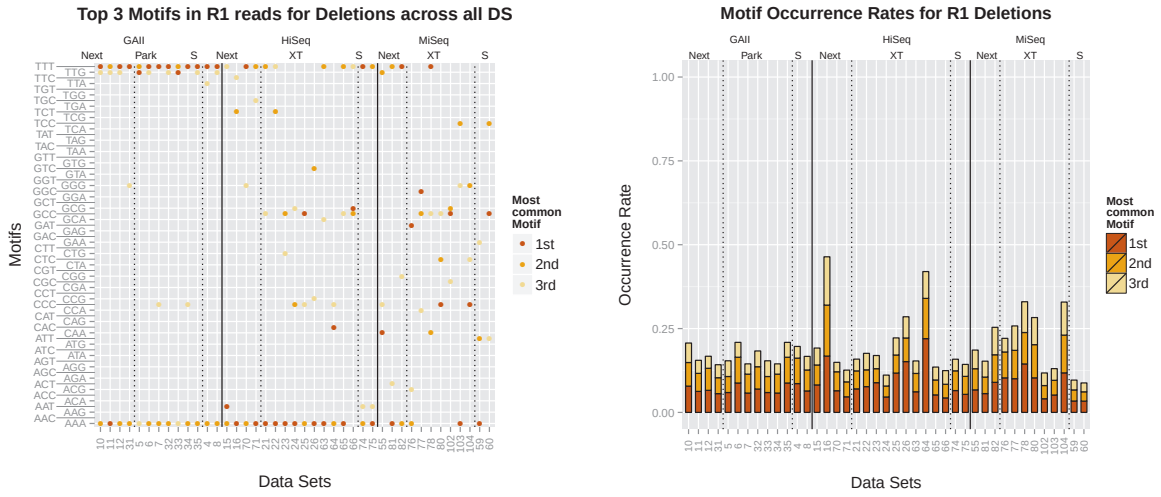
Platform	R1/R2	1st motif	2nd motif	3rd motif
GAII	R1	CGG	GGG	GCG,CCG,AAA
GAII	R2	CGG	GGG	CCG
HiSeq	R2	GGG	CGG	AGG
HiSeq	R1	GGG	CGG	AGG
MiSeq	R1	GGG,TTT	CGC,CGG,CCC,AAA	TGG,AGG
MiSeq	R2	GGG,CGG	GCT,TTT,GCG	AGG,AAA,TGG



(a) R1 substitutions

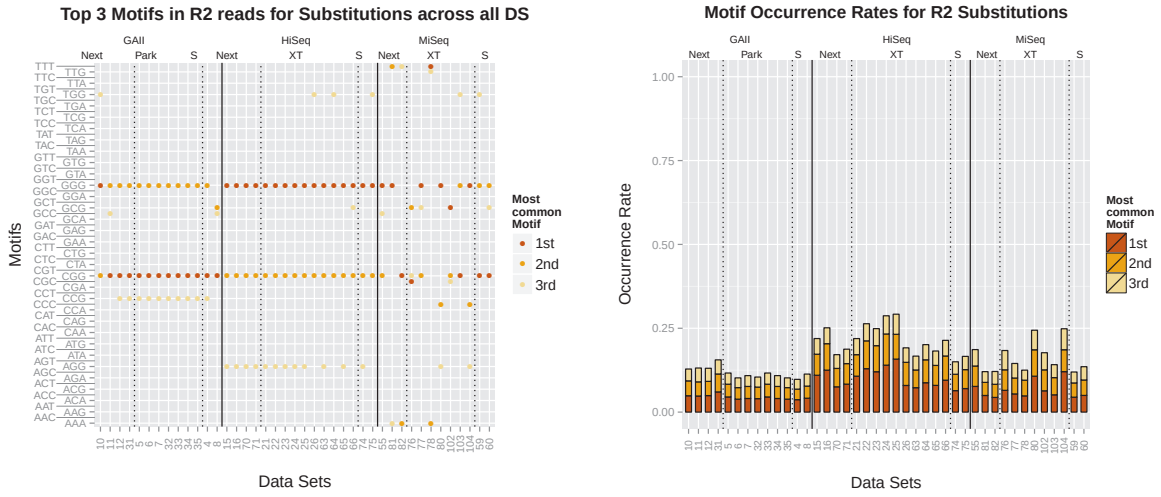


(b) R1 insertions

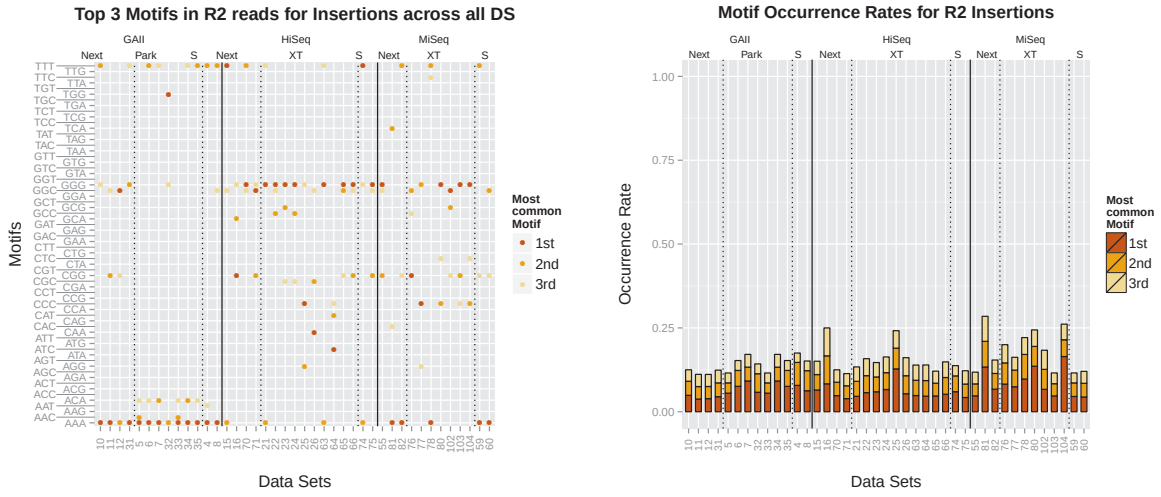


(c) R1 deletions

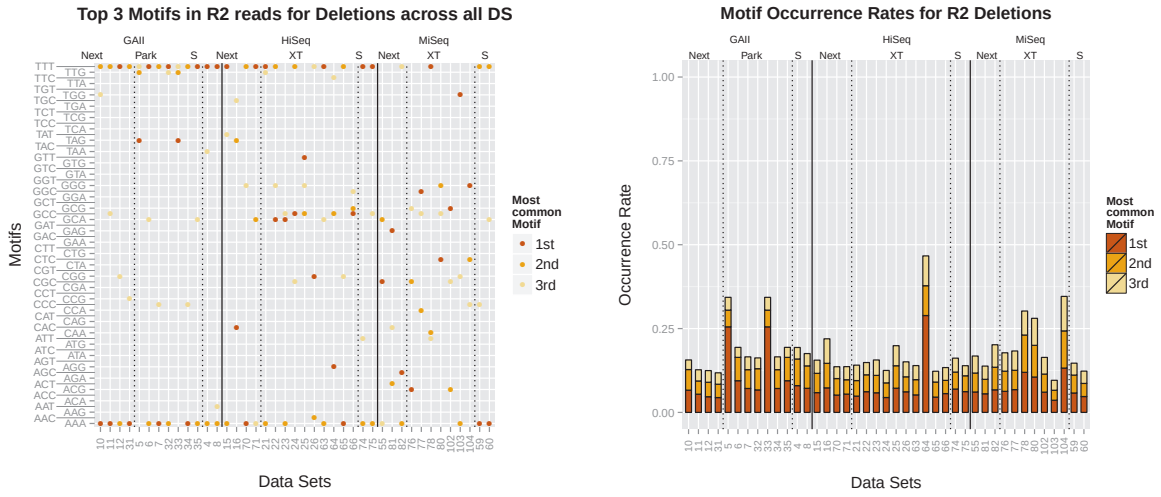
Figure 7.8: The top three motifs (3mers preceding errors) for R1 substitutions, insertions and deletions are displayed on the left. The rates associated with each motif are displayed on the right. Data sets are grouped by sequencing platform and library preparation method.



(a) R2 substitutions



(b) R2 insertions

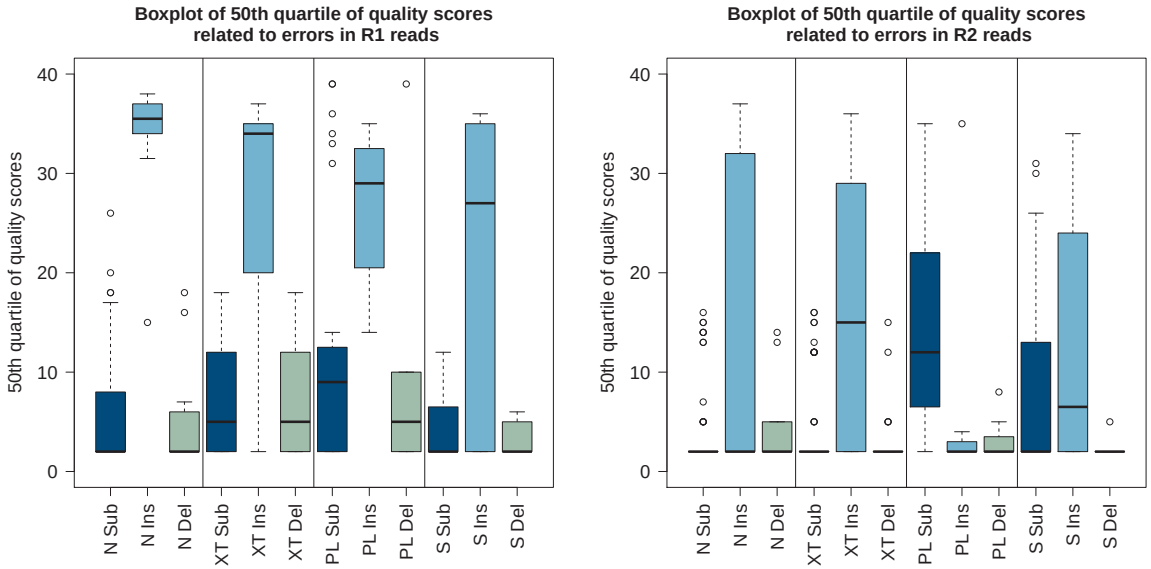


(c) R2 deletions

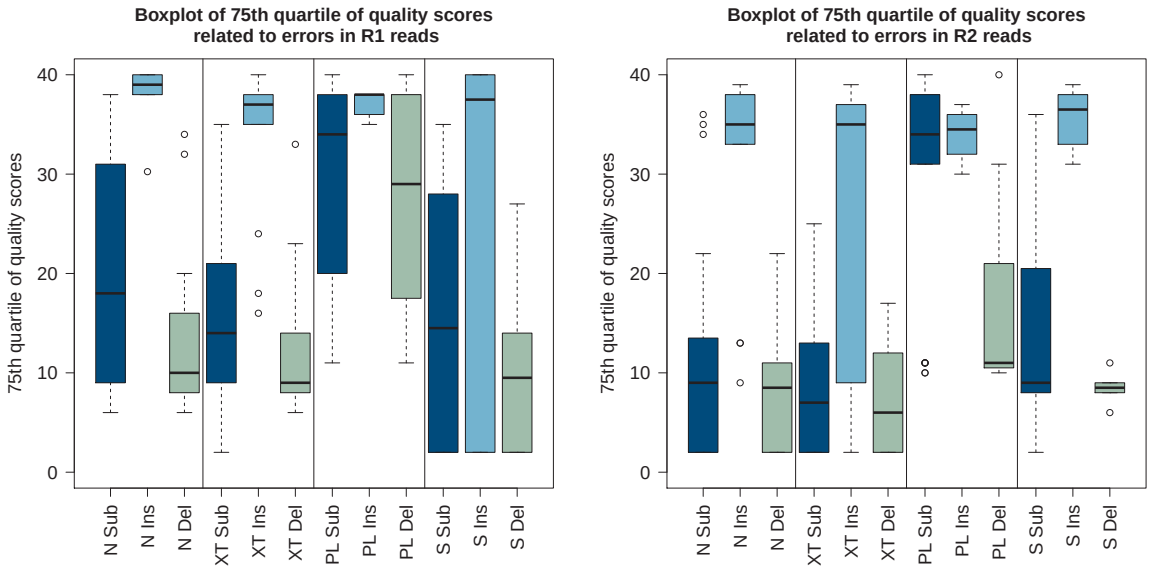
Figure 7.9: The top three motifs (3mers preceding errors) for R2 substitutions, insertions and deletions are displayed on the left. The rates associated with each motif are displayed on the right. Data sets are grouped by sequencing platform and library preparation method.

a total of 15.2% of the R1 and 16.2% of the R2 substitutions. We summarised the most common motifs for all three platforms in Table 7.4.

For insertions the most problematic 3mers were "AAA", "TTT" and "GGG". Overall, "AAA" was among the top three motifs in 31 data sets, "GGG" was observed 18 times and "TTT" 17 times for the R1 reads. We observed a similar bias for the R2 reads with "AAA" among the top three motifs in 22 data sets, "GGG" in 21 and "TTT" in



(a) 50th quartile



(b) 75th quartile

Figure 7.10: Overview of 50th and 75th quartile of quality scores associated with errors across all data sets. The results for the R1 reads are displayed on the left and the results for the R2 reads are on the right. Data sets were grouped by library preparation method (N = Nextera, XT = NexteraXT, PL = Parkinson, S = Standard TruSeq) and substitution, insertion an deletion errors are displayed separately.

16 data sets. The top three motifs accounted on average for 17.2%/18.8%/18.6% of the R1 insertions and 14.0%/15.3%/18.2% of the R2 insertions for GAII/HiSeq/MiSeq. However, the top three motifs were recorded to account for as much as 40.0% of R1 insertions and 28.5% of R2 insertions.

The two most common motifs in connection with deletion errors were "AAA" and "TTT". In 33 data sets "AAA" was either the first or second most common motif and in 26 data sets "TTT" was among the two most common motifs in the R1 reads. For the R2 reads, "TTT" and "AAA" were recorded as the two most common motifs in 28 and 26 data sets, respectively. The third motif showed more variation across data sets. The top three motifs accounted on average for 17.2%/19.9%/20.4% of the R1 insertions and 19.0%/17.1%/19.4% of the R2 insertions for GAII/HiSeq/MiSeq. The maximum rate for all three motifs was 46.4% and 46.6% for R1 and R2 deletions, respectively.

Quality Scores

Figure 7.10 displays the 50th and 75th quartile of the quality scores associated with errors for all data sets grouped by library preparation method and type of error. For the Nextera and NexteraXT data sets, the majority of deletions are well characterised by their quality scores. The majority of substitutions also showed low quality scores (below 20) expect for some R1 Nextera data sets. For the Parkinson data sets 25% of all substitutions as well as deletions on R1 reads were associated with high quality scores and therefore poorly represented. For the standard TruSeq library preparation method substitutions and deletions were overall associated with low quality scores, except for some R1 data sets where $\geq$25% of all substitutions were connected to high quality scores. Insertions were generally poorly characterised by their quality scores for R1 and R2 reads for all library preparation methods.

Comparison of error rate removal techniques

We tested two error removal strategies for the reads: quality trimming and error correction. For quality trimming the program sickle [11] (version 1.2000) with a minimum quality score of 20 and a minimum read length of 10 was used. For error correction we used the program BayesHammer which is part of the SPAdes assembler [38] (version 2.5.2). For the combination of both approaches the reads were first quality trimmed and then error corrected. Figures 7.11 and 7.12 display the results for the R1 and R2 reads, respectively. Similar rates in error reduction were observed for R1 and R2. For the GAII and MiSeq data sets, error correction removed on average more errors than quality trimming, for the HiSeq data sets quality trimming achieved better results. Averaged over all data sets, quality trimming reduced the R1 error rates by 48% (GAII: 55%, HiSeq: 50%, MiSeq: 40%) and R2 error rates by 58% (GAII: 51%, HiSeq: 66%,

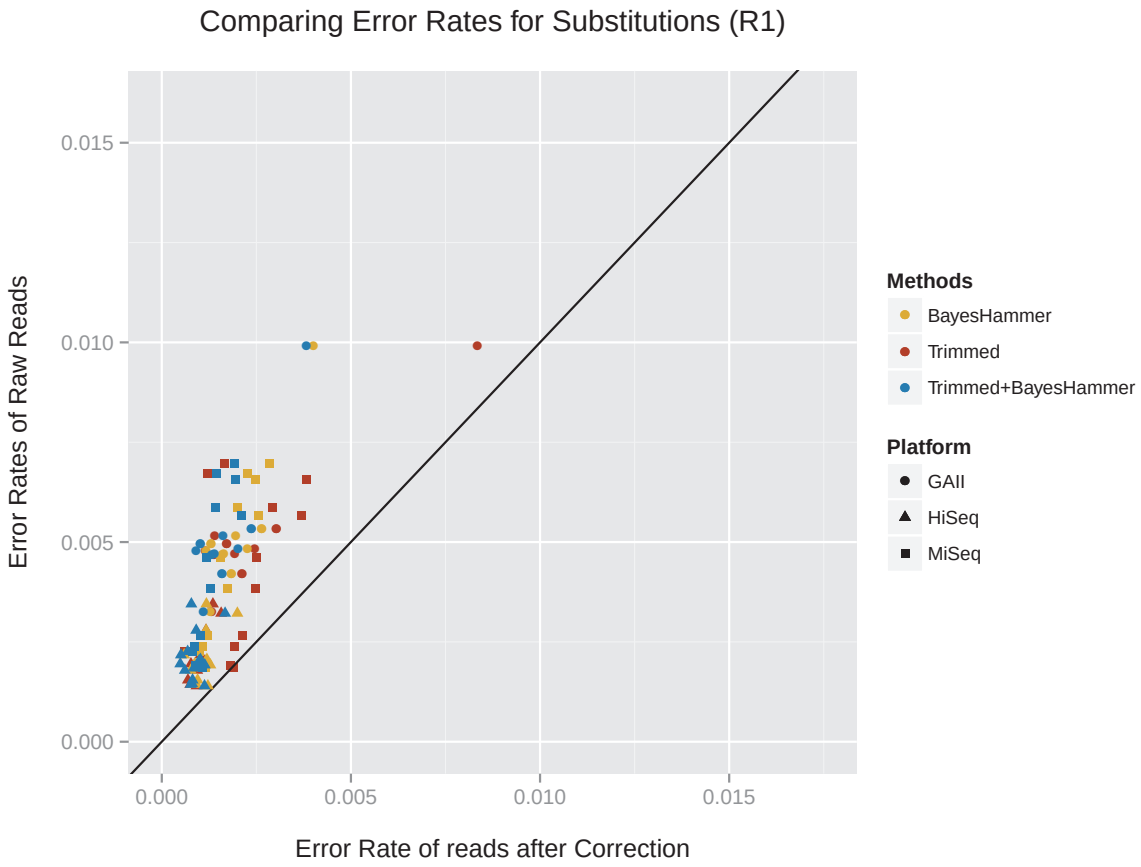


Figure 7.11: Comparison of error removal strategies for R1 reads: quality trimming with sickle (red), error correction with BayesHammer (yellow) and a combination of the two approaches (blue) was tested on all data sets.

MiSeq: 54%). Error correction with BayesHammer reduced the R1 rates by 54% (GAII: 61%, HiSeq: 46%, MiSeq: 57%) and R2 rates by 61% (GAII: 61%, HiSeq: 61%, MiSeq: 60%). The best results, on average, across all platforms were achieved by combining the two approaches: R1 error rates decreased by 62% (GAII: 67%, HiSeq: 55%, MiSeq: 65%) and R2 rates decreased by 70% (GAII: 69%, HiSeq: 70%, MiSeq: 71%).

Further, in Figure 7.13 we compared the substitution error rates for the different library preparation methods for all platforms. The grey error bars represent the initial errors based on the raw reads. The highest error rates were encountered for the GAII, followed by slightly lower rates for the MiSeq and the lowest rates were observed for the HiSeq. For each platform, the data sets prepared with the Nextera library preparation method yielded the lowest error rates. The low input libraries, NexteraXT and Parkinson, resulted in slightly higher error rates, however, the highest error rates were observed for the standard library preparation. The results based on quality trimming are presented by the red bars. The greatest error reduction was observed for the standard library MiSeq data sets. For these data sets as well as the standard library HiSeq data sets,

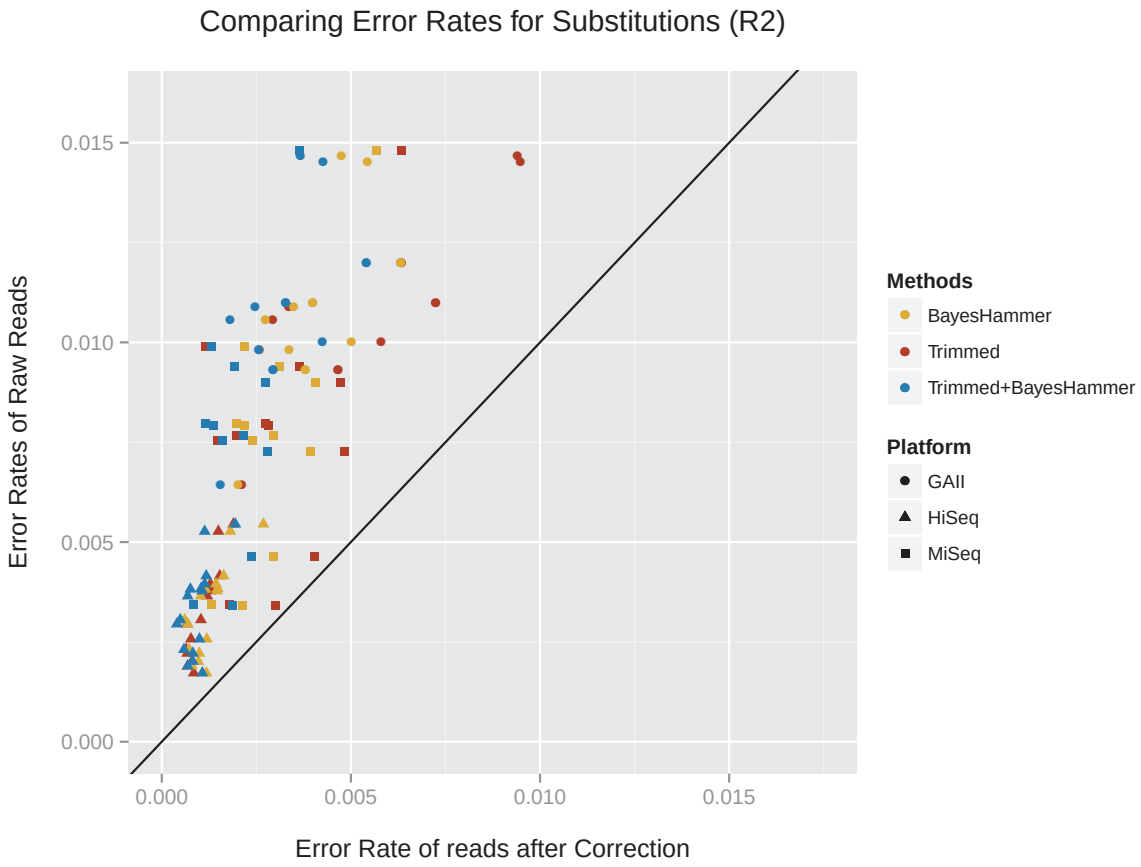


Figure 7.12: Comparison of error removal strategies for R2 reads: Quality trimming with sickle (red), error correction with BayesHammer (yellow) and a combination of the two approaches (blue) was tested on all data sets.

quality trimming worked better than error correction (represented by the yellow bars), and yielded the lowest average rates. Generally, quality trimming followed by error correction (displayed in blue) yielded the best results and the error rate showed less variability. Overall, the HiSeq data sets exhibited the lowest error rates after trimming and error correction and the best results were achieved in connection with the NexteraXT library preparation method. The MiSeq data sets showed comparable error rates after processing the reads and the best results were achieved with the Nextera library preparation method. The processed GAII data sets still exhibited the highest error rates where the best results were also achieved in connection with the Nextera library.

Table 7.5: Average percentage of aligned raw reads for the different platforms and library preparation methods (Nextera, NexteraXT, Parkinson, Standard).

GAII			**HiSeq**			**MiSeq**		
Nextera	Parkinson	Stand.	Nextera	NXT	Stand.	Nextera	NXT	Stand.
87.7%	80.6%	85.9%	88.8%	90.4%	84.1%	83.7%	95.1%	84.4%

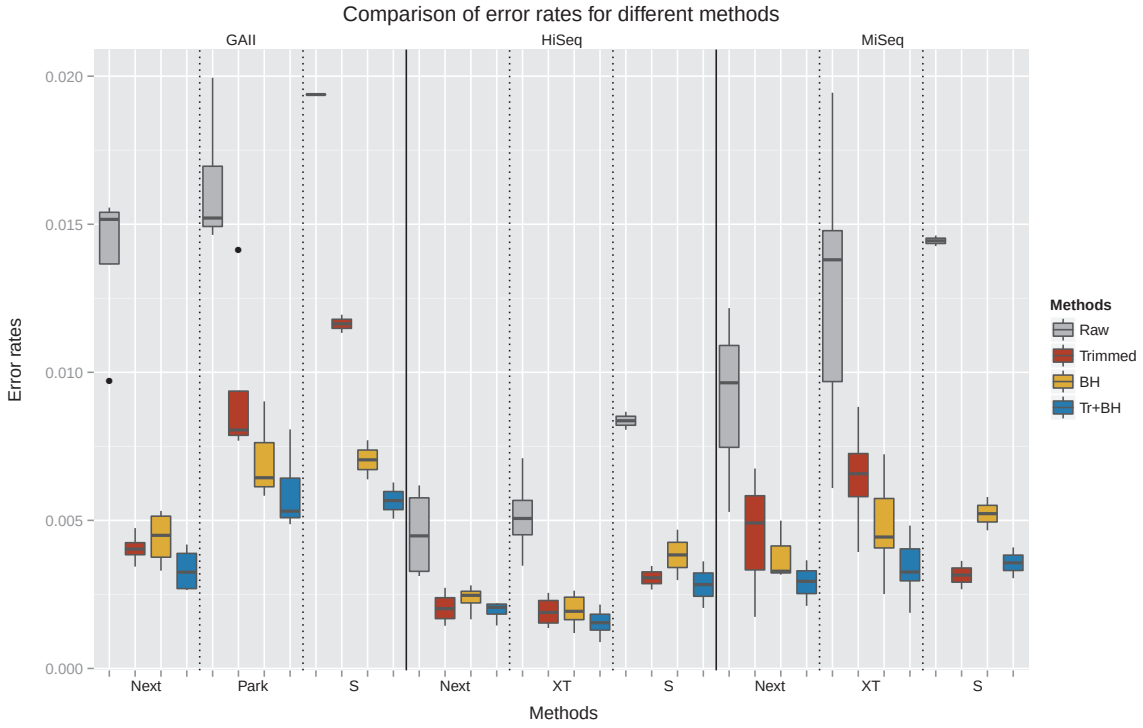


Figure 7.13: Comparison of overall substitution error rates split by sequencing platform and library preparation method. The grey bars display the error rates of the raw reads. The red bars represent the error rates after quality trimming with sickle (minimum quality score: 20, minimum read length: 10) and the yellow bars represent the results after error correction with BayesHammer. The results of the combination of both methods are displayed by the blue bars.

Aligned reads

All error rates and calculation are based on aligned reads. Figure 7.14 shows the fraction of aligned reads for all data sets and Table 7.5 shows the average rates for the raw reads across all sequencing platforms and library preparation methods. Overall, very good alignment rates were attained for all methods. The highest rates for each platform were obtained for the NexteraXT libraries sequenced on the HiSeq and MiSeq and the Nextera libraries sequenced on the GAII. The fraction of aligned reads slightly decreased after quality trimming and error correction, as reads may be shortened or discarded by the programs. After trimming and subsequent error correction 0.2-4.8% less GAII reads aligned, 0.5-5.5% less HiSeq reads and 1.9-7.4% less MiSeq reads.

7.5 Discussion

For sequencing-by-synthesis methods, such as Illumina, the DNA polymerase is a key element. The *E. coli* DNA polymerase I (Pol I) proteolytic (Klenow) fragment was the first polymerase used for Sanger sequencing and the only DNA polymerase available at the time. Fortunately, this polymerase permits the incorporation of chain termination

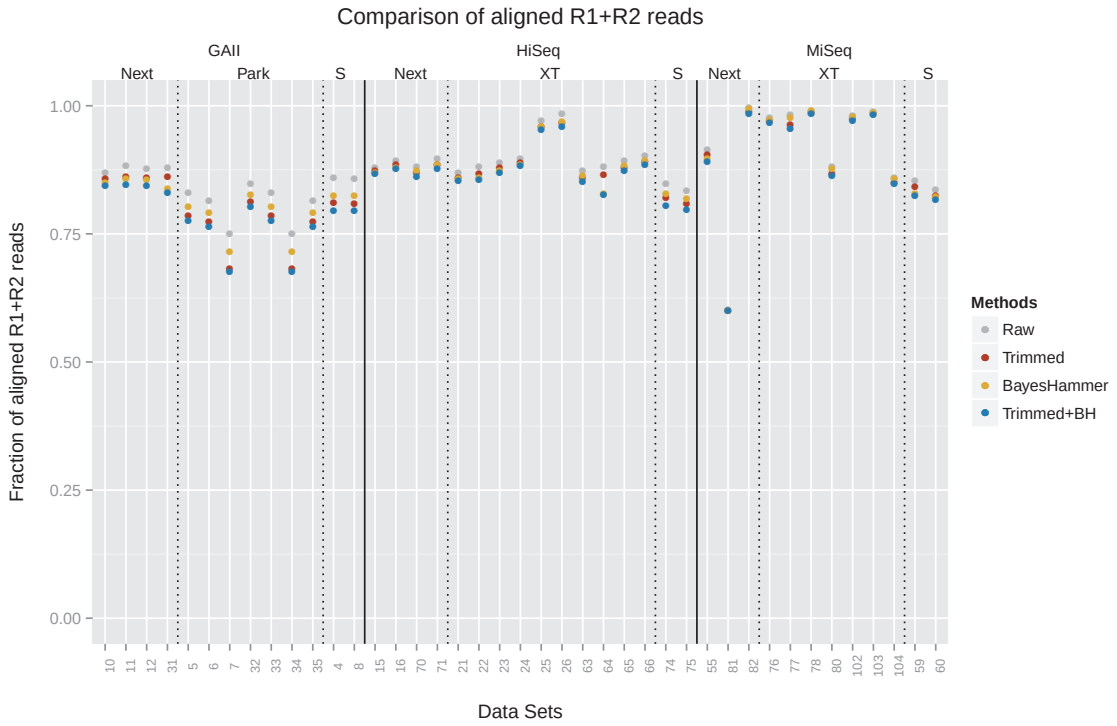


Figure 7.14: Fraction of aligned reads for the raw data sets, after quality trimming and error correction.

dideoxynucleotides (ddNTPs) which inhibit the DNA synthesis and form another key element for this sequencing method. Unlike natural dNTPs, the ddNTPs lack the 3'-hydroxyl (3'-OH) group that is required for the phosphodiester bond formation between the incorporating nucleotide and primer terminus. Therefore, the DNA polymerase terminates after the incorporation of a ddNTP. Different fluorescent labels are covalently attached to each of the four ddNTPs enabling automated DNA sequencing and single tube reactions. Further advances included the replacement of the 3'-hydroxyl group with a larger, cleavable chemical group facilitating the reversible termination of the DNA synthesis and facilitating the current NGS sequencing-by-synthesis methods (see Figure 7.15). An overview of the Illumina sequencing process can be found in Figure 7.16.

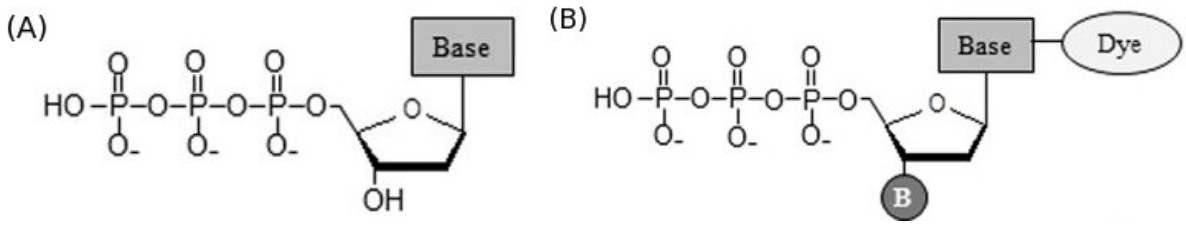


Figure 7.15: (A) Deoxynucleotides (dNTPs): natural nucleoside triphosphates that get incorporated during DNA polymerase. (B) Reversible dye-terminators: engineered nucleotides used for Illumina sequencing-by-synthesis. (Figure adapted from [50].)

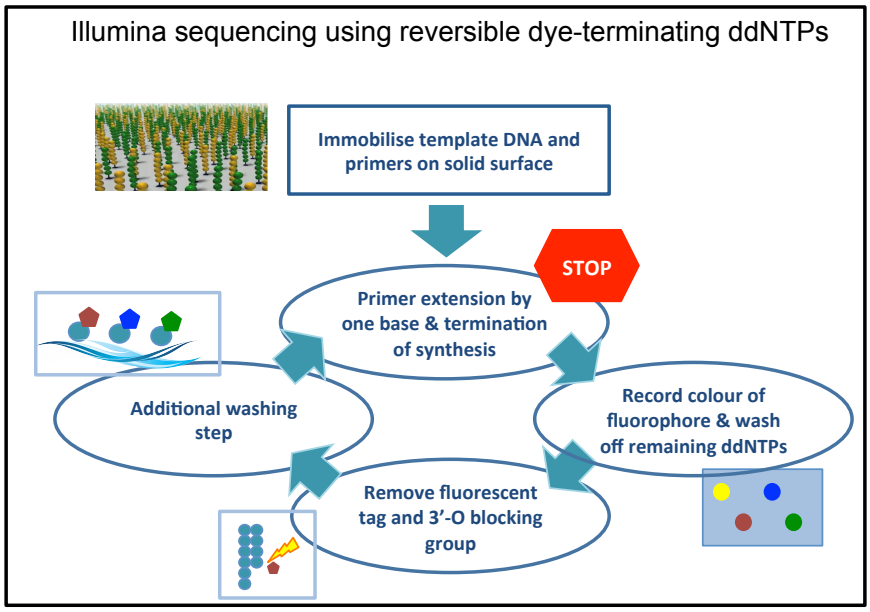


Figure 7.16: Overview of Illumina sequencing process: The template DNA sequences including the primers are first immobilised on a solid surface. During each cycle the polymerase incorporates one reversible dye-terminator base (ddNTP). The synthesis is temporarily paused and the dye is excited with a laser to identify the incorporated nucleotide. All remaining ddNTPs are washed off and the fluorescent tag and 3'-O blocking group is removed. This is followed by another washing step before the DNA polymerase recommences.

Changing the 3'-OH group results in a modified moiety and makes it harder for the DNA polymerase to accept the engineered nucleotides. The original Klenow enzyme was not capable of efficiently incorporating these modified nucleotides, creating a need for a new enzyme. Sequencing information facilitated the discovery of multiple DNA polymerases from mesophilic/thermophilic viruses, bacteria and archaea and greatly advanced the search for a new enzyme suitable for sequencing-by-synthesis methods.

The *Thermus aquaticus* (*Taq*) DNA polymerase has been a commonly used enzyme for DNA sequencing as the *Taq* pol is readily purified, thermostable and can be further modified. The original enzyme incorporates ddNTPs much slower than dNTPs. A mutation (F667Y) greatly increased the efficiency of ddNTP incorporations. However, the *Taq* pol enzyme favours the incorporation of ddGTP over the other ddNTPs, due to interactions between the guanidinium side chain of the arginine residue 660 (Arg660) and O6/N7 atoms of the guanine base. A substitution in the Arg660 residue with a negatively charged aspartic acid, aims at remediating this bias. However, this is no longer achieved if the larger reversible dye-terminators are used. Development of different 3'-O-blocking groups has been an active field of research. Illumina/Solexa developed the 3'-O-azidomehtyl 2'-deoxynucleoside triphosphates and a mutant of the archaeal 9°N DNA polymerase of the hyperthermophilic *Thermococcus sp. 9°N-7* is used during sequencing. Limited information is available on the exact mutations due to

commercial considerations. However, for all data sets we still observed a dominant bias towards the incorporation of ddGTPs.

For the *Taq* DNA polymerase, it has also been reported [98] (1999) that ddGTPs get incorporated ten times faster than the other ddNTPs. Li et al. subsequently studied the crystal structure for the different ddNTPs. The ddGTP ternary structure differs from the other ddNTPs as it possesses an additional hydrogen bond between the side chain of the Arg-660 residue and the base of the ddGTP complex. A mutation of the ARG-660 can reduce the incorporation rate of ddGTP and resolved the problem for Sanger sequencing methods [50][39][51]. We observed that T is more prone to substitution errors than the other nucleotides. This could be related to the different structure of the ddGTP nucleotide. If a T is sequenced the polymerase actually encounters a G in the template sequence and tries to add the complementary nucleotide T to the copied strand.

After cleavage of the linker group carrying the fluoresphore, extra chemical molecules on the normal purine and pyrimidine bases remain resulting in a vestige. These vestiges can perturb the DNA polymerase and limited the possible read length as they impair the stability of the DNA and hinders the substrate recognition and primer extension. Chen et al. (2013) [51] described an accumulation of these vestiges in Illumina sequencing. Illumina has been able to achieve longer reads by adding reversible terminator nucleotides without the fluorophore to reduce the effect of vestiges, but their impact is still apparent as increased error rates towards the end of the reads. We hypothesise that these vestiges encourage the accumulation of errors.

Furthermore, a bias associated with the Nextera libraries was recorded. The transposase used for this technology is based on a mutated Tn5 transposome [105]. Transposomes are capable of inserting themselves into a target DNA sequence. The wildtype Tn5 enzyme has been described as inactive [133], however, the mutations resulted in an increased insertion rate making the enzyme suitable for library preparations. For the wildtype Tn5, hot spots for insertions have been reported. The enzyme contains 19bp target recognition sites that are present at the ends of the transposase (Tnp), a protein that is part of the transposon complex and responsible for the catalytic steps. The target recognition sites are required in order for the transposon to bind to the template DNA for the subsequent insertion and it has been hypothesised that specific contacts must be formed between Tnp and the target DNA [32]. Ason et al. [32] observed high frequencies of insertions into A/T rich regions (in particular TTATA) flanked by GC pairs. As the recognition sequence of Tnp contains the same subsequence (TTATA), they suggest that Tnp favours insertions into regions containing a portion of the recognition site. Our data suggests that the mutated Tn5 enzyme used in the Nextera technology, shows a similar

bias accounting for the uneven distribution at the start of the R1 and R2 reads. The length of the fluctuations concurs with the length of the recognition site and higher rates of A/T were observed in the first part of this region followed by elevated G/C rates. However, this bias was not associated with errors and therefore these fluctuation do not need to be removed by trimming the start of the reads. It needs to be determined though if this tendency results in a coverage bias of the sequenced genomes and/or the coverage of the genomes in the community.

The Nextera method has many apparent advantages: it requires less DNA input material and the template DNA is simultaneously fragmented and tagmented facilitating shorter preparation times. A limited-cycle PCR step is involved in the tagmentation step, therefore, higher error rates were expected for the Nextera data sets. However, for all three platforms the data sets prepared with the Nextera kit showed the lowest error rates (see Figure 7.13).

7.6 Conclusion and Future Work

The individual error profiles confirmed an increase of the error rates towards the end of the reads, which has been previously reported and is attributed to an accumulation of phasing and pre-phasing problems during the run. The chemical and structural properties associated with the ddNTPs seem to contribute to this effect. We also observed a coherent preference for the substituting nucleotide and established a connection with crystal structures of the ddNTP that shows why incorporations of ddGTPs are favoured.

All motifs preceding substitution, insertion or deletion errors were recorded. The main motifs for substitutions were related to 3mers ending in "GG". This is assumed to be related to issues of the polymerase with the engineered ddNTPs and the structural properties of ddGTP. Overall, 16% of all substitution errors can be associated with only three motifs. Insertions and deletions were mainly preceded by the homopolymers "AAA", "TTT" and "GGG". The top three motifs accounted on average for 17% of all insertions and 19% of all deletions.

For the Nextera technology fluctuations across the first 20bp of the nucleotide distributions of the reads were identified. These seem to be related to the transposase recognition sites. Although introducing a bias, these fluctuations were not associated with errors and therefore trimming the start of the read is not required.

We showed that the quality scores can characterise the majority of substitution and deletion errors for Nextera, NexteraXT and the TruSeq library preparation method. However, quality scores are meaningless for insertions. Insertion and deletion rates are

1,000 times lower than substitution rates and therefore less significant. For applications where low frequency variants are important, the motifs identified in connection with indel errors can be used as further indication for the reliability of observed SNPs. Quality trimming (sickle) combined with subsequent error correction (BayesHammer) provided the best results in terms of error removal. Although the number of aligned reads slightly decreased on average by 3% due to shortening and discarding of the reads during the different error removal strategies, the error rates can be reduced by as much as 84%. On average substitution error rates were reduced by 66%.

The best accuracy was observed for the Nextera and NexteraXT library preparation methods. This technology facilitates simultaneous fragmentation and tagmentation of the DNA sample, resulting in shorter preparation times. In addition less input DNA is required for these methods. In connection with the proposed error removal strategy, we were able to reduce the error rates of the longer MiSeq reads to a level comparable to the HiSeq reads. This accentuates the MiSeq benchtop sequencer and the Nextera library preparation method as an excellent option for sequencing applications.

8 Validation of Taxonomic Classification Algorithms for *in vitro* Bacterial Metagenomes

8.1 Abstract

Sequencing has for the first time facilitated the identification of organisms in a community as well as the discovery of new species by directly sequencing environmental samples. Taxonomic classification is an essential tool for understanding the composition and function of complex microbial communities. We benchmarked several taxonomic classification algorithms based on *in vitro* bacterial metagenomes. We designed a complex *in vitro* mock community comprising 59 species of bacteria and archaea in known proportions (see Chapter 6.3 for details). The sample was sequenced on the Illumina Genome Analyzer II. Here, we present the results for an initial study comparing the performance of two new taxonomic classification programs developed at the University of Dusseldorf: PhyloPythiaS+ and taxator-tk. We identified the advantages and disadvantages of the different approaches. We also explored the limitations attributed to the different databases by excluding the sought-after genomes from various taxonomic levels. The benchmarking was later extended by the University of Dusseldorf for the publication of PhyloPythiaS+ which reinforced our initial findings.

Original Contributions

I conducted an initial study assessing the performance of two new taxonomic classification programs. This work was done within the scope of the European Cooperation in Science and Technology (COST) scientific programme on "*Microbial ecology & the earth system: collaborating for insight and success with the new generation of sequencing tools*" (Action ES1103). I was awarded a grant for a Short Term Scientific Mission (STSM) to spend two months at the University of Dusseldorf in Alice McHardy's group where several taxonomic classification algorithms were developed. I used a more complex mock community data set compared to previous validations studies (see Chapter 6.3). The results were then extended by Dusseldorf for the publication of PhyloPhythiaS+. For this, I developed error profiles representative of the Genome Analyzer II and generated *in silico* data sets with different abundance distributions for the benchmarking. I then assembled the reads and blasted the contigs which comprised the input for the programs.

This chapter is partly based on the publication:

> Ivan Gregor, Johannes Dröge, Melanie Schirmer, Christoper Quince, Alice C. McHardy. **PhyloPythiaS+: A self-training method for the rapid reconstruction of low-ranking taxonomic bins from metagenomes.** (PLOS Computational Biology: In review)

8.2 Introduction to Taxonomic Classification and Phylogenetics

Partitioning microbes into different species is one of the most fundamental components of microbial taxonomy and is essential for studying microbial diversity, albeit the definition of a species is still a controversial and difficult subject. Several studies [52] have shown that the current definition of a species for bacteria is too broad as bacteria can have highly similar 16S rRNA gene sequences but very divergent genomes [81].

For microbial taxonomy phenotypic and sequence-based phylogenetic data is combined. Well established standards and guidelines help us to identify and describe prokaryotes but there is no universal concept of a prokaryotic species. This is a fundamental question for studying the microbial diversity on our planet since it will form our understanding of the concept of diversity. Due to the small size of bacteria a phenotypic definition of a species is not feasible as our current technologies only provide limited information in this regard. The biological concept that describes a species as an isolated interbreeding population is also not applicable to the microbial world. New bacterial and archeal species arise due to evolution for which mutations and horizontal gene transfer play an important role. For microbes a functional differentiation seems more appropriate. But in order to measure the metabolic capabilities adequately, the species needs to be isolated on culture plates. This is problematic as approximately 99% of all bacteria are not cultivable yet. Therefore, next generation sequencing data currently provides the best approach for taxonomic classification.

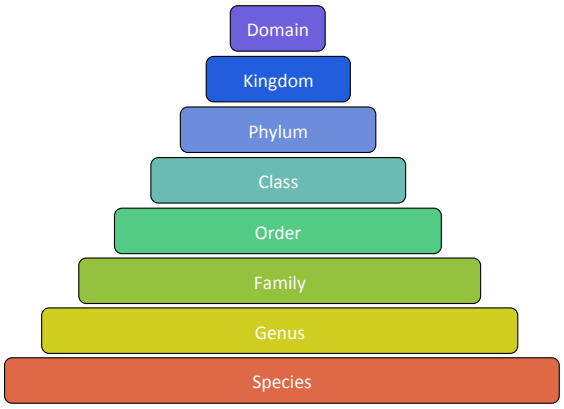


Figure 8.1: Overview of bacterial taxonomic ranks.

A prokaryotic species is currently defined "as a collection of strains sharing a high degree of similarity in several independent traits" [45, p. 391]. For instance at least 70% of DNA-DNA hybridisation (DDH) or a minimum of 97% similarity for the 16S rRNA gene are required. On average a 70% DNA-DNA hybridisation corresponds to 95% identical nucleotide sequences. So within a species different strains may well differ up to 5%. There are also cases were these two criteria are contradictory as there are species with more than 97% similarity in the hypervariable 16S rRNA regions though characterised as different species by the DDH criterion [156].

During the genotypic analysis organisms are grouped based on similarity which also

allows us to derive phylogenetic relationships. The DNA sequence of single genes, so-called markergenes, can be studied to infer ancestral relations. As mentioned above, one of the most important examples are the genes encoding for the ribosome and are thus always present in bacterial DNA. As these genes encode such a fundamental function, they have evolved slowly and are sufficiently conserved for phylogenetic analysis. The 16S rRNA gene is the most commonly used phylogenetic marker and has been of great use to distinguish organisms. With the accomplishment of being able to sequence single genes, the focus on the 16S rRNA gene revealed a far more complex composition and structure of microbial communities than previously imagined. So far our knowledge about the diversity and structure of microbial communities is in large parts based on the analysis of the 16S rRNA gene. Hence ribosomal databases are much more extensive than any other sequencing database. Encoding for such a fundamental process as translation, the 16S rRNA gene is highly conserved among species. This is also one of the limitations of 16S rRNA as the resolution at the species level is quite restricted.

The limitations of 16S rRNA on a species level can be illustrated with one of the most intensively studied microorganisms on earth: *E.coli*. *E.coli* is an essential component of the microbial communities in the human gut and important for our well-being. But at the same time some *E.coli* strains have pathogenic properties. "Fully sequenced representatives of the *E.coli* species [...] were found to differ in up to 30% of their genes" [99, p.12] although having identical rRNA genes. Analysis based on the 16S rRNA gene is not sufficient to distinguish between those strains and reveals limited information about the genomic content of an organism. For the distinction of different strains of a species we often need to shift our focus to other genes which are in many cases species specific. For example, for the genus *Photobacterium* a phylogeny for different strains can be based on the analysis of the *gyrB* gene, one of the housekeeping genes encoding for the gyrase subunit B, and the *luxABFE* genes which encode light-emitting enzymes.

Furthermore, genomes can contain multiple copies of the small subunit rRNA gene which can further complicate the analysis. Usually the variation within a genome is very low, but this is not always the case. Pei et al. [122] studied the intragenomic variation of 16S rRNA genes of 568 unique species, of which 425 species contained two to 15 copies of the 16S rRNA gene per genome. The observed sequence divergence ranged from 0.06% to 20.38% for 235 of these species. A threshold of 1 - 1.3% was applied to distinguish on the species level. (Note that when the complete 16S rRNA sequence is considered a threshold of 1 - 1.3% is commonly applied whereas a threshold of 3% is usually applied for highly variable regions (e.g. V5).) The authors found that ten species showed a nucleotide divergence between 1 - 1.3% and 14 showed a nucleotide divergence of more than 1.3%. Sequence divergence as high as 20.38% was observed in the case of

the bacterium *B.afzelli*. Seven of these species, that showed a diversity exceeding the threshold, are also associated with the human microbiome or diseases.

Overall only 4.2% of the species showed intragenomic variation of more than 1%. In the case of *B.afzelli* the authors assume that the high diversity is due to a nonfunctional 16S rRNA pseudogene. Gene truncation and intervening sequences are possible explanations for the high diversity levels as well. Pei et al. see no contradiction to the theory of ribosomal constraint but suggest that intragenomic variation may result in overestimation of a population's diversity. The simultaneous analysis of other markergenes may resolve this issue.

8.3 Taxonomic Classification Programs

Dusseldorf has developed various programs for taxonomic classification of metagenome contigs and reads:

- PhyloPythiaS [121]:
  PhyloPythiaS (PPS) is a composition-based taxonomic assignment software that is based on structural support vector machines (SVM). The composition-based classifier uses short substrings (k-mers) to represent the individual sequences as vectors of fixed-length. These vectors are used to determine the similarity among the sequences. This representation is also referred to as the "genomic signature" of an organism. It has been observed, that these signatures are more similar between closely related species than distant species. The structural SVM is a supervised learning method. It infers a taxonomic model based on sequences of known taxonomic origin. The taxonomic model is created during the initial phase, also referred to as the training phase. The user can supply sample specific data, for example a set of sample specific sequences with known taxonomic classification, or just a list of clades to be modelled in which case the NCBI genomes are utilised as training data. The derived model is then used to predict the taxonomical classification of new data (i.e. contigs of $\geq$ 1,000bp). For the case where no further information of the sample's taxonomic composition is available, a generic model is provided and can be used for the predictions. However, higher predictive accuracy is generally achieved if a sample specific model is created during the training phase, as the sample specific model includes clades for the most abundant organisms.

- PhyloPythiaS+ *(unpublished)*:
  PhyloPythiaS+ (PPS+) constitutes an extension of PPS. An overview of the pipeline can be found in Figure 8.2. PPS+ automatically derives a sample-specific

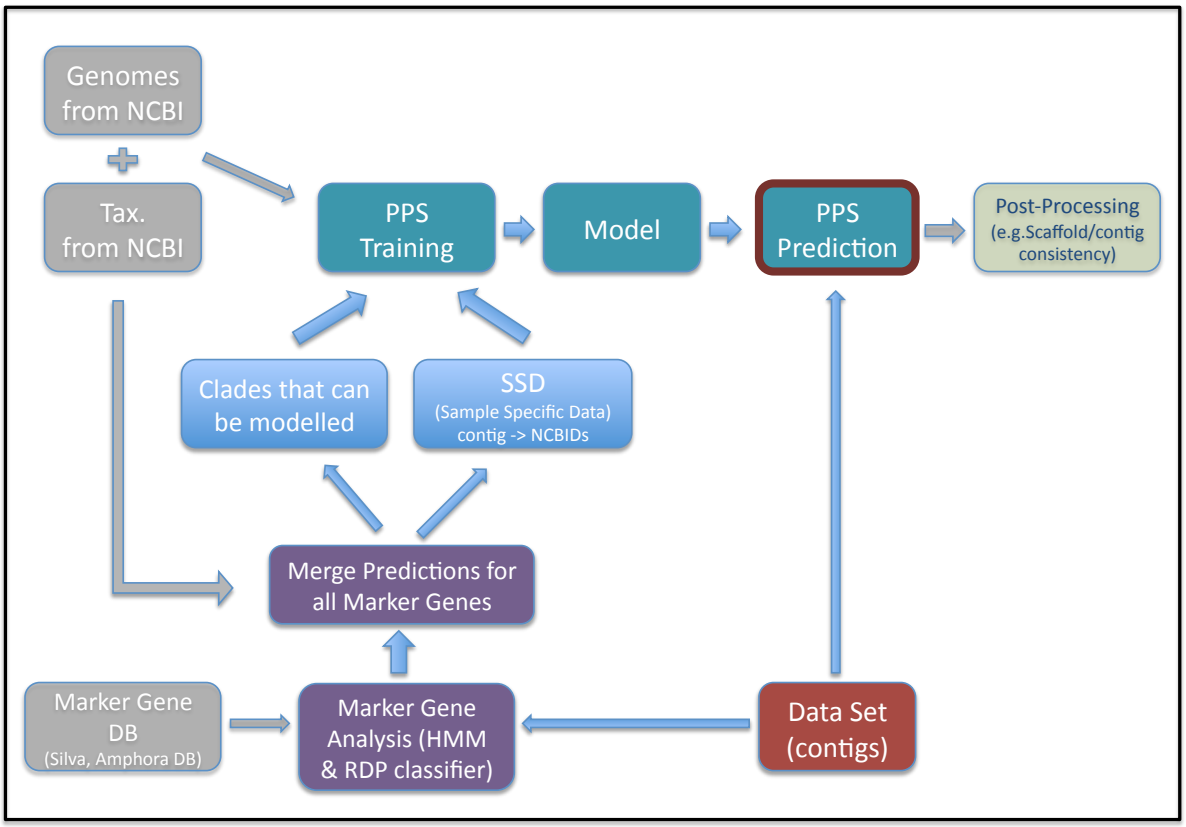
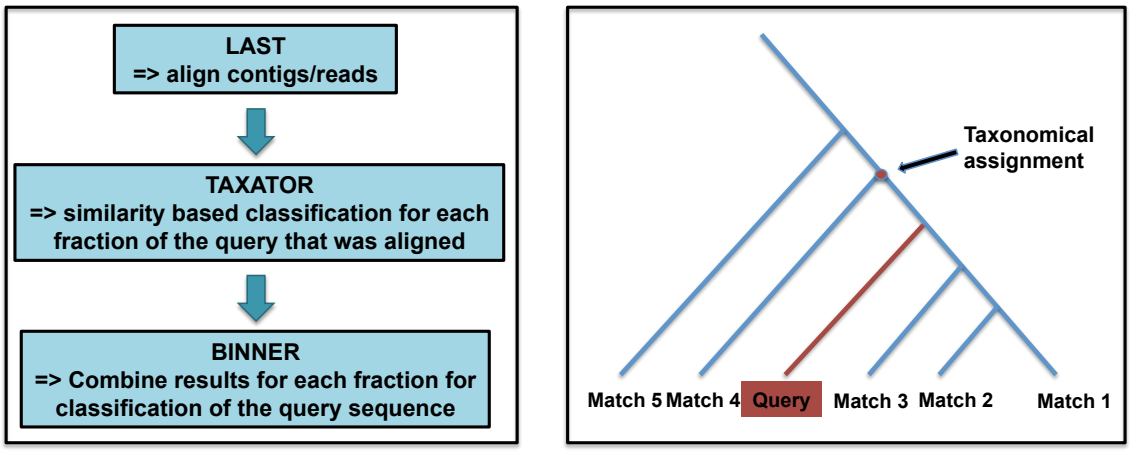


Figure 8.2: Overview of PhyloPythiaS+ workflow.

model based on markergene classification and determines the clades that can be modelled for the sample. A Hidden Markov Model and a Bayesian Classifier are utilised to identify and classify the markergenes of the input contigs (e.g. 16S, 23S, 5S & Amphora markergenes). The mapping of the markergenes to their taxonomic assignment is based on the Silva and Amphora database. The predictions for the markergenes are then merged by finding the lowest common ancestor to determine the taxonomic assignment of the entire contig. In addition the set of clades that can be modelled is inferred. For every clade at least 100kb or three genomes from different species are required. Subsequently, predictions are made with PPS using the model derived from the sample specific data and the clade information.

- taxator-tk [59]:
  Taxator-tk is a sequence similarity-based classification tool. Figure 8.3a provides an overview of the different steps during the analysis. First the contigs/reads are aligned with the software LAST. Taxator-tk then separately classifies each fraction of the query that was aligned. Figure 8.3b illustrates the process of the taxonomical assignment. Here, "Match1" corresponds to the best alignment. The pairwise alignment scores to "Match 1" determine the order of the alignments and the position of the query sequence in the graph (see Figure 8.3b). The process is repeated by aligning everything to the match that is closest to the query but with a



(a) Workflow Overview (b) Taxanomic assignment

Figure 8.3: Workflow of taxator-tk and illustration of taxonomic assignment of a query sequence.

lower pairwise alignment score ("Match4" in Figure 8.3b). The two predictions are then merged by taking the lowest common ancestor. In the last step the predictions for the fragments are merged in order to obtain the taxonomic assignment for the entire query sequence.

8.4 Measurements for Performance Evaluation

We used the following measurement for the performance evaluation [35]:

$$\textbf{Recall} = \frac{1}{N} \cdot \left(\sum_{i=1}^{N} \frac{tp_i}{t_i} + \frac{tp_{\text{other}}}{t_{\text{other}}}\right)$$

$$\textbf{Precision} = \frac{1}{M} \cdot \sum_{i=1}^{M} \frac{tp_i}{p_i}$$

Where:

$$\begin{aligned} t_i &= \#\text{ of bp of clade } i, \qquad p_i = \#\text{ of bp predicted to clade } i \\ tp_i &= \#\text{ of bp correctly assigned to clade } i \\ N &= \#\text{ of true clades}, \qquad M = \#\text{ of predicted clades} \\ other &= N \backslash i \end{aligned}$$

In other words, the recall is the average fraction in bp of the "true" bins that was correctly classified, weighted by the size of the bin. The precision can be interpreted as

the average fraction in bp of the predicted bins that was correctly classified, weighted by the size of the bin.

In addition we calculated the **"Total Assigned"**, which is the fraction of bp of the query sequences that were assigned to a taxonomy.

8.5 Results of the Taxonomic Evaluation

Our mock community was sequenced on the Genome Analyzer II. This yielded about 22,500,000 paired-end reads of 100bp. The reads were assembled into contigs using MetaVelvet and Minimus2. We then blasted the contigs against the 59 species in our population and filtered out contigs that had no blast hit or were shorter than 1,000bp. We used the resulting set of contigs for the validation of the taxonomic classification algorithms.

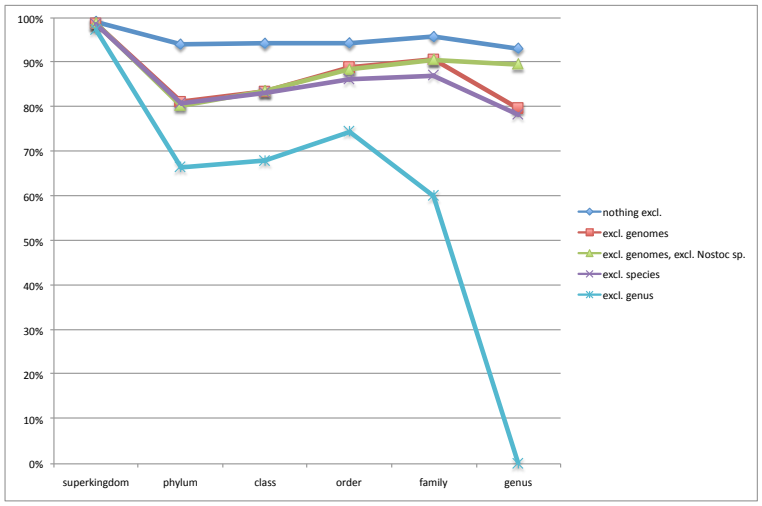
The three graphs in Figure 8.4 show the results for PPS+ for different scenarios. PPS+ utilises two databases - a markergene database and the NCBI database (genomes & taxonomy). We compared the performance for the following cases:

- The sought-after genomes are present in the databases (nothing excluded):
  All of the organisms in our mock community have been sequenced and studied previously and the corresponding information has been added to the respective database. It is highly advantageous for the taxonomic classification programs if the genome of the organism that they are trying to classify is already known and present in the databases.

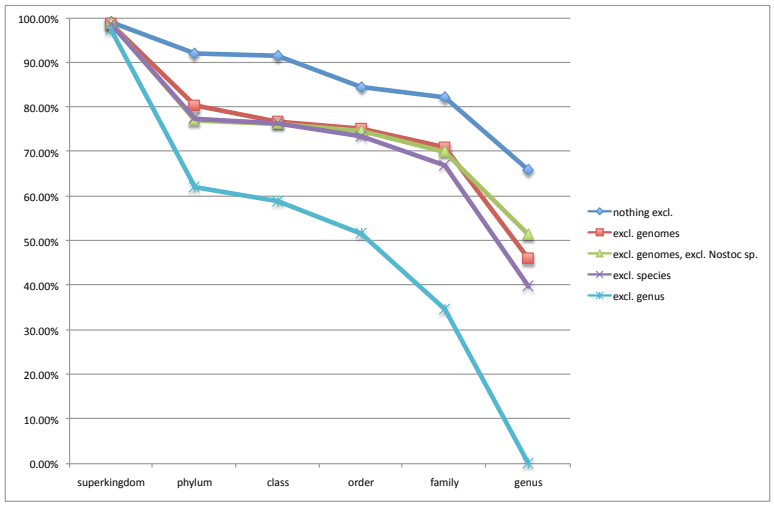
- Excluding the sought-after genomes:
  To test the ability of the programs to classify new organisms, we successively removed information from the databases. Here, the genomes of the organisms in our mock community were excluded from the NCBI database as well as their corresponding markergenes in the case of the markergene database. We tested how this effects the ability of the programs to correctly classify the input data.

- Excluding species information:
  In addition to excluding the genome sequences of the organisms that are present in our mock community, all information on the species level was removed from the databases. This corresponds to the case where the programs encounter a new species.

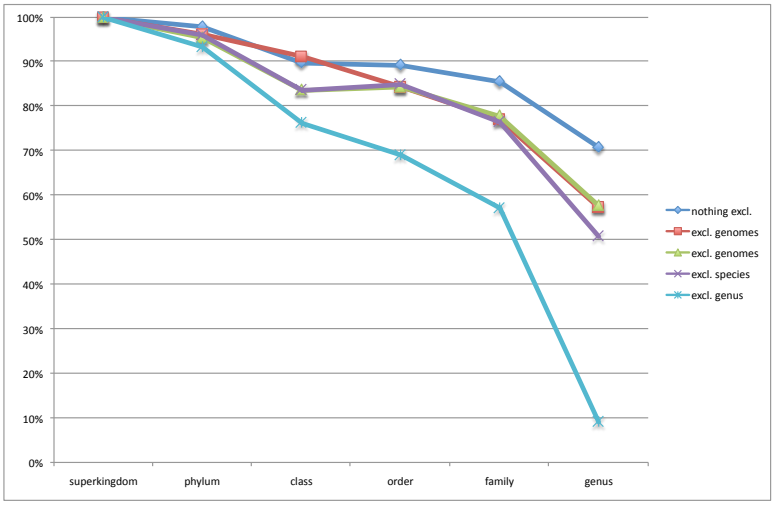
- Excluding genus information:
  Here, additionally all information on the genus level was removed from the databases



(a) Precision PPS+



(b) Recall PPS+

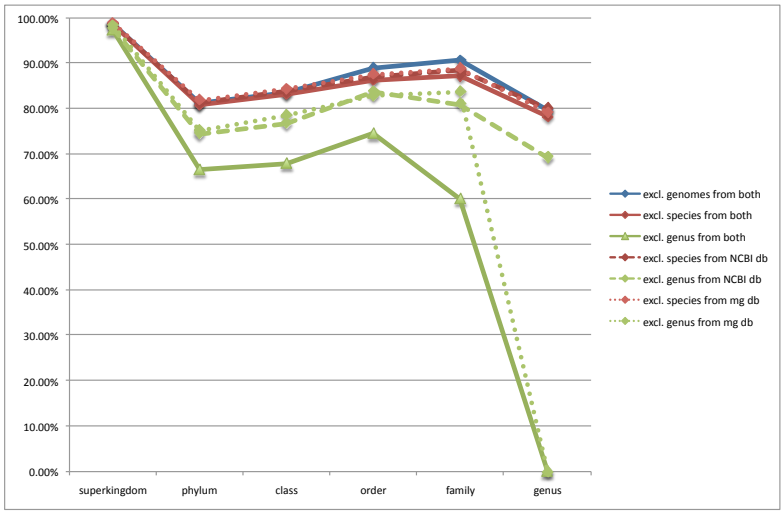


(c) Total Assigned PPS+

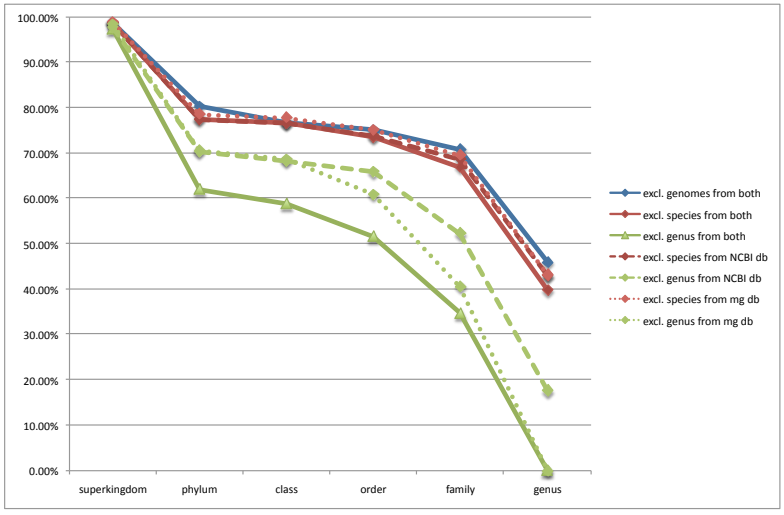
Figure 8.4: Precision, recall and total fraction of assigned bp for PPS+ across various scenarios where the organisms in the mock community were successively excluded from the database.

which corresponds to the case where the programs try to classify an organism of a previously unknown genus.

The dark blue line in Figure 8.4 illustrates the ideal scenario where nothing was excluded from the databases and the sought-after genomes are present in the databases. For the red line the genome sequences were excluded from both the NCBI database and the markergene database. For one of the organisms in our population, *Nostoc sp. PCC 7120*, all sequences on the genus level were assigned to the organism *Anabaena variables* and rated as misclassified although these are in fact referring to the same organism [53]. Excluding all of the sequences related to *Nostoc sp. PCC 7120* from the query data set showed a much better precision rate (green line). The purple line corresponds to the case where we excluded everything on the species level from both databases and for the light blue line we excluded everything on the genus level from the databases. Excluding



(a) Precision PPS+

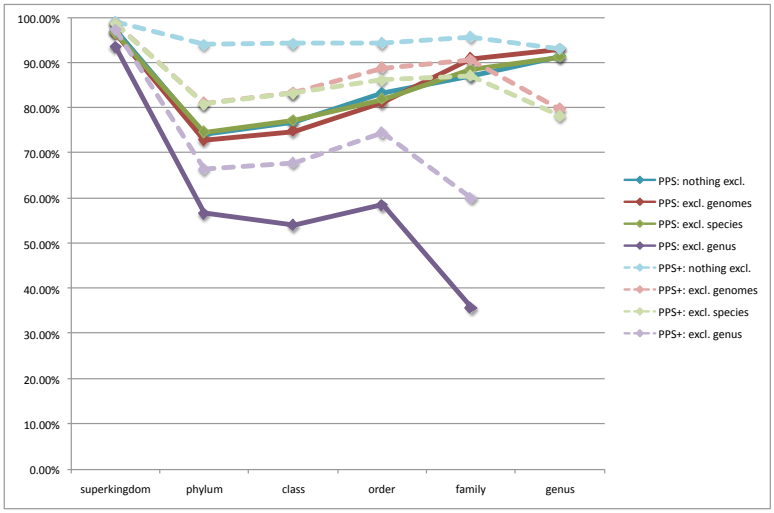


(b) Recall PPS+

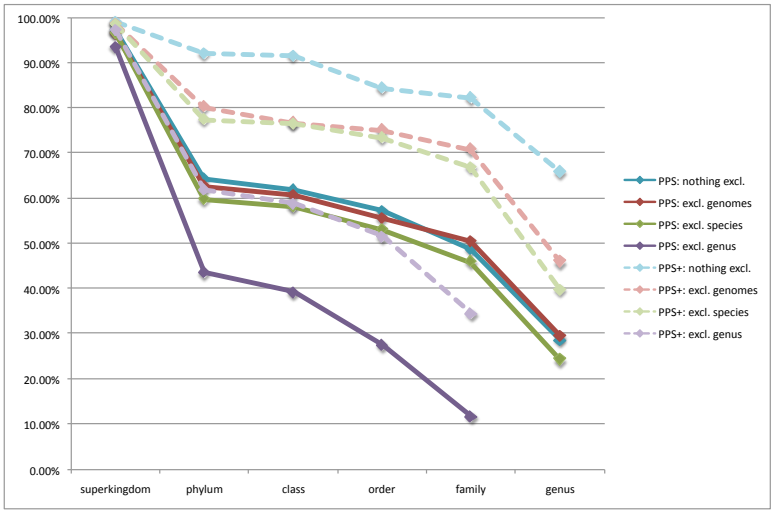
Figure 8.5: Impact of markergene database in contrast to the NCBI database on precision and recall.

all related sequences on the genus level, which corresponds to the case where we want to classify an organism of a new genus, showed the most drastic decrease in performance.

The markergene databases are more extensive as markergenes such as the 16S rRNA gene are much more frequently sequenced. Thus we additionally contrasted the impact of the markergene database against the NCBI database. We only excluded the genome sequences from the markergene database and then successively removed the species and the genus level from the NCBI database. And conversely, we excluded only the genome sequences from the NCBI database and then successively removed the species and the genus level from the markergene database. The results are shown in Figure 8.5. For the precision as well as for the recall we can only see a slight improvement if the sequences on



(a) Precision PPS



(b) Recall PPS

Figure 8.6: Taxonomic benchmarking results for PPS across various scenarios with a generic model and compared to PPS+ where a sample specific model is derived based on the markergenes.

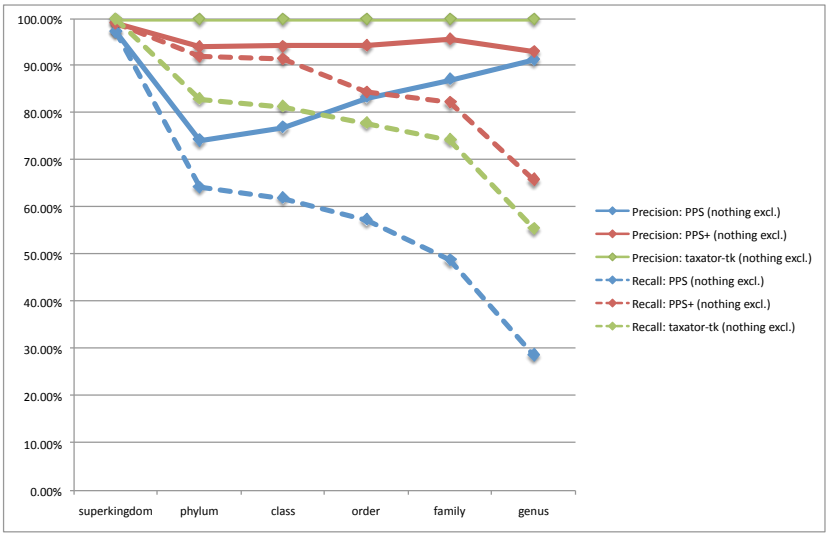


Figure 8.7: Comparison of PPS+, taxator-tk and PPS for the idealistic scenario where nothing was filtering from the databases.

the species level of the sought-after genomes are present in one or both of the databases. When we are looking for a new genus though we can see a significant improvement (≈10%) if the sequences on the genus level are only missing from one of the databases. The graphs also show that overall it is more important that the sequences are present in the markergene database (as shown by the green dashed line).

Figure 8.6 shows the results for PhyloPythiaS with a generic model. We tested four scenarios: In the first case nothing was excluded from the database (PPS only utilises the NCBI database). We then successively excluded the genomes, the species and the genus from the database. The graphs show clearly the improvement achieved with PPS+ (plotted as dashed lines) where sample specific data based on the markergenes is used during the training phase to infer the model.

We also ran taxator-tk on the contigs for the ideal scenario where nothing was filtered out from the databases, i.e. all sought-after organisms are present in the database. The last graph of this section (Figure 8.7) compares the results of PPS, PPS+ and taxator-tk. Taxator-tk returned very reliable predictions as indicated by the perfect precision rate. PPS+ outperformed taxator-tk in terms of recall which is most likely due to the tendency of taxator-tk to make conservative predictions. Both, taxator-tk and PPS+, showed a significant improvement over PPS.

8.6 Conclusion and Future Work

Our initial evaluation showed that the programs PhyloPythiaS+ and taxator-tk are able to achieve a significant improvement in terms of precision and recall over an established

program for taxonomic classification. We were also able to show that each method excelled in a different area. For example, PPS+ outperformed taxator-tk in terms of recall and was able to correctly classify a larger fraction of the "true" bins. On the other hand, taxator-tk was able to achieve an almost perfect precision rate for the ideal scenario where the sought-after genomes are present in the databases.

For the publication of PPS+ the validation was extended to two simulated data sets and two real metagenomic data sets. PPS+ was compared to PhyloPythia, PhyloPythiaS, MEGAN4 [78] and taxator-tk. I generated the simulated data sets using my read simulation program (see Chapter 3). Both simulated data sets are based on GAII error profiles where the standard library preparation method was used. The insert size distribution was also based on an experimental data set.
For each *in silico* data set 15 million paired-end reads of 90bp were generated with an average insert size of 291bp. The first 10bp of the 100bp reads in the experimental data set were trimmed due to fluctuations in the nucleotide distributions at the starting positions which indicated partial remains of the barcode sequence. My read simulation program outputs fasta format, which was converted into a pseudo FASTQ format for the downstream analysis with uniformly high quality scores. I then assembled the reads with Metassembler [12] using Velvet [164], run with different kmer sizes ranging between 19-75 and subsequently merged with Minimus2 [148]. This assembly yielded better results than SOAPdenovo2 [101], MetaVelvet [114] and Newbler [104] in terms of number of contigs above 1,000bp and their total summed length. As PPS and PPS+ require contigs of at least 1,000bp only contig sequences longer than this threshold were considered for the analysis. The contigs were subsequently mapped with BLAST [29] onto the reference genomes to obtain contig labels. Table 8.1 provides a brief overview of the data sets.

Distribution	contigs	MB
Uniform	14,393	13
Log-norm	13,284	66

Table 8.1: Overview of assembled simulated data sets.

The extended benchmarking study confirmed that PPS+ can achieve higher overall precision and recall than PPS used with the generic model and PPS+ outperformed MEGAN4 in most scenarios in terms of precision and recall. Taxator-tk performed best in terms of precision but assigned substantially fewer sequences to low taxonomic ranks. In general, a substantial increase in correct assignments to low taxonomic ranks was observed for PPS+ compared to methods that are based on detecting sequence similarity (e.g. MEGAN4, taxator-tk). In addition PPS+ produced very few false positives. PhymmBL [43], CARMA3 [65] and SOrt-ITEMS [70] were not included due to prohibitive runtimes.

Future work will include the testing of additional Illumina platforms as well as new sequencing technologies such as PacBio to evaluate the potential of longer reads with higher error rates. Also, due to the decline in sequencing cost, more data is becoming available resulting in more extensive and more accurate databases, at the same time the assembly programs are becoming more precise. This should be followed by improved predictions of the taxonomic assignment programs. In addition, other programs such as metAMOS [153], CREST [91] and Metaphlan [144] could be included in the comparison.

9 A Collapsed Variational Dirichlet Process Mixture Model for Error Correction in Illumina Data

9.1 Abstract

Error correction is a crucial step during the analysis of sequencing data. Noise mistaken for diversity results in incorrect conclusions, vast overestimation of sample diversity and misidentification of organisms. Therefore, the recognition and removal of systematic errors is important for the analysis and the accurate interpretation of sequencing data. Although there are many noise removal programs available there is currently still no well-established method for error correction in Illumina data due to a limited knowledge of the biases and systematic errors in Illumina sequencing. Also, it has proven challenging for algorithms to accommodate the vast amounts of data that these technologies produce. We developed an algorithm based on a collapsed variational Dirichlet process mixture model (DPMM) for correcting errors in Illumina amplicon data. DPMMs are a class of Bayesian nonparametric models which offer a great degree of flexibility with a countably infinite number of possible clusters without the problem of overfitting. The number of true underlying sequences (clusters) is automatically determined and the variational inference provides an efficient approximation facilitating computations for the ever increasing amounts of sequencing data. In addition, our approach uses a position and nucleotide specific error model to accommodate the peculiarities of Illumina error patterns identified in a large *in vitro* study.

Original Contributions

I derived the noise removal algorithm based on a collapsed variational Dirichlet process mixture model introduced by Kurihara et al. [88]. This included the formulation of a model appropriate for sequencing data as well as a model for accommodating Illumina biases and errors patterns. In addition, I derived the update equations for the variational inference in the context of a multinomial distribution. Dr Keith Harris contributed to the project with discussions and advice on the calculations.

9.2 Introduction to Dirichlet Processes and Variational Bayes

This chapter introduces some basic definitions for my model including Dirichlet distributions and their generalisation to Dirichlet processes (DP). This is followed by a short description of Dirichlet process mixture models (DPMM) and the Chinese restaurant process, which is a popular representation for DPMMs. I end this section with a short introduction to variational inference (VI) which is the approximation method I used for my algorithm. [40] [41] [88] [17]

Dirichlet distributions

A Dirichlet distribution is a probability distribution of probability distributions. For example, rolling a dice can be described as a multinomial distribution where $p_1, \ldots, p_6$ specify the probabilities for obtaining a $1, \ldots, 6$, respectively. A Dirichlet distribution describes the probability for encountering such a distribution. Dirichlet distributions are finite dimensional and draws from a Dirichlet distribution return a discrete distribution. They are often used as a prior in Bayesian statistics as they impose few restrictions and are conjugate to the multinomial distribution.

More precisely, $(x_1, \ldots, x_K)$ are Dirichlet distributed

$$(x_1, \ldots, x_K) \sim Dir\,(\alpha_1, \ldots, \alpha_K)$$

with order $K \geq 2$ and parameters $\alpha_1, \ldots, \alpha_K > 0$ if the probability density function (pdf) is of the form:

$$f\,(x_1, \ldots, x_K; \alpha_1, \ldots, \alpha_K) = \frac{1}{B\,(\alpha)} \prod_{i=1}^{K} x_i^{\alpha_i - 1}$$

The normalising constant $B(\alpha)$ is the multinomial beta function and $x_1, \ldots, x_K \in (0, 1)$ with $||x||_1 = 1$. The parameter $\alpha = \alpha_1, \ldots, \alpha_K$ is a concentration parameter, which allows the incorporation of prior information and to increase or decrease the likelihood of certain components. If no prior information is available a symmetric Dirichlet distribution with $\alpha_1 = \ldots = \alpha_K$ can be chosen where all components are equally likely.

The pdf of a K-dimensional Dirichlet distribution can be visualised as the open (K-1)-dimensional probability simplex in $\mathbb{R}^{K-1}$. Figure 9.1 demonstrates the impact of the parameter α for K=3.

A Dirichlet distribution has many useful properties such as:

1. Combining entries preserves the Dirichlet property

$$\begin{aligned} & (\pi_1, \ldots, \pi_K) \sim Dir\,(\alpha_1, \ldots, \alpha_K) \\ \Rightarrow \quad & (\pi_1 + \pi_2, \pi_3, \ldots, \pi_K) \sim Dir\,(\alpha_1 + \alpha_2, \alpha_3, \ldots, \alpha_K) \end{aligned}$$

2. And more generally, if $(I_1, \ldots, I_j)$ is a partition of $(1, \ldots, K)$, then:

$$\left(\sum_{i \in I_1} \pi_i, \ldots, \sum_{i \in I_j} \pi_i \right) \sim Dir \left(\sum_{i \in I_1} \alpha_i, \ldots, \sum_{i \in I_j} \alpha_i \right)$$

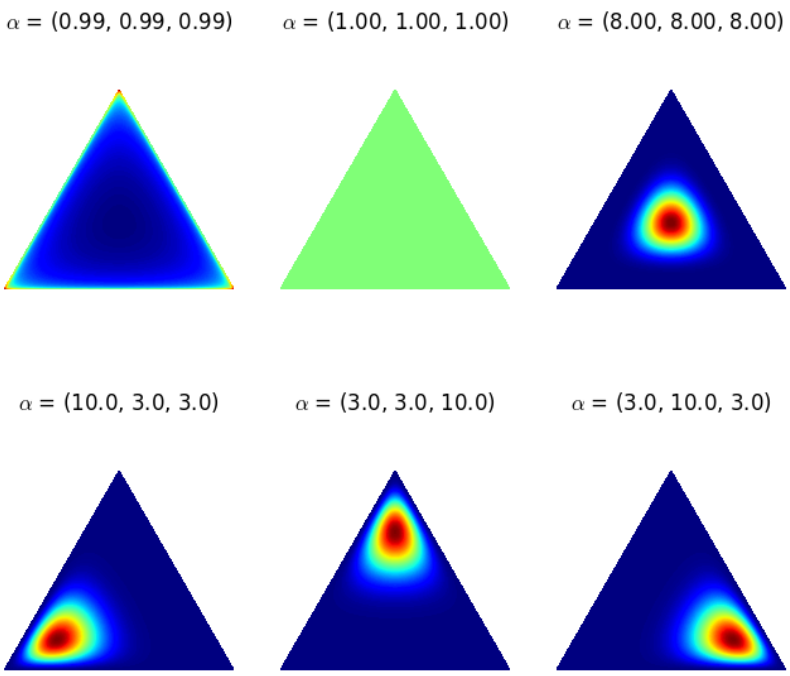


Figure 9.1: Dirichlet distribution for K=3. Each point in the 2-simplex represents a draw from a Dirichlet distribution of order 3 and the colour indicates the likelihood of the particular point. Red corresponds to a large value and blue corresponds to a small value. The larger the parameter α, the more concentrated the probability density function and the symmetric Dirichlet distribution with α=(1,1,1) is a uniform distribution over the 2-simplex.

3. Conversely, if $(\pi_1, \ldots, \pi_K) \sim Dir(\alpha_1, \ldots, \alpha_K)$ and $(\tau_1, \tau_2) \sim Dir(\alpha_1\beta_1, \alpha_1\beta_2)$ with $\beta_1 + \beta_2 = 1$, then:

$$(\pi_1\tau_1, \pi_1\tau_2, \pi_2, \ldots, \pi_K) \sim Dir(\alpha_1\beta_1, \alpha_1\beta_2, \alpha_2, \ldots, \alpha_K)$$

Dirichlet processes

Dirichlet processes (DP) are a class of Bayesian nonparametric models with a very large or infinite number of parameters. It is called a Dirichlet process as its marginal distributions are finite dimensional Dirichlet distributions. Each draw from a DP is a discrete distribution but one that cannot be described by a finite number of parameters. Traditional parametric models use a fixed and finite number of parameters which can result in problems related to over- and underfitting, if the choice of complexity is not appropriate for the data. Bayesian nonparametric models with unbounded complexity avoid both problems by additionally integrating out parameters. The choice of prior in Bayesian statistics is important as it can limit the scope and type of inferences. In nonparametric models a prior with a wide support is desirable. However, the flexibility of the prior is limited by the need for a tractable posterior distribution. The only

limitation of DPs is that distributions drawn from a DP are discrete, however, their posterior is still tractable.

A Dirichlet process can also be described as the infinite-dimensional generalisation of a Dirichlet distribution:

$$\begin{aligned} 1 &\sim Dir\left(\alpha\right) \\ \left(\pi_1, \pi_2\right) &\sim Dir\left(\frac{\alpha}{2}, \frac{\alpha}{2}\right) & \pi_1 + \pi_2 = 1 \\ \left(\pi_{11}, \pi_{12}, \pi_{21}, \pi_{22}\right) &\sim Dir\left(\frac{\alpha}{4}, \frac{\alpha}{4}, \frac{\alpha}{4}, \frac{\alpha}{4}\right) & \pi_{i1} + \pi_{i2} = \pi_i \\ &\vdots \end{aligned}$$

At first we only consider a point at 1 that is Dirichlet distributed with parameter α. We can obtain the infinite-dimensional generalisation by repeatedly applying property 3 described above. First we draw $(\tau_1, \tau_2) \sim Dir\left(\frac{\alpha}{2}, \frac{\alpha}{2}\right)$ and split the point 1 into π_1 and π_2 such that $1 = \pi_1 + \pi_2$ by setting $\pi_1 = 1 \cdot \tau_1$ and $\pi_2 = 1 \cdot \tau_2$. Therefore, $(\pi_1, \pi_2) \sim Dir\left(\frac{\alpha}{2}, \frac{\alpha}{2}\right)$. Next we split π_1 into π_{11} and π_{12} such that $\pi_1 = \pi_{11} + \pi_{12}$. Analogously, we get $\pi_2 = \pi_{21} + \pi_{22}$. Again, $(\pi_{11}, \pi_{12}, \pi_{21}, \pi_{22}) \sim Dir\left(\frac{\alpha}{4}, \frac{\alpha}{4}, \frac{\alpha}{4}, \frac{\alpha}{4}\right)$. This process is repeated indefinitely and results in a Dirichlet process.

A more formal definition is as follows: G is a Dirichlet-process-distributed random probability measure with base distribution H and strength parameter α

$$G \sim DP\left(\alpha, H\right)$$

if for any finite set of partitions of X

$$A_1 \dot{\cup} \ldots \dot{\cup} A_K = X$$

we have $\left(G\left(A_1\right), \ldots, G\left(A_K\right)\right)$ is Dirichlet distributed with

$$\left(G\left(A_1\right), \ldots, G\left(A_K\right)\right) \sim Dir\left(\alpha H\left(A_1\right), \ldots, \alpha H\left(A_K\right)\right)$$

This yields the following expectation

$$\mathbb{E}[G(A)] = H(A)$$

and variance

$$\mathbb{V}[G(A)] = \frac{H(A)(1 - H(A))}{\alpha - 1}$$

where A is any measurable subset of X. Also note, that the larger the strength parameter α the smaller the variance.

For our application, we take advantage of the DP clustering property. Suppose G $\sim$ DP(α,H) is a random probability measure over X. Then G can be treated as a discrete distribution over X and is of the form:

$$G = \sum_{k=1}^{\infty} \pi_k \delta_{\theta^{k*}}$$

which is a possibly infinite sum of point masses with weight π_k at the point θ^{k*}. We can now draw samples $\theta^1, \ldots, \theta^n \sim G$. Any θ^i that take on the same value θ^{k*} are assigned to the same cluster.

Next, we compute the posterior distribution. Suppose G is DP-distributed

$$G \sim DP(\alpha, H)$$

and we draw a random variable θ from G

$$\theta \sim G$$

We can then compute the marginal distribution by integrating out G and the posterior distribution of G given θ:

$$p(\theta) = \int p(\theta|G)\, p(G)\, dG \qquad \text{(marginal distribution)}$$

$$p(G|\theta) = \frac{p(\theta|G)\, p(G)}{p(\theta)} \qquad \text{(posterior)}$$

Now, we take advantage of the conjugacy between Dirichlet distributions and multinomial distributions. Suppose:

$$(\pi_1, \ldots, \pi_K) \sim Dir(\alpha_1, \ldots, \alpha_K)$$

$$z|(\pi_1, \ldots, \pi_K) \sim Discrete(\pi_1, \ldots, \pi_K)$$

where $P(z = k) = \pi_k$.

Then the marginal and posterior distribution are as follows:

$$z \sim Discrete\left(\frac{\alpha_1}{\sum_i \alpha_i}, \ldots, \frac{\alpha_K}{\sum_i \alpha_i}\right) \qquad \text{(marginal)}$$

$$(\pi_1, \ldots, \pi_K)|z \sim Dir(\alpha_1 + \delta_1(z), \ldots, \alpha_K + \delta_K(z)) \qquad \text{(posterior)}$$

where $\delta_i(z) = 1$ if z takes on value i and 0 otherwise.

The conjugacy is also true for Dirichlet processes. Suppose, we fix a partition $(A_1, \ldots, A_K)$ of X. Then:

$$(G(A_1), \ldots, G(A_K)) \sim Dir(\alpha H(A_1), \ldots, \alpha H(A_K))$$

$$p(\theta \in A_i | G) = G(A_i)$$

Using the Dirichlet-multinomial conjugacy we obtain:

$$p(\theta \in A_i) = H(A_i)$$

$$(G(A_1), \ldots, G(A_K))|\theta \sim Dir(\alpha H(A_1) + \delta_\theta(A_1), \ldots, \alpha H(A_K) + \delta_\theta(A_K))$$

So choosing a very fine partition results in:

$$p(\theta)\, d\theta = H(d\theta) \qquad \text{(marginal)}$$

$$G|\theta \sim DP\left(\alpha + 1, \frac{\alpha H + \delta_\theta}{\alpha + 1}\right) \qquad \text{(posterior)}$$

Now, we consider a popular representation of the Dirichlet process: the **Chinese restaurant process** (CRP). The CRP returns a partition B_n of the set $\{1, 2, 3, \ldots, n\}$ at any time $n \in \mathbb{N}$ according to the following probability distribution:

- For $n = 1$ we have the trivial partition $\{\{1\}\}$ with probability 1
- At time $n + 1$ the element $n + 1$ is added:
  → either, to one of the existing blocks b of the partition B_n, where each existing block has probability
  $$\frac{|b|}{n + \alpha}$$
  → or the element forms a new block with probability
  $$\frac{\alpha}{n + \alpha}$$

This process can be pictured as a Chinese restaurant with infinitely many tables where each table can accommodate infinitely many customers. The first customer is seated at an empty table. Every costumer after that will sit at one of the already occupied tables with probability proportional to the number of people at the particular table or the costumer will sit at a new table with probability $\frac{\alpha}{n+\alpha}$.

DP Mixture Model

Next, we will consider Dirichlet process mixture models (DPMM) and their clustering property. Suppose we want to model a set of data points $\{x_1, \ldots, x_n\}$. We denote the

corresponding latent parameters with $\{\theta^1, \ldots, \theta^n\}$ where each θ is drawn from a Dirichlet distribution G. Therefore, $x_i \sim F(\theta^i)$. Technically, any $G \sim DP(\alpha, H)$ is a discrete distribution and has therefore no probability density function. We can overcome this limitation by convolving G with a smooth distribution F:

$$F_x(.) = \int F(.|\theta)\, dG(\theta)$$

F_x in turn is then a smooth distribution and has therefore a pdf. We then draw $x_i \sim F_x$.

Since G is of the form $G = \sum_{k=1}^{\infty} \pi_k \delta_{\theta^{k*}}$ we get the following mixture distribution:

$$F_x(.) = \sum_{k=1}^{\infty} \pi_k F\left(.|\theta^{k*}\right)$$

with mixing proportions π and a countably infinite number of mixing components F.

We can now rewrite the above model as follows:
For every $x_i \sim F_x$, we draw

$$z_i \sim Discrete(\pi)$$

where z_i indicates the cluster to which x_i is assigned. Then:

$$\theta^i = \theta^{z_i*} \qquad \text{and} \qquad x_i|z_i \sim F\left(.|\theta^i\right) = F\left(.|\theta^{z_i*}\right)$$

The potential number of clusters is infinite, though the number of active clusters is always finite and the number of observed data points x_i presents an upper bound. As cluster assignments are chosen proportionally to the number of data points already assigned to the clusters, the number of active clusters is usually much smaller than the number of observations. Therefore, the number of clusters is automatically determined during computations which constitutes one of the major advantages of DPMMs.

Variational Inference

Dirichlet process mixture models (DPMM) have become a popular method for clustering data as they facilitate automatic determination of the appropriate number of clusters. Current methods often use Gibbs sampling, a Markov Chain Monte Carlo (MCMC) method, to approximate the DPMM. In many cases though, this is not efficient enough to cope with the ever increasing amounts of data and it can be difficult to evaluate the convergence of the Markov Chain.

Variational inference (VI) is a family of techniques for approximating intractable integrals. VI is a deterministic approximation and will therefore never return the exact

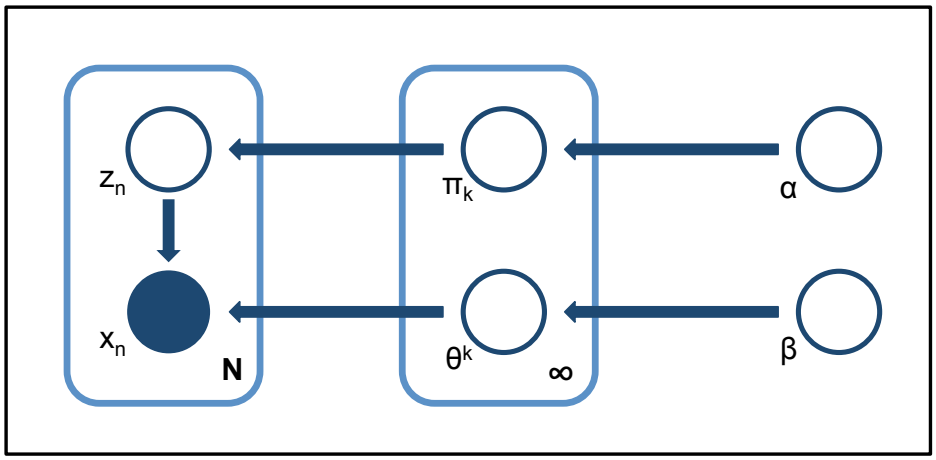


Figure 9.2: Graphical model for DPMM: $\{x_1, \ldots, x_N\}$ are the observed data points with parameters θ^k and $\{z_1, \ldots, z_N\}$ are the latent variables with parameters π_k. α and β denote the hyperparameters.

result, however, it scales much better than stochastic MCMC approximations. For VI we assume that the variational distribution factorises between the latent variables and the parameters.

Suppose, we have N independent, identically distributed data points $X = \{x_1, \ldots, x_N\}$ and latent variables $Z = \{z_1, \ldots, z_N\}$. The joint probability distribution is given by $p(X, Z)$, where X comprises all observed variables and Z comprises all latent variables and parameters (see Figure 9.2). We now want to approximate the posterior distribution for $p(Z|X)$ and $p(X)$ with a variational distribution q. The difference between p and the approximate distribution q is measured by the Kullbeck-Leibler (KL) divergence:

$$ln\, p(X) = \mathcal{L}(q) + KL(q||p)$$

where

$$KL\,(q||p) = -\int q\,(Z)\; ln\left(\frac{p\,(Z|X)}{q(Z)}\right)\; dZ$$

And the **lower bound** is defined as

$$\mathcal{L}\,(q) = \int q\,(Z)\; ln\left(\frac{p(X,Z)}{q(Z)}\right)\; dZ$$

As the true posterior is often intractable, it is necessary to restrict the distribution $q(Z)$ to a family of distributions and then maximise the lower bound $\mathcal{L}(q)$ to obtain an approximation of the posterior. In the following we will assume that the hidden variables and parameters are independent, i.e. the distribution $q(Z)$ factorises. (This approach is also known as mean field theory.) We will optimise the posterior with respect to each of the factors in turn. The lower bound cannot decrease between iterations and

hence convergence is guaranteed (for details see [40]). Note, that the assumption of independence of the hidden variables is only a limitation for the variational distribution and not for the true distribution p.

This yields the following optimum solution:

$$q_j^*(Z_j) = \mathbb{E}_{i \neq j}\left(ln \underbrace{p(X,Z)}_{\text{jnt prob. distr.}} \right) + const \tag{14}$$

9.3 A Collapsed Variational Dirichlet Process Mixture Model

In this section we describe the collapsed variational Dirichlet process mixture model formulated by Kurihara et al. [88] using a standard variational inference approximation. They proposed a finite symmetric Dirichlet approximation to the DP where the mixture weights are integrated out. This permits exchangeable cluster labels (i.e. random permutations of the cluster labels have no effect on the probability of the data).

For the approximation we assume a finite (but large) number of clusters, which we denoted by K. A symmetric Dirichlet distribution is chosen as a prior for the cluster weights:

$$\pi \sim Dir\left(\pi; \frac{\alpha}{K}, \ldots, \frac{\alpha}{K}\right)$$

$$p(\pi) = Dir(\pi|\alpha) = \underbrace{C(\alpha)}_{\text{norm.}} \prod_{k=1}^{K} \pi_k^{\alpha-1}$$

The parameter α can be interpreted as the prior number of observations per cluster. If we choose α to be small (i.e. $\alpha < 1$) then more emphasis is put on the observations and less on the prior. The prior cluster size is on average the same for all components, which illustrates that cluster labels are exchangeable under this prior.

The joint probability distribution for our model (see Figure 9.2) is as follows [88]:

$$P(X, z, \pi, \theta) = \left[\prod_{n=1}^{N} p\left(x_n|\theta^{z_n}\right)\ p\left(z_n|\pi\right)\right]\ \left[\prod_{i=1}^{K} p\left(\theta^i\right)\right]\ Dir\left(\pi; \frac{\alpha}{K}, \ldots, \frac{\alpha}{K}\right)$$

where $X = \{x_1, \ldots, x_N\}$ are the observed data points, $Z = \{z_1, \ldots, z_N\}$ are the cluster assignment variables, π are the mixture weights and θ denote the cluster parameters.

Marginalising out the mixture weights yields:

$$P(X,z,\theta) = \left[\prod_{n=1}^{N} p\left(x_n|\theta^{z_n}\right)\right] \frac{\Gamma(\alpha)\prod_{k=1}^{K}\Gamma\left(N_k+\frac{\alpha}{K}\right)}{\Gamma(N+\alpha)\,\Gamma\left(\frac{\alpha}{K}\right)^K} \left[\prod_{i=1}^{\infty} p\left(\theta^i\right)\right]$$

with $N_k = \sum_{n=1}^{N} \mathbb{I}(z_n = k)$.

Now, the lower bound can be approximated by using the assumption that the hidden variables (denoted by Z) and the parameters (denoted by θ) are independent. This yields:

$$\mathcal{L}(X) \geq \mathcal{B}(X) = \sum_z \int \underbrace{Q(z)Q(\theta)}_{\text{will be optim. in turns}} \; ln\frac{P(X,z,\theta)}{Q(z)Q(\theta)} \; d\theta$$

With the prior:

$$Q(z,\theta) = \left[\prod_{n=1}^{N} q\left(z_n\right)\right]\left[\prod_{k=1}^{K} q\left(\theta^k\right)\right]$$

Approximation is then achieved by alternating optimisation over $Q(z)$ and $Q(\theta)$.

$\mathcal{B}(X)$ can be converted to the form:

$$\mathcal{B}(X) = \sum_{n=1}^{N}\sum_{z_n=1}^{K} \int q(z_n)q(\theta^{z_n}) ln\; p\left(x_n|\theta^{z_n}\right)\; d\theta^{z_n} + \sum_{i=1}^{K}\int q\left(\theta^i\right)\; ln\frac{p(\theta^i)}{q(\theta^i)}\, d\theta^i \tag{15}$$

$$- \sum_{n=1}^{N}\sum_{z_n=1}^{K} q(z_n)\; ln\; q(z_n) + \sum_{z=1}^{K}\left[\prod_{n=1}^{N} q(z_n=k)\right]\; ln\; p(z) \tag{16}$$

This results in the following update equations for the variational inference:

Update equation for the cluster parameters:

For $i = 1,\ldots,K$:

$$q\left(\theta^i\right) \propto \underbrace{p\left(\theta^i\right)}_{prior}\; exp\left(\sum_n q(z_n=i)\; ln\; p\left(x_n|\theta^i\right)\right)$$

Update equation for the cluster assignment variables:

For $n = 1, \ldots, N$:
  For $k = 1, \ldots, K$:

$$q(z_n = k) \propto exp\left(\sum_{z_{\neg n}} \prod_{m \neq n} q\left(z_m = k\right)\ ln\ p\left(z_n = k | z_{\neg n}\right)\right) \cdot exp\left(\int q\left(\theta^{z_n}\right)\ ln\ p\left(x_n | \theta^{z_n}\right) d\theta^{z_n}\right)$$

with $p\left(z_n = k | z_{\neg n}\right) = \frac{N_k^{\neg n} + \frac{\alpha}{K}}{N^{\neg n} + \alpha}$, $N_k^{\neg n} = N_k - \mathbb{I}(z_n = k)$ and $N_k = \sum_{n=1}^{N} \mathbb{I}(z_n = k)$.

For the implementation, θ^i and z_n are updated in turn and the lower bound is monitored for convergence. The lower bound can also function as a verification step as it must not decrease from one iteration to the next. Furthermore, running the algorithm multiple times with different starting parameters can be used as an additional step for maximising the lower bound.

9.4 Amplicon Noise Removal Using a Collapsed Variational DPMM

Our noise removal algorithm is designed for Illumina ampilcon data sets. Therefore, the read length is constant and coincides with the length of the true sequences, all reads cover the same region and have the same starting position.

Model description

Let $x_1, \ldots, x_N$ denote the reads, which correspond to the observations in our model, and the cluster assignments are denoted by $z_1, \ldots, z_N \in \{1, \ldots, K\}$, representing the latent variables. For our model all positions on the reads as well as the true sequences are assumed to be independent. This allows us to model the reads/sequences as a product of independent multinomial distributions with individual parameters p_1, p_2, p_3, p_4 (representing A, C, G and T) for each position.

$$Mult\ (n_1, n_2, n_3, n_4 \mid p_1, p_2, p_3, p_4, N) = \frac{N!}{n_1!\ n_2!\ n_3!\ n_4!} \prod_{i=1}^{4} p_i^{n_i}$$

where N is the total number of reads and $n_1, \ldots, n_4$ the number of occurrences of A, C, G and T, respectively.

The base distribution G_0, which describes the space of all possible true sequences, is

a product of independent symmetric Dirichlet distributions with the same parameter $\beta = (\beta_1, \beta_2, \beta_3, \beta_4)$ (for A, C, G and T). The cluster parameters $\theta^1, \ldots, \theta^K$ represent the true sequences from which the reads originate. Each θ^k is of the following form:

$$\underbrace{\theta^k}_{\text{matrix}} = \underbrace{\left(\theta_j^k\right)_{j=1,\ldots,L}}_{\text{L column vectors}} = \underbrace{\left(\theta_{ij}^k\right)_{i=1,\ldots,4,\ j=1,\ldots,L}}_{\text{4xL matrix entries}}$$

where L is the length of the true sequence. So, the individual positions of each true sequence are represented as distributions. This allows us to account for position and nucleotide specific errors.

In each iteration of the algorithm, we start by updating the cluster parameters $\theta^1, \ldots, \theta^K$ by computing their posterior distributions. This is followed by computing the posteriors of the cluster assignment variables. The reads are then assigned to the cluster with the largest probability and the parameters of the true sequences are updated based on the reads assigned to the respective cluster. If the cluster is empty, we draw a new distribution from G_0.

Update equations for the cluster parameters

In the following, we infer the update equation for the posterior distribution of the cluster parameters $\theta^1, \ldots, \theta^K$. Since we assume position-independence, the variational distribution for each θ^k can be written as the following product:

$$q(\theta^k) \underbrace{=}_{\text{indep. of pos.}} \prod_{j=1}^{L} q\left(\theta_{ij}^k\right)_{i=1,\ldots,4}$$

So for each position $j \in \{1, \ldots, L\}$ of the true sequence associated with cluster $k \in \{1, \ldots, K\}$, we have:

For $i = 1, \ldots, 4$:

$$q\left(\left(\theta_{ij}^k\right)_i\right) \propto \underbrace{p\left(\left(\theta_{ij}^k\right)_i\right)}_{\substack{\text{prior prob.} \\ \text{of cluster}}} \cdot \underbrace{\exp\left(\sum_{n=1}^{N}\left(\underbrace{q\left(z_n = k\right)}_{\text{prob. of } x_n \text{ belong. to cluster k}} \; ln \; \underbrace{p\left(x_{nj} | \left(\theta_{ij}^k\right)_i\right)}_{\text{prob of jth base of read n}}\right)\right)}_{(1)}$$

We can now simplify (1) as follows:

$$(1): \quad \exp\left(\sum_{n=1}^{N}\left(q\left(z_n=k\right)\ ln\ p\left(x_{nj}|\left(\theta_{ij}^{k}\right)_i\right)\right)\right)$$

$$\begin{aligned}
&= \prod_{n=1}^{N}\exp\left(q\left(z_n=k\right)\ ln\ p\left(x_{nj}|\left(\theta_{ij}^{k}\right)_i\right)\right)\\
&\overbrace{=}^{y^x=\exp(x\ ln\ y)} \prod_{n=1}^{N} p\left(x_{nj}|\left(\theta_{ij}^{k}\right)_i\right)^{q(z_n=k)}\\
&= \prod_{n=1}^{N} p\left(\underbrace{x_{nj}}_{\in\{1,2,3,4\}}|\left(\theta_{ij}^{k}\right)_i\right)^{q(z_n=k)}
\end{aligned}$$

$$\begin{aligned}
&\overbrace{=}^{(*)} \prod_{n=1}^{N}\left(\theta_{(x_{nj}),j}^{k}\right)^{q(z_n=k)}\\
&= \left(\theta_{1j}^{k}\right)^{\sum_{\{n \text{ with } x_{nj}=1\}} q(z_n=k)} \cdot \ldots \cdot \left(\theta_{4j}^{k}\right)^{\sum_{\{n \text{ with } x_{nj}=4\}} q(z_n=k)}\\
&= \prod_{i=1}^{4}\left(\theta_{ij}^{k}\right)^{\sum_{\{n \text{ with } x_{nj}=i\}} q(z_n=k)}
\end{aligned}$$

where $(*)\ \ p\left(x_{nj}|\theta_{1j}^{k},\theta_{2j}^{k},\theta_{3j}^{k},\theta_{4j}^{k}\right) = \theta_{(x_{nj}),j}^{k}$

Since $p\left(\left(\theta_{ij}^{k}\right)_i\right)$ is a Dirichlet prior, we can further simplify the update equation:

$$\begin{aligned}
\Rightarrow \quad q\left(\left(\theta_{ij}^{k}\right)_i\right) &\propto \underbrace{p\left(\left(\theta_{ij}^{k}\right)_i\right)}_{=\frac{1}{B(\beta^0)}\prod_{i=1}^{4}\left(\theta_{ij}^{k}\right)^{\beta^0-1}} \cdot \prod_{i=1}^{4}\left(\theta_{ij}^{k}\right)^{\sum_{\{n \text{ with } x_{nj}=i\}} q(z_n=k)}\\
&\propto \frac{1}{B\left(\beta^0\right)}\prod_{i=1}^{4}\left(\theta_{ij}^{k}\right)^{(\sum q(z_n=k))+\beta^0-1}\\
&\propto \prod_{i=1}^{4}\left(\theta_{ij}^{k}\right)^{(\sum q(z_n=k))+\beta^0-1}
\end{aligned}$$

As the prior distribution is conjugate to the multinomial distribution, the posterior is again a Dirichlet distribution with the following updated parameters:

Update equation for cluster parameters:

For $l = 1, \ldots, L$:

$$\left(\beta_i^k\right)^{\text{new}} = \sum_{\{n \text{ with } x_{nj} = i\}} q\left(z_n = k\right) + \beta^0 \qquad \text{for } i \in \{1, \ldots, 4\}$$

Update equation for the cluster assignment variables

For each read x_i with $i \in 1, \ldots, N$ we compute the probability that this read originates from the true sequence associated with cluster k where $k \in 1, \ldots, K$.

For $n = 1, \ldots, N$:

For $k = 1, \ldots, K$:

$$q\left(z_n = k\right) \propto exp\left(\underbrace{\sum_{z_{\neg n}} \prod_{m \neq n} q\left(z_m\right)\ ln\ \underbrace{p\left(z_n = k | z_{\neg n}\right)}_{(2)}}_{(3)}\right) \cdot exp\left(\underbrace{\int q\left(\theta^{z_n = k}\right)\ ln\ p\left(x_n | \theta^{z_n = k}\right) d\theta^{z_n = k}}_{(1)}\right)$$

We can simply (1) as follows:

$$\begin{aligned}
\int q(\theta^k)\ ln\ p\left(x_n | \theta^k\right) d\theta^k &= \prod_{j=1}^{L} \int \underbrace{q\left(\theta_j^k\right)}_{\underbrace{=}_{\text{Dirichlet}}\ \frac{1}{B(\beta^k)} \prod_{i=1}^{4} (\theta_{ij}^k)^{\beta_i^k - 1}}\ ln\ \underbrace{p\left(x_{nj} | \theta_j^k\right)}_{\theta_{x_{nj},j}^k} d\theta_j^k \\
&= \prod_{j=1}^{L} \int \frac{1}{B\left(\beta^k\right)} \prod_{i=1}^{4} \left(\theta_{ij}^k\right)^{\beta_i^k - 1}\ ln\left(\theta_{x_{nj},j}^k\right)\ d\theta_j^k \\
&= \prod_{j=1}^{L} \frac{1}{B\left(\beta^k\right)} \underbrace{\int \prod_{i=1}^{4} \left(\theta_{ij}^k\right)^{\beta_i^k - 1}\ ln\left(\theta_{x_{nj},j}^k\right) d\theta_j^k}_{\int \theta_{1j}^{\beta_1^k - 1} \cdot \theta_{2j}^{\beta_2^k - 1} \cdot \theta_{3j}^{\beta_3^k - 1} \cdot \theta_{4j}^{\beta_4^k - 1} \cdot ln\left(\theta_{x_{nj},j}^k\right) d\theta_j^k}
\end{aligned}$$

$$\overset{(*)}{=} \prod_{j=1}^{L} \frac{1}{B\left(\beta^k\right)} \left(\prod_{i \in 1,\ldots,4; i \neq x_{nj}} \frac{1}{\beta_i^k} \underbrace{\theta_{ij}^{\beta_i^k}}_{(**)} \right) \cdot \left(\frac{1}{\beta_{x_{nj}}^k} \theta_{x_{nj},j}^{\beta_{x_{nj}}^k} \cdot ln\left(\theta_{x_{nj},j}^k\right) - \frac{1}{\left(\beta_{x_{nj}}^k\right)^2} \theta_{x_{nj},j}^{\beta_{x_{nj}}^k} \right)$$

$$= \prod_{j=1}^{L} \frac{1}{B\left(\beta^k\right)} \left(\prod_{i=1}^{4} \frac{1}{\beta_i^k} \theta_{ij}^{\beta_i^k} \right) \cdot \left(ln\left(\theta_{x_{nj},j}^k\right) - \frac{1}{\beta_{x_{nj}}^k} \right)$$

with $(*) \int x^z \cdot ln(x) dx = \frac{1}{z+1} x^{z+1} ln(x) - \frac{1}{(z+1)^2} x^{z+1}$

$(**)$ In the following we will write $\left(\theta_{ij}^k\right)^{\beta_i^k}$ as $\theta_{ij}^{\beta_i^k}$ for better readability.

Part (2) of the equation can be expressed as:

$$p(z_n = k | z_{\neg n}) \overset{\text{see [88]}}{=} \frac{N_k^{\neg n} + \overbrace{\frac{\alpha}{K}}^{\text{symm. Dir. } (*)}}{N^{\neg n} + \alpha}$$

$$= \frac{\sum_{m=1}^{N} \mathbb{I}(z_m = k) - \mathbb{I}(z_n = k) + \frac{\alpha}{K}}{\sum_{k=1}^{K} \left[\sum_{m=1}^{N} \mathbb{I}(z_m = k) - \mathbb{I}(z_n = k) \right] + \alpha}$$

$$= \frac{N_k - \mathbb{I}(z_n = k) + \frac{\alpha}{K}}{\sum_{k=1}^{K} \left[N_k - \mathbb{I}(z_n = k) \right] + \alpha}$$

$(*)$ Here, we will use the initial parameter α.

Note, that $\mathbb{I}$ denotes the indicator function with

$$\mathbb{I}(z_n = k) = \begin{cases} 1, & \text{if } z_n = k \\ 0, & \text{otherwise} \end{cases}$$

We can treat $ln\ (p(z_n = k | z_{\neg n}))$ as a function of $z_{\neg n}$ and apply the *law of the unconscious statistician* (Proposition 4.1 in [135]), which states that for any discrete random variable

X with probability density function f_x

$$\mathbb{E}\left[g(X)\right] = \sum_{x \in X} g(x) f_x(x) = \sum_{x \in X} g(x) P(X = x)$$

Therefore, we can rewrite part (3) as follows:

$$\sum_{z_{\neg n}=1}^{K} \prod_{m \neq n} q(z_m)\ ln\ p\left(z_n = k | z_{\neg n}\right) \ = \ \mathbb{E}\left[ln\ \left(p(z_n = k | z_{\neg n})\right]\right.$$

To approximate this expectation we use the Gaussian approximation described in [88]. As N_i is a sum over Bernoulli variables, it can be approximated by a Gaussian distributions using the central limit theorem. The mean and variance of the Gaussian distribution are given by:

$$\mathbb{E}[N_k] \ = \ \sum_{n=1}^{N} q(z_n = k) \tag{17}$$

$$\mathbb{V}[N_k] \ = \ \sum_{n=1}^{N} q(z_n = k)(1 - q(z_n = k)) \tag{18}$$

We will use the following second order Taylor expansion for our approximation (as described in [88]):

$$\mathbb{E}\left[f(m)\right] \approx f\left(\mathbb{E}[m]\right) + \frac{1}{2} f''\left(\mathbb{E}[m]\right) \mathbb{V}[m]$$

$$\begin{aligned}
\mathbb{E}\left[ln\ \left(p\left(z_n = k | z_{\neg n}\right)\right)\right] &= \mathbb{E}\left[ln\ \left(\frac{N_k^{\neg n} + \frac{\alpha}{K}}{N^{\neg n} + \alpha}\right)\right] = \mathbb{E}\left[ln\ \left(N_k^{\neg n} + \frac{\alpha}{K}\right) - ln\left(N^{\neg n} + \alpha\right)\right] \\
&= \underbrace{\mathbb{E}\left[ln\ \left(N_k^{\neg n} + \frac{\alpha}{K}\right)\right]}_{(1)} - \underbrace{\mathbb{E}\left[ln\left(N^{\neg n} + \alpha\right)\right]}_{(2)}
\end{aligned}$$

(1) $\mathbb{E}\left[ln\ \left(N_k^{\neg n} + \frac{\alpha}{K}\right)\right]$

$$\overbrace{\approx}^{\text{Taylor}}\ ln\left(\mathbb{E}\left[N_k^{\neg n} + \frac{\alpha}{K}\right]\right) \ + \ \frac{1}{2}\ \frac{d^2}{\left(dN_k\right)^2}\left(ln\left(\mathbb{E}\left[N_k^{\neg n} + \frac{\alpha}{K}\right]\right)\right) \ \cdot \ \mathbb{V}\left[N_k^{\neg n} + \frac{\alpha}{K}\right]$$

$$\overbrace{=}^{N_k^{\neg n}=N_k-\mathbb{I}(z_n=k)} \quad ln\left(\mathbb{E}\left[N_k\right]-\underbrace{\mathbb{E}\left[\underbrace{\mathbb{I}(z_n=k)}_{=1}\right]}_{=1}+\frac{\alpha}{K}\right)$$

$$+\frac{1}{2}\,\frac{d^2}{(dN_k)^2}\left(ln\left(\mathbb{E}[N_k]-\mathbb{E}\left[\mathbb{I}(z_n=k)\right]+\frac{\alpha}{K}\right)\right)\cdot\underbrace{\mathbb{V}\left[N_k-\underbrace{\mathbb{I}(z_n=k)+\frac{\alpha}{K}}_{\text{constant}}\right]}_{=\mathbb{V}[N_k]}$$

$$= \quad ln\left(\mathbb{E}[N_k]-1+\frac{\alpha}{K}\right)\;-\;\frac{1}{2}\,\frac{\mathbb{V}[N_k]}{\left(\mathbb{E}[N_k]-1+\frac{\alpha}{K}\right)^2}$$

(2) $\mathbb{E}\left[ln\left(N^{\neg n}+\alpha\right)\right]$ will take on the same value for all $k\in\{1,\dots K\}$ and can thus be absorbed into the normalisation constant.

Combining these results and using the expectation and variance from Equation (17) and (18) yields:

$$\mathbb{E}\left[ln\;\left(p\left(z_n=k|z_{\neg n}\right)\right)\right] \quad = \quad ln\left(\sum_{n=1}^{N}q(z_n=k)-1+\frac{\alpha}{K}\right)-\frac{1}{2}\,\frac{\sum_{n=1}^{N}q(z_n=k)(1-q(z_n=k))}{(\sum_{n=1}^{N}q(z_n=k)-1+\frac{\alpha}{K})^2}$$

Overall this yields:

$$q(z_n=k) \quad \propto \quad exp\left(ln\left(\sum_{n=1}^{N}q(z_n=k)-1+\frac{\alpha}{K}\right)-\frac{1}{2}\,\frac{\sum_{n=1}^{N}q(z_n=k)(1-q(z_n=k))}{\left(\sum_{n=1}^{N}q(z_n=k)-1+\frac{\alpha}{K}\right)^2}\right)$$

$$\cdot exp\left(\prod_{j=1}^{L}\frac{1}{B\left(\beta^k\right)}\left(\prod_{i=1}^{4}\frac{1}{\beta_i^k}\theta_{ij}^{\beta_i^k}\right)\cdot\left(ln\left(\theta_{x_{nj},j}^k\right)-\frac{1}{\beta_{x_{nj}}^k}\right)\right)$$

$$= \quad \left(\sum_{n=1}^{N}q(z_n=k)-1+\frac{\alpha}{K}\right)\cdot exp\left(-\frac{1}{2}\,\frac{\sum_{n=1}^{N}q(z_n=k)(1-q(z_n=k))}{\left(\sum_{n=1}^{N}q(z_n=k)-1+\frac{\alpha}{K}\right)^2}\right)$$

$$\cdot exp\left(\prod_{j=1}^{L}\frac{1}{B(\beta^k)}\left(\prod_{i=1}^{4}\frac{1}{\beta_i^k}\theta_{ij}^{\beta_i^k}\right)\cdot\left(ln\left(\theta_{x_{nj},j}^k\right)-\frac{1}{\beta_{x_{nj}}^k}\right)\right)$$

Update equation for cluster assignment:

For $n = 1, \ldots, N$

For $k = 1, \ldots, K$:

$$q(z_n = k) \propto \left(\sum_{n=1}^{N} q(z_n = k) - 1 + \frac{\alpha}{K}\right) \cdot exp\left(-\frac{1}{2} \frac{\sum_{n=1}^{N} q(z_n = k)(1 - q(z_n = k))}{\left(\sum_{n=1}^{N} q(z_n = k) - 1 + \frac{\alpha}{K}\right)^2}\right) \cdot exp\left(\prod_{j=1}^{L} \frac{1}{B(\beta^k)} \left(\prod_{i=1}^{4} \frac{1}{\beta_i^k} \theta_{ij}^{\beta_i^k}\right) \cdot \left(ln\left(\theta^k_{x_{nj},j}\right) - \frac{1}{\beta^k_{x_{nj}}}\right)\right)$$

Lower bound for monitoring convergence

Starting with the lower bound defined in (15):

$$\mathcal{B}(X) = \sum_{n=1}^{N}\sum_{z_n=1}^{K} \int q(z_n) q(\theta^{z_n}) ln\ p(x_n|\theta^{z_n})\ d\theta^{z_n} + \sum_{i=1}^{K} \int q(\theta^i)\ ln \frac{p(\theta^i)}{q(\theta^i)}\ d\theta^i - \sum_{n=1}^{N}\sum_{z_n=1}^{K} q(z_n)\ ln\ q(z_n) + \sum_{z=1}^{K} \left[\prod_{n=1}^{N} q(z_n = k)\right]\ ln\ p(z)$$

$$= \sum_{n=1}^{N}\sum_{z_n=1}^{K} q(z_n = k) \prod_{j=1}^{L} \underbrace{\int q\left(\theta_j^{z_n}\right) ln\ p\left(x_{nj}|\theta_j^{z_n}\right)\ d\theta_j^{z_n}}_{\overbrace{=}^{prev.calc.} \frac{1}{B(\beta^k)}\left(\prod_{i=1}^{4} \frac{1}{\beta_i^k} \theta_{ij}^{\beta_i^k}\right)\left(ln\left(\theta^k_{x_{nj},j}\right) - \frac{1}{\beta^k_{x_{nj}}}\right)} + \sum_{k=1}^{K}\prod_{j=1}^{L} \int \underbrace{q\left(\theta_j^k\right)}_{\frac{1}{B(\beta^k)} \prod_{i=1}^{4} \theta_{ij}^{\beta_i^k - 1}} ln \underbrace{\frac{p(\theta_j^k)}{q\left(\theta_j^k\right)}}_{= \frac{\frac{1}{B(\beta^0)} \prod_{i=1}^{4} \left(\theta_{ij}^k\right)^{\beta_i^0 - 1}}{\frac{1}{B(\beta^k)} \prod_{i=1}^{4} \left(\theta_{ij}^k\right)^{\beta_i^k - 1}} = \frac{B(\beta^k)}{B(\beta^0)} \prod_{i=1}^{4} \left(\theta_{ij}^k\right)^{\beta_i^0 - \beta_i^k}} d\theta_j^i - \sum_{n=1}^{N}\sum_{k=1}^{K} q(z_n = k)\ ln\ q(z_n = k) + \sum_{z=1}^{K} \left[\prod_{n=1}^{N} q(z_n = k)\right]\ ln \underbrace{p(z)}_{\frac{\Gamma(\alpha) \prod_{k=1}^{K} \Gamma\left(N_k + \frac{\alpha}{K}\right)}{\Gamma(N+\alpha) \Gamma\left(\frac{\alpha}{K}\right)^K}}$$

$$
\begin{aligned}
= & \sum_{n=1}^{N}\sum_{z_n=1}^{K} q(z_n=k)\prod_{j=1}^{L}\frac{1}{B(\beta^k)}\left(\prod_{i=1}^{4}\frac{1}{\beta_i^k}\left(\theta_{ij}^k\right)^{\beta_i^k}\right)\left(ln\left(\theta_{x_{nj},j}^k\right)-\frac{1}{\beta_{x_{nj}}^k}\right) \\
& + \sum_{k=1}^{K}\prod_{j=1}^{L}\underbrace{\int \frac{1}{B(\beta^k)}\prod_{i=1}^{4}\left(\theta_{ij}^k\right)^{\beta_i^k-1} ln\left(\frac{B(\beta^k)}{B(\beta^0)}\prod_{i=1}^{4}\left(\theta_{ij}^k\right)^{\beta_i^0-\beta_i^k}\right) d\theta_j^k}_{(1)} \\
& - \sum_{n=1}^{N}\sum_{k=1}^{K} q(z_n=k)\ ln\ q(z_n=k) + \sum_{z=1}^{K}\left[\prod_{n=1}^{N} q(z_n=k)\right] \\
& \cdot ln\left(\frac{\Gamma(\alpha)\prod_{k=1}^{K}\Gamma\left(N_k+\frac{\alpha}{K}\right)}{\Gamma(N+\alpha)\Gamma\left(\frac{\alpha}{K}\right)^K}\right)
\end{aligned}
$$

Part (1) can be further split into:

$$
\int \frac{1}{B(\beta^k)}\prod_{i=1}^{4}\left(\theta_{ij}^k\right)^{\beta_i^k-1} \underbrace{ln\left(\frac{B(\beta^k)}{B(\beta^0)}\prod_{i=1}^{4}(\theta_{ij}^k)^{\beta_i^0-\beta_i^k}\right)}_{=ln\left(\frac{B(\beta^k)}{B(\beta^0)}\right)+ln\left(\prod_{i=1}^{4}\left(\theta_{ij}^k\right)^{\beta_i^0-\beta_i^k}\right)} d\theta_j^k
$$

$$
= \underbrace{\int \frac{1}{B(\beta^k)}\prod_{i=1}^{4}\left(\theta_{ij}^k\right)^{\beta_i^k-1} ln\left(\frac{B(\beta^k)}{B(\beta^0)}\right) d\theta_j^k}_{(1.1)} + \underbrace{\int \frac{1}{B(\beta^k)}\prod_{i=1}^{4}\left(\theta_{ij}^k\right)^{\beta_i^k-1} ln\left(\prod_{i=1}^{4}\left(\theta_{ij}^k\right)^{\beta_i^0-\beta_i^k}\right) d\theta_j^k}_{(1.2)}
$$

The integral in (1.1) equates to:

$$
\begin{aligned}
\int \frac{1}{B(\beta^k)} & \prod_{i=1}^{4}\left(\theta_{ij}^k\right)^{\beta_i^k-1} ln\left(\frac{B(\beta^k)}{B(\beta^0)}\right) d\theta_j^k \\
& = \frac{1}{B(\beta^k)} ln\left(\frac{B(\beta^k)}{B(\beta^0)}\right) \int\prod_{i=1}^{4}\left(\theta_{ij}^k\right)^{\beta_i^k-1} d\theta_j^k \\
& = \frac{1}{B(\beta^k)} ln\left(\frac{B(\beta^k)}{B(\beta^0)}\right) \prod_{i=1}^{4}\frac{1}{\beta_i^k}\left(\theta_{ij}^k\right)^{\beta_i^k}
\end{aligned}
$$

And the integral in (1.2) can be computed as follows:

$$\frac{1}{B(\beta^k)} \int \prod_{i=1}^{4} \left(\theta_{ij}^k\right)^{\beta_i^k - 1} \; ln\left(\prod_{i=1}^{4} \left(\theta_{ij}^k\right)^{\beta_i^0 - \beta_i^k}\right) \; d\theta_j^k$$

$$= \frac{1}{B(\beta^k)} \int \prod_{i=1}^{4} (\theta_{ij}^k)^{\beta_i^k - 1} \; ln\left(\theta_{1j}^{\beta_1^0 - \beta_1^k} \cdot \theta_{2j}^{\beta_2^0 - \beta_2^k} \cdot \theta_{3j}^{\beta_3^0 - \beta_3^k} \cdot \theta_{4j}^{\beta_4^0 - \beta_4^k}\right) \; d\theta_j^k$$

$$= \frac{1}{B(\beta^k)} \int \prod_{i=1}^{4} (\theta_{ij}^k)^{\beta_i^k - 1} \; (\left(\beta_1^0 - \beta_1^k\right) ln\left(\theta_{1j}^k\right) + \left(\beta_2^0 - \beta_2^k\right) ln\left(\theta_{2j}^k\right) + \left(\beta_3^0 - \beta_3^k\right) ln\left(\theta_{3j}^k\right) + (\beta_4^0 - \beta_4^k) ln\left(\theta_{4j}^k\right)) \; d\theta_j^k$$

$$= \frac{1}{B(\beta^k)} \Bigg[\int \prod_{i=1}^{4} \left(\theta_{ij}^k\right)^{\beta_i^k - 1} \; \left(\beta_1^0 - \beta_1^k\right) ln\left(\theta_{1j}^k\right) d\theta_j^k + \int \prod_{i=1}^{4} \left(\theta_{ij}^k\right)^{\beta_i^k - 1} \; \left(\beta_2^0 - \beta_2^k\right) ln\left(\theta_{2j}^k\right) \; d\theta_j^k + \int \prod_{i=1}^{4} \left(\theta_{ij}^k\right)^{\beta_i^k - 1} \left(\beta_3^0 - \beta_3^k\right) ln\left(\theta_{3j}^k\right) \; d\theta_j^k + \int \prod_{i=1}^{4} \left(\theta_{ij}^k\right)^{\beta_i^k - 1} \; \left(\beta_4^0 - \beta_4^k\right) ln\left(\theta_{4j}^k\right) \; d\theta_j^k \Bigg]$$

$$\overbrace{=}^{\text{prev. calc.}} \frac{1}{B(\beta^k)} \Bigg[\left(\beta_1^0 - \beta_1^k\right) \prod_{i=1}^{4} \frac{1}{\beta_i^k} \theta_{ij}^{\beta_i^k} \; \left(ln\left(\theta_{1j}^k\right) - \frac{1}{\beta_1^k}\right) + \left(\beta_2^0 - \beta_2^k\right) \prod_{i=1}^{4} \frac{1}{\beta_i^k} \theta_{ij}^{\beta_i^k} \; \left(ln\left(\theta_{2j}^k\right) - \frac{1}{\beta_2^k}\right) + \left(\beta_3^0 - \beta_3^k\right) \prod_{i=1}^{4} \frac{1}{\beta_i^k} \theta_{ij}^{\beta_i^k} \; \left(ln\left(\theta_{3j}^k\right) - \frac{1}{\beta_3^k}\right) + \left(\beta_4^0 - \beta_4^k\right) \prod_{i=1}^{4} \frac{1}{\beta_i^k} \theta_{ij}^{\beta_i^k} \; \left(ln\left(\theta_{4j}^k\right) - \frac{1}{\beta_4^k}\right) \Bigg]$$

$$= \frac{1}{B(\beta^k)} \; \prod_{i=1}^{4} \frac{1}{\beta_i^k} \theta_{ij}^{\beta_i^k} \; \sum_{m=1}^{4} (\beta_m^0 - \beta_m^k) \; \left(ln(\theta_{mj}^k) - \frac{1}{\beta_m^k}\right)$$

Lower bound:

$$\mathcal{B}(X) = \sum_{n=1}^{N}\sum_{z_n=1}^{K} q(z_n=k)\prod_{j=1}^{L}\frac{1}{B(\beta^k)}\left(\prod_{i=1}^{4}\frac{1}{\beta_i^k}(\theta_{ij}^k)^{\beta_i^k}\right) ln\left(\theta_{x_{nj},j}^k - \frac{1}{\left(\beta_{x_{nj}}^k\right)^2}\right)$$

$$+ \frac{1}{B(\beta^k)}\prod_{i=1}^{4}\frac{1}{\beta_i^k}(\theta_{ij}^k)^{\beta_i^k}\sum_{k=1}^{K}\prod_{j=1}^{L}\left(ln\left(\frac{B(\beta^k)}{B(\beta^0)}\right)+\sum_{m=1}^{4}(\beta_m^0-\beta_m^k)\left(ln(\theta_{mj}^k)-\frac{1}{\beta_m^k}\right)\right)$$

$$- \sum_{n=1}^{N}\sum_{k=1}^{K} q(z_n=k)\ ln\ q(z_n=k)$$

$$+ \sum_{z=1}^{K}\left[\prod_{n=1}^{N} q(z_n=k)\right]\cdot ln\left(\frac{\Gamma(\alpha)\prod_{k=1}^{K}\Gamma(N_k+\frac{\alpha}{K})}{\Gamma(N+\alpha)\Gamma(\frac{\alpha}{K})^K}\right)$$

9.5 Conclusion and Future Work

We formulated a collapsed variational Dirichlet process mixture model in the context of noise removal for Illumina amplicon data sets. Variational inference is a computationally efficient approximation suitable for high throughput data. No prior knowledge on the sample diversity is necessary as the number of true sequences giving rise to the reads is automatically determined. The model incorporates a flexible error model that is designed to reflect the idiosyncrasies encountered in Illumina data. The position and nucleotide specific amplicon error profiles presented in Chapter 6 have revealed biases introduced by various experimental parameters such as library preparation method and choice of primers. We also recorded an accumulation of errors at certain positions throughout the reads related to motifs triggering substitutions and indels, respectively. A position and nucleotide specific error model has the potential to accommodate the impact of the motifs for any experimental design.

The next steps will include the implementation of the derived algorithm. This will require an efficient alignment algorithm to compute the probability that a read originates from each true sequence in each iteration step. In addition, the algorithm can be generalised to handle metagenomic data sets. Incorporating varying read lengths will also make our error correction algorithm applicable to other sequencing technologies.

10 Conclusion and Future Work Directions

In the following, I reiterate the research objectives of this thesis and outline the major discoveries including their implications, limitations and future work objectives. I also give a brief synopsis of my view on the future of bioinformatics and DNA sequencing as well as imminent bottlenecks and obligations that need to be addressed to ensure the continuing success of research in bioinformatics and genetics. Lastly, I point out the contributions of this thesis to the general research area and further include recommendations for sequencing projects based on our findings.

10.1 Thesis Research Objective: Major Discoveries, Implications, Limitations and Future Work Objectives

Next generation sequencing has enormous potential and the capability to have a major impact on many aspects of our lives. Some of the most fundamental applications can be found in medical research. NGS enables detailed studies on drug resistance in viruses, research on the impact of the microbiota on disease and it can help to trace the origin of acute pathogen outbreaks. These are only a few examples where sequencing opens up new avenues for curing and containing diseases and for the development of new and effective treatments. Other areas of application range from the development of drinking and wastewater treatment systems as well as personal care products and cleaning agents. However, in order to realise the potential of NGS, the data needs to be translated into meaningful and useful information. A thorough understanding of the biases and errors in the sequencing data is crucial in order to achieve this and we need to be aware of the reliability and limitation of results returned by bioinformatic analyses. Tremendous advances in sequencing technologies have resulted in several high throughput platforms capable of creating vast data sets. At the same time, the price of sequencing has significantly dropped, reaching the goal of the $1,000 genome. Therefore, sequencing has become affordable and accessible to many research laboratories and companies and has found applications in more and more research areas. However, the development of bioinformatic analysis tools has not been able to keep pace with the technological advancements. Many programs and algorithms were not designed for the newly emerging technologies and are not capable of handling the complexity encountered in metagenomic data sets.

This study set out to explore the potential of next generation sequencing (NGS) in the context of resolving fine-scale variation in viral quasi-species and microbial communities. I studied the nature of artificial variation in the form of biases and errors in Illumina

data, their effects on the down-stream analysis as well as different error removal and correction approaches. Furthermore, I analysed the ability of different next generation sequencing technologies to resolve true variation in the context of state-of-the-art and established library preparation methods, including new low-input methods.

Error profiles for Illumina sequencing

One of the main objectives of my PhD was to establish a better understanding of biases and errors in Illumina sequencing data and to determine how they affect the validity of results created with currently available bioinformatic programs. Past experiences with Roche 454 sequencing data have demonstrated the importance of error correction, as errors mistaken for true genetic variation result in vast overestimations of sample diversity and can lead to misidentification of organisms. Illumina has now replaced Roche 454 as the market leader in DNA sequencing. However, Illumina errors differ fundamentally from 454 errors. Furthermore, better knowledge of these systematic errors will facilitate the development of more effective error correction approaches. In addition, the availability of novel library preparation methods necessitates detailed studies on the impact of experimental design factors.

In order to gain a solid understanding of the Illumina sequencing technology and the sample preparation process, I conducted a small laboratory based project looking at library preparation for the MiSeq platform. The samples included several bacterial and archaeal mock communities and three environmental samples. I amplified the full length 16S rRNA gene as well as the hypervariable V4 region with polymerase chain reaction (PCR) followed by gel electrophoresis (an example is displayed in Figure 10.1), performed clean-up with AMPure XP beads and precipitation. Libraries were prepared with the NexteraXT kit involving quantification on the BioAnalyzer and fragments size selection on the PippinPrep. The samples were subsequently sequenced on the Illumina MiSeq platform.

Building on this knowledge, I conducted the largest *in vivo* study on error profiles in Illumina amplicon and metagenomic data sets to date. For the first time state-of-the-art library preparation methods and experimental protocols were tested in the context of sequencing errors and bias. The profiles are based on 73 amplicon and 41 metagenomic data sets, respectively. My study confirmed the motif-based nature and revealed that the error-causing motifs in the amplicon data sets highly depend on the choice of library preparation method and primers. Furthermore, I was able to expose biases in connection with the sequencing chemistry as well as the transposon based library preparation methods. The engineered polymerase and ddNTPs that are used for Illumina sequencing introduce a bias that results in the preferential incorporation of ddGTPs. Further, we established a link between the recognition site of the transposomes used for the Nextera

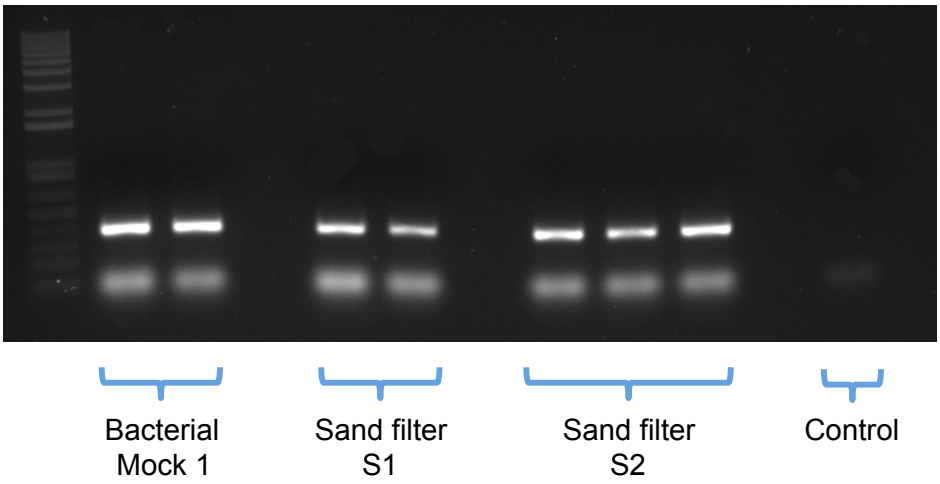


Figure 10.1: Gel electrophoresis image: 1% agarose gel image of seven samples plus control. In all samples the V4 regions was amplified with PCR.

and Parkinson library preparation methods and an uneven nucleotide distribution at the read start of these data sets.

In addition, I explored the efficiency of different error removal techniques. In the case of the amplicon data sets, quality trimming and error correction had a relatively small impact. I attribute this to the fact that these methods are not specifically designed to address the peculiarities encountered in Illumina data and mainly rely on quality scores. However, the quality scores are not able to reflect PCR error and are not able characterise sequencing errors reliably for all data sets. For most of the amplicon data sets the size of the sequenced fragments allowed overlapping of the reads. This approach delivered very good results in terms of error correction, which can be ascribed to the difference in motifs of the forward and reverse reads, which was established for the first time in my *in vivo* study. The combination of quality trimming, error correction and overlapping was identified as the optimal processing strategy and removed on average 94% of the substitution errors in the amplicon data sets. For the majority of the metagenomic data sets (except for the Parkinson data sets) substitution and deletion errors were much better characterised by the quality scores and substitution error rates could on average be reduced by 66% with quality trimming and error correction. These results provide significant knowledge to other researchers with regards to optimal experimental design and data processing strategies.

Although I was able to identify the factors associated with the occurrence of errors, the underlying mechanisms that cause these biases are not apparent for amplicon data sets. Nevertheless, the identification of the biases provides the necessary knowledge to design programs that can handle these peculiarities. An important application is the

development of error correction programs specifically designed for Illumina data. These will be particularly useful for correcting errors if read overlapping is not possible. Future work will also involve the testing of different motif lengths and interactions between the erroneous nucleotide and the motif. Furthermore, although the preferential insertion of the Nextera transposon into certain region of the target DNA was not associated with increased error rates, this could potentially introduce a coverage bias and requires further attention.

Further, I developed a sophisticated Illumina amplicon noise removal algorithm based on a collapsed variational Dirichlet process mixture model that is capable of addressing Illumina specific peculiarities. The algorithm uses a position and nucleotide specific error model that can accommodate the motif-based nature of Illumina errors. Once implemented, the program has the potential to greatly reduce error rates in Illumina amplicon data. The motifs are indirectly modelled by the position and nucleotide specific error distributions. In the future, this approach can also be adapted for metagenomic sequencing in the context of read assembly, though the motif-based nature needs to be directly addressed as reads cover various regions of the template DNA.

Development of simulation tools

The second objective of my PhD was the development of simulation tools that can accurately reflect fine-scale variation. Reliable *in silico* data sets are essential to test and benchmark novel and existing programs in order to reveal their capacities and especially their limitations. Benchmarking studies are indispensable in order to draw valid conclusions based on the analysis. However, many of the current NGS read simulation programs lack the complexity for metagenomic simulations and are not able to simulate the idiosyncrasies encountered in Illumina data. This requires the development of better and more flexible simulation tools that are able to mimic fine-scale variation encountered in real sequencing data. I developed a flexible and efficient read simulation program that focuses on the simulation of Illumina reads for amplicon and metagenomic data sets. This is the only available tool that can directly reflect the connection between experimental design factors, such as library preparation method and choice of primers, and error patterns in Illumina sequencing. Effects due to the strong motif-based nature of Illumina errors need to be tested in connection with any bioinformatic program that aims at processing Illumina data sets. There is currently only one other program available that tries to address these biases [109]. However, the impact of the experimental design factors is not considered and the only error profiles supplied with the program are based on Genome Analyzer II data sets and therefore outdated by current standards. I provide a range of pre-computed profiles that reflect a variety of experimental design factors and facilitate quick and easy simulations of test data sets. In the future, more

biases including the effect of GC content on the genome coverage will be included in the program as well as sample coverage bias to simulate multiplexed sequencing runs.

Updates will be required for the read simulation program as new library preparation methods and novel sequencing technologies become available. The programs, that I developed for the computations of the error profiles, facilitates this in a fast and simple manner for any sequenced mock community with known reference genomes.

In addition, programs intended for the analysis of viral haplotypes are often tested on over-simplified *in silico* data sets. Therefore, algorithms capable of simulating complex viral quasi-species are required. The structure and complexity of viral quasi-species remains a controversial subject. To avoid making assumptions on the structure and complexity of the population, I based the computations on mutations encountered in a real data set. I developed an algorithm that simulates the evolution of a viral quasi-species based on the SNPs encountered in empirical data sets. To my knowledge, this is currently the only available algorithm that is able to simulate the evolution of a quasi-species based on real sequencing data. First, the reads of the experimental data sets are aligned against a reference genome and the mutations at each position are recorded. For the simulation, all positions in the genome are considered successively and mutations get incorporated one at a time. The number of haplotypes in the population is automatically determined. This approach relies on effective error correction of the experimental data prior to the simulation; otherwise errors are mistaken for diversity. I applied my algorithm to an experimental foot-and-mouth virus data sets. Additional filtering steps after trimming the data were necessary to achieve an overall mutation rate that is in accordance with the literature.

Benchmarking studies of viral haplotype reconstruction programs and taxonomic classification tools for metagenomics

Another focus of my thesis was the exploration of the capabilities and limitations of currently available viral haplotype reconstruction programs and taxonomic classification tools for metagenomic data sets. Viral haplotype reconstruction is a key task for enhancing our understanding of life threatening diseases caused by viruses such as the human immunodeficiency virus (HIV) and hepatitis C virus (HCV) and an essential step on the way to developing effective treatments and cures. There are several programs available for reconstructing viral haplotypes from NGS data. However, initial tests showed that the reconstructed haplotype populations returned by the different programs differ immensely. These discrepancies emphasised the need for an independent benchmarking study to expose their true potential. Also, many of the programs are designed for 454 sequencing data and were only tested on over-simplified communities.

I tested the accuracy and completeness of the returned quasi-species for a variety of divergence and complexity levels. The reads as well as the test data sets were created with my simulation tools. I showed that none of the available programs was able to resolve haplotypes with low sequence divergence and the programs failed to detect rare haplotypes. For the first time, the extremely high number of false positives produced by some of the programs was revealed. Missed variants can have disastrous effects on the effectiveness of a treatment if the missed haplotypes are treatment resistant. In contrast, a high number of false positives complicates and possibly prevents the development of treatments. More research on viral haplotype reconstruction methods is required and programs need to incorporate better error correction algorithms to reduce the number of false positives by taking platform specific biases into account. In addition, phylogenetic relationships could be considered to ensure the correctness of the reconstructed haplotypes in the context of a sequenced population. Also the availability of longer reads will facilitate the detection of more variants and increased coverage achieved by current platforms should resolve low frequency haplotypes - as long as effective error correction methods are in place.

I identified read length as one of the key factors for reconstructing haplotypes. Illumina read lengths were limited at the time of our benchmarking study and precluded haplotype reconstruction from Illumina data. New kits enabling reads of up to 2x300bp may be able to change this. I designed a study in collaboration with MRC Centre for Virus Research (University of Glasgow) that will test the suitability of overlapped paired-end Illumina reads. For viral haplotype reconstruction the increased read length of novel sequencing technologies is of particular interest. However, very high error rates currently prevent the direct use of these technologies. In our project, we will also test the potential of combining PacBio reads with Illumina reads to infer viral haplotypes.

For microbial metagenomic data sets, taxonomic classification tools provide important information on the community. However, these programs need to be tested in the context of complex microbial communities. Furthermore, the capabilities of the programs in connection with shorter Illumina reads need to be identified. I conducted a study on taxonomic classification algorithms in collaboration with Alice McHardy's group (University of Dusseldorf) that will assist researchers in choosing the most appropriate program for a particular research question. Taxator-tk makes conservative predictions and is the appropriate choice if accuracy is more important than resolution. However, if the identification of low taxonomic ranks and rare organisms is a key element for a particular project, then PPS+ is more suitable. Future work should will include more programs as well as new sequencing technologies.

10.2 The Future of Bioinformatics and DNA Sequencing

The rapid advances in sequencing technologies have the potential to transform research areas like human genomics and medical diagnostics. However, several imminent bottlenecks and computational challenges need to be addressed. Furthermore, the ability to sequence the human genome brings about obligations with respect to the genetic privacy of indivuals that require due consideration to ensure the continuing success of DNA sequencing.

Medical diagnostic

Some of the most promising applications of metagenomics can be found in clinical microbiology, where metagenomic sequencing could revolutionise the way we detect pathogens and determine optimal treatment strategies [119][56]. Current techniques can be complex and time consuming, involving several independent steps. Further, many of the methods are target-specific and therefore require prior speculations on the cause of the infection or disease outbreak. For any infection or outbreak it is important to identify the pathogen, determine its characteristics, such as virulence and antibiotic resistance properties, and to detect the origin and spread of the pathogen. Current methods such as culturing, microscopy and biochemical reaction still dominate diagnostic bacteriology. These methods lack automation and rely greatly on the experience and knowledge of an individual. Furthermore, they can mostly be used to support initial speculations preventing the detection of unsuspected pathogens as well as interactions of organisms. Also, the majority of bacteria cannot be cultured and in particular viruses are difficult and often impossible to culture, further limiting the range of pathogens that can be detected.

One of the major advantages of sequencing is that no prior assumptions are required. Amplicon sequencing has been available for many years and the sequencing of marker genes, such as the 16S rRNA gene, has provided deep insight into the microbial world. Limitation of amplicon sequencing are related to the use of primers necessary for the amplification step. So-called "universal primers" are available for 16S rRNA sequence amplification, however, they fail to detect all organisms and introduce a bias due to preferential amplification of certain sequences. Furthermore, different primers are required for the amplification of eukaryotes (e.g. targeting the 18S rRNA gene) and there are no universal primers for the 18S rRNA gene or viruses. In addition, this approach only provides information on the presence of organisms but not on their pathogenic potential and susceptibility to different treatments (e.g. antibiotic resistance in the case of bacteria).

Metagenomics has the potential to replace current techniques with a single efficient

workflow. Omitting the culturing step as well as any target-specific amplification offers many advantages including information on a wider range of organisms and the possibility to characterise individual organisms as well as the whole community. This can be important for diseases, such as colon cancer [143] and inflammatory bowel disease [139], where the entire microbiota is assumed to be of importance rather than a single pathogenic organisms. Shotgun metagenomics can detect DNA from bacteria, eukaryotes and viruses simultaneously and no prior knowledge on the cause of the infection or outbreak is required for a metagenomic approach.

For diagnostic virology, metagenomics has already been successfully used to identify unknown pathogens in serious infections and several outbreaks - a task previously hindered by the difficulty associated with culturing viruses and their lack of a universal gene. For example, metagenomics facilitated the identification of a novel Arenavirus in the hospital outbreak of haemorrhagic fever in southern Africa [44] and a novel Ebola virus species was detected in the recent outbreak in Guinea [34]. In addition, Pallen [119] described a range of studies where bacterial pathogens have been successfully detected and identified by metagenomic sequencing. For example, metagenomic sequencing based on DNA extracted from fecal samples has been used to detect bacterial pathogens such as *Campylobacter* and the Shiga-toxigenic *E. coli* strain in the recent outbreak in Germany. This study showed that metagenomics can detect and characterise bacterial pathogens within a sample and also illustrates the suitability of benchtop sequencers for this task.

In summary, metagenomics offers a culture-independent approach for pathogen detection, including bacteria, fungi, viruses and parasites. It offer a target-independent approach where no prior knowledge on the pathogen is required. Metagenomics reveals extensive information about the whole community and detailed information on the individual organisms that can uncover properties such as virulence and antibiotic resistance. For routine application, the price of current methods needs to be further reduced and more automated workflows are required, both for sample preparation and for the subsequent bioinformatic analysis.

Human genomics

Many medical applications arise from the ability to directly sequence the human genome. With recently introduced sequencing technologies the $1,000 genome has become reality. The Illumina HiSeq X is intended for population-scale sequencing and can sequence the 3.2 billion base pairs of the human genome for less than $1,000. Cancer, cystic fibrosis, Down's syndrome and Parkinson's disease are just a few examples of genetic disorders, where such detailed personal genetic information can provide huge benefits and assist in determining the personal risk of a disease including early diagnosis. Furthermore, this information can be used to develop personalised medical care. These newly emerging

sequencing technologies enable large scale studies of genetic disorders and will facilitate new treatment strategies. However, this also gives rise to new concerns and dangers, where data sharing in order to advance research needs to be balanced with genetic privacy. Also, *in vivo* and *in vitro* screening of embryos raises many ethical questions. With great potential comes great responsibility and these issues need to be addressed in the near future.

Computational challenges

One of the great challenges that we are facing nowadays, are bottlenecks due to computational limitations. These limitations are not only related to data processing and a shortage of appropriate tools and programs. The huge data sets that are produced on a daily basis are causing a drastic increase in data storage requirements. It can now take longer to transfer these enormous amounts of data between computers than the actual sequencing process. Furthermore, powerful computers are required to analyse these data sets. While sequencing has outcompeted Moore's law, computer performance generally follows Moore's law by doubling every 18-24 months. This has proven challenging for the design of effective sequencing analysis pipelines. Furthermore, access to state-of-the-art computer equipment is not implicit.

Cloud computing presents a possible solution to circumvent many of these problems. Less data transfer would be required if data is directly uploaded to the cloud by, e.g. the sequencing centres, and data analysis is also performed in the computing cloud. This would allow researchers to access advanced computational resources without making huge investments to acquire computational equipment. Issues related to data security and privacy need be addressed in this context and algorithms may need to be redesigned for parallel computations in the cloud.

Sample preparation for sequencing

Another important step is the library preparation of the samples as all currently available technologies require adequate sample preparation prior to sequencing. New instruments are under development that will facilitate the automatisation of the sample preparation workflow. This will reduce hands-on time and the level of expertise required for the library preparation and offer more coherent results with less contamination and wider availability.

Third generation sequencing technologies

New sequencing technologies are currently emerging. The MinION is a device that is not much bigger then a cell phone and has been previously announced to sell for less than $1,000. The USB-powered sequencer can be directly connected to a computer. This new level of portability combined with cloud computing offers a wealth of opportunities.

Patient sample can be directly sequenced in hospitals and medical practices, reducing the time from sampling to diagnosis and effective treatment. Bacterial and viral infections can be unambiguously determined without the need for speculations based on symptoms. Further, entire human genomes can be sequenced rapidly to identify genetic disorders and to establish a personalised treatment plan. This portability also offers new opportunities for developing countries. Samples can be sequenced on-site, avoiding long transportation and shipping times as well as issues related to sample storage and import regulations. In the case of serious outbreaks of viruses and bacteria short response times are often critical.

The device is not commercially available yet. However, the launch of the early access programme, an initial test phase, is a promising step. Also, error rates for the MinION are currently much higher than for any other sequencing technology. This makes the data adequate for the identification of the organisms but currently the recognition of novel strains and species is out of reach with this technology. In addition, the development of efficient and effective bioinformatic algorithms and tools will be required.

10.3 Conclusion

So far we have only scratched the surface of what sequencing and bioinformatics can accomplish. Ever growing data sets and rapidly evolving sequencing technologies need to be matched by an equally rapid development of sophisticated bioinformatic tools for the analysis as well as strategies for appropriate data storage and processing. Therefore, a greater focus and more research in bioinformatics is essential to facilitate the ongoing success of genomic sequencing.

The popularity and high demand of next generation sequencing and bioinformatics is fuelled by their great potential and their large range of applications. This has also sparked great expectation. However, using the right approach and appropriate tools for the analysis is essential in order to derive valid conclusions. In particular, biases and errors need to be thoroughly considered. The quality of the experimental work has great impact on the value and explanatory power of the data, however, meaningful conclusions can only be drawn if the data is analysed correctly using carefully tested and verified programs and techniques.

It is difficult to predict the impact of all experimental factors, especially for such rapidly developing technologies and experimental kits. In particular for large sequencing projects, the inclusion of a mock community can provide detailed insight into biases and quality of the sequencing data and assist in developing optimal analysis strategies. Further, different analysis strategies should always be taken into consideration prior to the

experimental work. For instance, read overlapping offers great potential for error correction but can only be applied if paired-end reads have a significant overlap; therefore the right fragment sizes need to be selected during the library preparation.

Within this thesis, I developed programs to simulate more realistic *in silico* data sets that will facilitate more rigorous evaluation and testing of novel and existing bioinformatic programs. Furthermore, they can assist in maximising the value of experiments by testing various design strategies prior to the experimental work. My benchmarking studies will aid researchers in choosing the most appropriate programs for a particular question and raise awareness of the limitations of these approaches. In addition, my research on error patterns and biases provides detailed insight into the nature of systematic Illumina errors. This will facilitate optimal analysis strategies and programs for Illumina sequencing data and improve the overall value and significance of the results. I also demonstrated the potential of various error correction and removal techniques, which will enable researchers to choose the optimal approach for their data. Overall, these findings will improve bioinformatic research and analysis and help to realise the promise and potential of next generation sequencing.

11 References

[1] http://454.com/products/gs-flx-system/ (last check May 2014).

[2] http://archive.archaeology.org/1003/etc/neanderthals.html (last checked May 2014).

[3] http://blog.pacificbiosciences.com/2013/01/a-closer-look-at-accuracy-in-pacbio.html (last checked May 2014).

[4] http://bmda.cs.unibas.ch/hivhaplotyper/ (last checked July 2014).

[5] http://eartheasy.com/blog/2011/05/perennial-crops-being-developed-to-produce-food-with-less-environmental-impact/ (last checked July 2014).

[6] http://en.wikipedia.org/wiki/file:sequencing.jpg.

[7] http://files.pacb.com/pdf/pacbio_rs_ii_brochure.pdf (last checked May 2014).

[8] http://mmwfbd.org/index.php/program-item/water-sanitation/ (last checked June 2014).

[9] http://omicsmaps.com/ (last checked June 2014).

[10] https://github.com/lh3/wgsim (last checked July 2014).

[11] https://github.com/najoshi/sickle (last checked July 2014).

[12] http://sourceforge.net/apps/mediawiki/metassembler/index.php?title=metassembler (last checked February 2014).

[13] http://supportres.illumina.com/documents/myillumina/900851dc-01cf-4b70-9e95-d590531c5bd4/nextera_xt_sample_preparation_guide_15031942_c.pdf (last checked April 2014).

[14] https://www3.appliedbiosystems.com/cms/groups/mcb_marketing/documents/generaldocuments/cms_058265.pdf (last checked May 2014).

[15] https://www.nanoporetech.com/technology/introduction-to-nanopore-sensing/introduction-to-nanopore-sensing (last checked July 2014).

[16] https://www.nanoporetech.com/technology/the-minion-device-a-miniaturised-sensing-system/the-minion-device-a-miniaturised-sensing-system (last checked June 2014).

[17] http://videolectures.net/mlss07_teh_dp/ (last check July 2014).

[18] http://www.aaas.org/news/science-study-reveals-when-cattle-can-transmit-foot-and-mouth-disease (last checked July 2014).

[19] http://www.genome.gov/sequencingcosts/ (last checked June 2014).

[20] http://www.hkpr.on.ca/infoset/babieschildren/immunization.aspx (last checked June 2014).

[21] http://www.illumina.com/systems.ilmn (last check July 2014).

[22] http://www.illumina.com/systems/nextseq-sequencer/technology.ilmn (last checked May 2014).

[23] http://www.lifetechnologies.com/uk/en/home/life-science/sequencing/next-generation-sequencing/ion-torrent-next-generation-sequencing-technology.html (last checked July 2014).

[24] http://www.seqll.com/dna-sequencing.html (last checked May 2014).

[25] http://www.vicbioinformatics.com/software.velvetoptimiser.shtml (last checked June 2014).

[26] N. Loman (2014): Wiggle plot showing Oxford Nanopore signal data for a *P. aeruginosa* read. figshare: http://dx.doi.org/10.6084/m9.figshare.1053026 (last checked July 2014).

[27] Nature cover: Volume 486 number 7402.

[28] *The New Science of metagenomics: revealing the secrets of our microbial planet*. The National Academies Press, 2007.

[29] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman. Basic local alignment search tool. *Journal of Molecular Biology*, 215(3):403–410, 1990.

[30] F. E. Angly, D. Willner, F. Rohwer, P. Hugenholtz, and G. W Tyson. Grinder: a versatile amplicon and shotgun sequence simulator. *Nucleic Acids Research*, page 251, 2012.

[31] J. Archer, G. Baillie, S.J. Watson, P. Kellam, A. Rambaut, and D.L. Robertson. Analysis of high-depth sequence data for studying viral diversity: a comparison of next generation sequencing platforms using segminator II. *BMC Bioinformatics*, 13(1):47, 2012.

[32] B. Ason and W. S. Reznikoff. DNA sequence bias during Tn5 transposition. *Journal of Molecular Biology*, 335(5):1213–1225, 2004.

[33] I. Astrovskaya, B. Tork, S. Mangul, K. Westbrooks, I. Mandoiu, P. Balfe, and A. Zelikovsky. Inferring viral quasispecies spectra from 454 pyrosequencing reads. *BMC Bioinformatics*, 12, 2011.

[34] S. Baize, D. Pannetier, L. Oestereich, T. Rieger, L. Koivogui, N. Magassouba, B. Soropogui, M. S. Sow, S. Keita, H. De Clerck, et al. Emergence of *Zaire Ebola* virus disease in Guinea: preliminary report. *New England Journal of Medicine*, 2014.

[35] P. Baldi, S. Brunak, Y. Chauvin, C. A. F. Andersen, and H. Nielsen. Assessing the accuracy of prediction algorithms for classification: an overview. *Bioinformatics*, 16(5):412–424, 2000.

[36] S. Balzer, K. Malde, and I. Jonassen. Systematic exploration of error sources in pyrosequencing flowgram data. *Bioinformatics*, 27(13):i304–i309, 2011.

[37] S. Balzer, K. Malde, A. Lanzen, A. Sharma, and I. Jonassen. Characteristics of 454 pyrosequencing data enabling realistic simulation with flowsim. *Bioinformatics*, 26(18):i420–i425, 2010.

[38] A. Bankevich, S. Nurk, D. Antipov, A. A. Gurevich, M. Dvorkin, A. S. Kulikov, V. M. Lesin, S. I. Nikolenko, S. Pham, A. D. Prjibelski, et al. SPAdes: a new genome assembly algorithm and its applications to single-cell sequencing. *Journal of Computational Biology*, 19(5):455–477, 2012.

[39] D. R. Bentley, S. Balasubramanian, H. P. Swerdlow, G. P. Smith, J. Milton, C. G. Brown, K. P. Hall, D. J. Evers, C. L. Barnes, H. R. Bignell, et al. Accurate whole human genome sequencing using reversible terminator chemistry. *Nature*, 456(7218):53–59, 2008.

[40] C. M. Bishop and N. M. Nasrabadi. *Pattern recognition and machine learning*, volume 1. Springer New York, 2006.

[41] D. M. Blei and M. I. Jordan. Variational inference for Dirichlet process mixtures. *Bayesian Analysis*, 1(1):121–143, 2006.

[42] S. Bonhoeffer and M. A. Nowak. Pre-existence and emergence of drug resistance in HIV-1 infection. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 264(1382):631–637, 1997.

[43] A. Brady and S. L. Salzberg. Phymm and PhymmBL: metagenomic phylogenetic classification with interpolated Markov models. *Nature Methods*, 6(9):673–676, 2009.

[44] T. Briese, J. T. Paweska, L. K. McMullan, S. K. Hutchison, C. Street, G. Palacios, M. L. Khristova, J. Weyer, R. Swanepoel, M. Egholm, et al. Genetic detection and characterization of Lujo virus, a new hemorrhagic fever-associated arenavirus from southern Africa. *PLoS Pathogens*, 5(5):e1000455, 2009.

[45] T. D. Brock, M. T. Madigan, J. M. Martinko, and J. Parker. *Brock biology of microorganisms*. Prentice Hall, 12th edition, 2009.

[46] T. D. Brock, M. T. Madigan, J. M. Martinko, and J. Parker. *Brock biology of microorganisms*. Prentice Hall, 13th edition, 2011.

[47] R. A. Bull, F. Luciani, K. McElroy, S. Gaudieri, S. T. Pham, A. Chopra, B. Cameron, L. Maher, G. J. Dore, P. A. White, et al. Sequential bottlenecks drive viral evolution in early acute hepatitis C virus infection. *PLoS Pathogens*, 7(9):e1002243, 2011.

[48] C. Carrillo, E. R. Tulman, G. Delhon, Z. Lu, A. Carreno, A. Vagnozzi, G. F. Kutish, and D. L. Rock. Comparative genomics of foot-and-mouth disease virus. *Journal of Virology*, 79(10):6487–6504, 2005.

[49] E. Chargaff. Chemical specificity of nucleic acids and mechanism of their enzymatic degradation. *Cellular and Molecular Life Sciences*, 6(6):201–209, 1950.

[50] C. Chen. DNA polymerases drive DNA sequencing-by-synthesis technologies: Both past and present. *Evolutionary and Genomic Microbiology*, 5:305, 2014.

[51] F. Chen, M. Dong, M. Ge, L. Zhu, L. Ren, G. Liu, and R. Mu. The history and advances of reversible terminators used in new generations of sequencing technology. *Genomics, Proteomics & Bioinformatics*, 11(1):34–40, 2013.

[52] F. M. Cohan. What are bacterial species? *Annual Review of Microbiology*, 56(1):457–487, 2002.

[53] A. C. Cumino, C. Marcozzi, R. Barreiro, and G. L. Salerno. Carbon cycling in *Anabaena sp. PCC 7120*. Sucrose synthesis in the heterocysts and possible role in nitrogen fixation. *Plant Physiology*, 143(3):1385–1397, 2007.

[54] L. D'Amore, U. Z. Ijaz, M. Schirmer, N. Hall, and C. Quince. A comprehensive benchmarking study of next-generation sequencing platforms for 16S rRNA community profiling. *(In preparation)*.

[55] M. A. DePristo, E. Banks, R. Poplin, K. V. Garimella, J. R. Maguire, C. Hartl, A. A. Philippakis, G. del Angel, M. A. Rivas, M. Hanna, et al. A framework for variation discovery and genotyping using next-generation DNA sequencing data. *Nature Genetics*, 43(5):491–498, 2011.

[56] X. Didelot, R. Bowden, D. J. Wilson, T. E. A. Peto, and D. W. Crook. Transforming clinical microbiology with bacterial genome sequencing. *Nature Reviews Genetics*, 13(9):601–612, 2012.

[57] J. C. Dohm, C. Lottaz, T. Borodina, and H. Himmelbauer. Substantial biases in ultra-short read data sets from high-throughput DNA sequencing. *Nucleic Acids Research*, 36(16):e105–e105, 2008.

[58] R. Drmanac, A. B. Sparks, M. J. Callow, A. L. Halpern, N. L. Burns, B. G. Kermani, P. Carnevali, I. Nazarenko, G. B. Nilsen, G. Yeung, et al. Human genome sequencing using unchained base reads on self-assembling DNA nanoarrays. *Science*, 327(5961):78–81, 2010.

[59] J. Dröge, I. Gregor, and A. C. McHardy. Taxator-tk: Fast and precise taxonomic assignment of metagenomes by approximating evolutionary neighborhoods. *arXiv preprint arXiv:1404.1029*, 2014.

[60] N. Eriksson, L. Pachter, Y. Mitsuya, S. Y. Rhee, C. Wang, B. Gharizadeh, M. Ronaghi, R. W. Shafer, and N. Beerenwinkel. Viral population estimation using pyrosequencing. *PLoS Comput Biol*, 4(5):e1000074, 2008.

[61] N. Fierer, M. Hamady, C. L. Lauber, and R. Knight. The influence of sex, handedness, and washing on the diversity of hand surface bacteria. *Proceedings of the National Academy of Sciences*, 105(46):17994–17999, 2008.

[62] M. Frampton and R. Houlston. Generation of artificial FASTQ files to evaluate the performance of next-generation sequencing pipelines. *PloS One*, 7(11):e49110, 2012.

[63] M. Fukushima, K. Kakinuma, and R. Kawaguchi. Phylogenetic analysis of *Salmonella*, *Shigella*, and *Escherichia coli* strains on the basis of the *gyrB* gene sequence. *Journal of Clinical Microbiology*, 40(8):2779, 2002.

[64] M. Galassi, J. Davies, J. Theiler, B. Gough, G. Jungman, P. Alken, M. Booth, and F. Rossi. GNU scientific library reference manual (v1. 12). *Network Theory Ltd*, 2009.

[65] W. Gerlach and J. Stoye. Taxonomic classification of metagenomic shotgun sequences with CARMA3. *Nucleic Acids Research*, 39(14):e91–e91, 2011.

[66] B. Ghebremedhin, F. Layer, W. Konig, and B. Konig. Genetic classification and distinguishing of Staphylococcus species based on different partial gap, 16S rRNA, hsp60, rpoB, sodA, and tuf gene sequences. *Journal of Clinical Microbiology*, 46(3):1019, 2008.

[67] T. C. Glenn. Field guide to next-generation DNA sequencers. *Molecular Ecology Resources*, 2011.

[68] S. J. Haig, G. Collins, R. L. Davies, C. C. Dorea, and C. Quince. Biological aspects of slow sand filtration: past, present and future. *Water Science & Technology: Water Supply*, 11(4):468–472, 2011.

[69] K. Hanada, Y. Suzuki, and T. Gojobori. A large variation in the rates of synonymous substitution for RNA viruses and its relationship to a diversity of viral infection and transmission modes. *Molecular Biology and Evolution*, 21(6):1074–1080, 2004.

[70] M. M. Haque, T. S. Ghosh, D. Komanduri, and S. S. Mande. SOrt-ITEMS: Sequence orthology based approach for improved taxonomic estimation of metagenomic sequences. *Bioinformatics*, 25(14):1722–1730, 2009.

[71] E. C. Hayden. Nanopore genome sequencer makes its debut. *Nature*, 10, 2012.

[72] M. R. Henn, C. L. Boutwell, P. Charlebois, N. J. Lennon, K. A. Power, A. R. Macalalad, A. M. Berlin, C. M. Malboeuf, E. M. Ryan, S. Gnerre, et al. Whole genome deep sequencing of HIV-1 reveals the impact of early minor variants upon immune recognition during acute infection. *PLoS Pathogens*, 8(3):e1002529, 2012.

[73] M. Holtgrewe. Mason - a read simulator for second generation sequencing data. *Technical Report FU Berlin*, 2010.

[74] X. Hu, J. Yuan, Y. Shi, J. Lu, B. Liu, Z. Li, Y. Chen, D. Mu, H. Zhang, N. Li, et al. pIRS: profile-based Illumina pair-end reads simulator. *Bioinformatics*, 28(11):1533–1535, 2012.

[75] W. Huang, L. Li, J. R. Myers, and G. T. Marth. ART: a next-generation sequencing read simulator. *Bioinformatics*, 28(4):593–594, 2012.

[76] S. M. Huse, J. A. Huber, H. G. Morrison, M. L. Sogin, D. M. Welch, et al. Accuracy and quality of massively parallel DNA pyrosequencing. *Genome Biology*, 8(7):R143, 2007.

[77] S. M. Huse, D. M. Welch, H. G. Morrison, and M. L. Sogin. Ironing out the wrinkles in the rare biosphere through improved OTU clustering. *Environmental Microbiology*, 12(7):1889–1898, 2010.

[78] D. H. Huson, A. F. Auch, J. Qi, and S. C. Schuster. Metagenome analysis using MEGAN. In *Proceedings of the 5th Asia-Pacific Bioinformatics Conference*, volume 5, pages 7–16, 2007.

[79] Illumina. http://supportres.illumina.com/documents/myillumina/f5f619d3-2c4c-489b-80a3-e0414baa4e89/truseq_dna_sampleprep_guide_15026486_c.pdf (last checked June 2014).

[80] T. Iwase, Y. Uehara, H. Shinji, A. Tajima, H. Seo, K. Takada, T. Agata, and Y. Mizunoe. *Staphylococcus epidermidis Esp* inhibits *Staphylococcus aureus* biofilm formation and nasal colonization. *Nature*, 465(7296):346–349, 2010.

[81] E. Jaspers and J. Overmann. Ecological significance of microdiversity: identical 16S rRNA gene sequences can be found in bacteria with highly divergent genomes and ecophysiologies. *Applied and Environmental Microbiology*, 70(8):4831–4839, 2004.

[82] B. Jia, L. Xuan, K. Cai, Z. Hu, L. Ma, and C. Wei. NeSSM: A next-generation sequencing simulator for metagenomics. *PloS One*, 8(10):e75448, 2013.

[83] W. M. Jou, G. Haegeman, M. Ysebaert, and W. Fiers. Nucleotide sequence of the gene coding for the bacteriophage MS2 coat protein. *Nature*, 237:82–88, 1972.

[84] M. Kircher, U. Stenzel, J. Kelso, et al. Improved base calling for the Illumina Genome Analyzer using machine learning strategies. *Genome Biology*, 10(8):R83, 2009.

[85] D. C. Koboldt, Q. Zhang, D. E. Larson, D. Shen, M. D. McLellan, L. Lin, C. A. Miller, E. R. Mardis, L. Ding, and R. K. Wilson. VarScan 2: somatic mutation and copy number alteration discovery in cancer by exome sequencing. *Genome Research*, 22(3):568–576, 2012.

[86] S. Koren, M. C. Schatz, B. P. Walenz, J. Martin, J. T. Howard, G. Ganapathy, Z. Wang, D. A. Rasko, W. R. McCombie, E. D. Jarvis, et al. Hybrid error correction and de novo assembly of single-molecule sequencing reads. *Nature biotechnology*, 30(7):693–700, 2012.

[87] S. Krishna and L.S. Miller. Host-pathogen interactions between the skin and *Staphylococcus aureus*. *Current Opinion in Microbiology*, 2011.

[88] K. Kurihara, M. Welling, and Y. W. Teh. Collapsed variational Dirichlet process mixture models. In *IJCAI*, volume 7, pages 2796–2801, 2007.

[89] E. S. Lander and M. S. Waterman. Genomic mapping by fingerprinting random clones: a mathematical analysis. *Genomics*, 2(3):231–239, 1988.

[90] P. Langridge and D. Fleury. Making the most of omics for crop breeding. *Trends in Biotechnology*, 29(1):33–40, 2011.

[91] A. Lanzén, S. L. Jørgensen, D. H. Huson, M. Gorfer, S. H. Grindhaug, I. Jonassen, L. Øvreås, and T. Urich. CREST-classification resources for environmental sequence tags. *PloS One*, 7(11):e49334, 2012.

[92] R. S. Lasken and J. S. McLean. Recent advances in genomic DNA sequencing of microbial species from single cells. *Nature Reviews Genetics*, 15(9):577–584, 2014.

[93] T. Le, J. Chiarella, B. B. Simen, B. Hanczaruk, M. Egholm, M. L. Landry, K. Dieckhaus, M. I. Rosen, and M. J. Kozal. Low-abundance HIV drug-resistant viral variants in treatment-experienced persons correlate with historical antiretroviral use. *PLoS One*, 4(6):e6079, 2009.

[94] H. Y. Lee, E. E. Giorgi, B. F. Keele, B. Gaschen, G. S. Athreya, J. F. Salazar-Gonzalez, K. T. Pham, P. A. Goepfert, J. Michael Kilby, M. S. Saag, et al. Modeling sequence evolution in acute HIV-1 infection. *Journal of Theoretical Biology*, 261(2):341–360, 2009.

[95] P. A. Levene. The structure of yeast nucleic acid. *Studies from the Rockefeller Institute for Medical Research: Reprints*, 36:183, 1921.

[96] H. Li and R. Durbin. Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics*, 25(14):1754–1760, 2009.

[97] H. Li, B. Handsaker, A. Wysoker, T. Fennell, J. Ruan, N. Homer, G. Marth, G. Abecasis, R. Durbin, et al. The sequence alignment/map format and SAMtools. *Bioinformatics*, 25(16):2078–2079, 2009.

[98] Y. Li, V. Mitaxov, and G. Waksman. Structure-based design of Taq DNA polymerases with improved properties of dideoxynucleotide incorporation. *Proceedings of the National Academy of Sciences*, 96(17):9491–9496, 1999.

[99] W. Liu and J. K. Jansson. *Environmental Molecular Microbiology*. Caister Academic Press, 2010.

[100] N. J. Loman and A. R. Quinlan. Poretools: a toolkit for analyzing nanopore sequence data. *Bioinformatics*, page btu555, 2014.

[101] R. Luo, B. Liu, Y. Xie, Z. Li, W. Huang, J. Yuan, G. He, Y. Chen, Q. Pan, Y. Liu, et al. SOAPdenovo2: an empirically improved memory-efficient short-read de novo assembler. *Gigascience*, 1(1):18, 2012.

[102] A. R. Macalalad, M. C. Zody, P. Charlebois, N. J. Lennon, R. M. Newman, C. M. Malboeuf, E. M. Ryan, C. L. Boutwell, K. A. Power, D. E. Brackney, et al. Highly sensitive and specific detection of rare variants in mixed viral populations from

massively parallel sequence data. *PLoS Computational Biology*, 8(3):e1002417, 2012.

[103] D. MacLean, J. D. G. Jones, and D. J. Studholme. Application of next-generation sequencing technologies to microbial genetics. *Nature Reviews Microbiology*, 7(4):287–296, 2009.

[104] M. Margulies, M. Egholm, W. E Altman, S. Attiya, J. S. Bader, L. A. Bemben, J. Berka, M. S. Braverman, Y. Chen, Z. Chen, et al. Genome sequencing in microfabricated high-density picolitre reactors. *Nature*, 437(7057):376–380, 2005.

[105] R. Marine, S. W. Polson, J. Ravel, G. Hatfull, D. Russell, M. Sullivan, F. Syed, M. Dumas, and K. E. Wommack. Evaluation of a transposase protocol for rapid generation of shotgun high-throughput sequencing libraries from nanogram quantities of DNA. *Applied and Environmental Microbiology*, 77(22):8071–8079, 2011.

[106] A. P. Masella, A. K. Bartram, J. M. Truszkowski, D. G. Brown, and J. D. Neufeld. PANDAseq: paired-end assembler for Illumina sequences. *BMC Bioinformatics*, 13(1):31, 2012.

[107] A. M. Maxam and W. Gilbert. A new method for sequencing DNA. *Proceedings of the National Academy of Sciences*, 74(2):560–564, 1977.

[108] L. F. McCaig, L. C. McDonald, S. Mandal, D. B. Jernigan, et al. *Staphylococcus aureus*-associated skin and soft tissue infections in ambulatory care. *Emerging Infectious Diseases*, 12(11):1715, 2006.

[109] K. E. McElroy, F. Luciani, and T. Thomas. GemSIM: general, error-model based simulator of next-generation sequencing data. *BMC Genomics*, 13(1):74, 2012.

[110] F. Meacham, D. Boffelli, J. Dhahbi, D. Martin, M. Singer, and L. Pachter. Identification and correction of systematic error in high-throughput sequence data. *BMC Bioinformatics*, 12(1):451, 2011.

[111] A. E. Minoche, J. C. Dohm, H. Himmelbauer, et al. Evaluation of genomic high-throughput sequencing data generated on Illumina HiSeq and Genome Analyzer systems. *Genome Biology*, 12(11):R112, 2011.

[112] M. Morey, A. Fernández-Marmiesse, D. Castiñeiras, J. M. Fraga, M. L. Couce, and J. A. Cocho. A glimpse into past, present, and future DNA sequencing. *Molecular Genetics and Metabolism*, 110(1):3–24, 2013.

[113] K. Nakamura, T. Oshima, T. Morimoto, S. Ikeda, H. Yoshikawa, Y. Shiwa, S. Ishikawa, M. C. Linak, A. Hirai, H. Takahashi, et al. Sequence-specific error profile of Illumina sequencers. *Nucleic Acids Research*, 39(13):e90–e90, 2011.

[114] T. Namiki, T. Hachiya, H. Tanaka, and Y. Sakakibara. MetaVelvet: an extension of Velvet assembler to de novo metagenome assembly from short sequence reads. *Nucleic Acids Research*, 40(20):e155–e155, 2012.

[115] S. I. Nikolenko, A. I. Korobeynikov, and M. A. Alekseyev. BayesHammer: Bayesian clustering for error correction in single-cell sequencing. *BMC Genomics*, 14(Suppl 1):S7, 2013.

[116] M. A. Nowak. What is a quasispecies? *Trends in Ecology & Evolution*, 7(4):118–121, 1992.

[117] V. O'Flaherty, G. Collins, and T. Mahony. The microbiology and biochemistry of anaerobic bioreactors with relevance to domestic sewage treatment. *Reviews in Environmental Science and Bio/Technology*, 5(1):39–55, 2006.

[118] O. Ojosnegros and N. Beerenwinkel. Models of RNA virus evolution and their roles in vaccine design. *Immunome Research*, 6, 2010.

[119] M. J. Pallen. Diagnostic metagenomics: potential applications to bacterial, viral and parasitic infections. *Parasitology*, pages 1–7, 2014.

[120] N. J. Parkinson, S. Maslau, B. Ferneyhough, G. Zhang, L. Gregory, D. Buck, J. Ragoussis, C. P. Ponting, and M. D. Fischer. Preparation of high-quality next-generation sequencing libraries from picogram quantities of target DNA. *Genome Research*, 22(1):125–133, 2012.

[121] K. R. Patil, L. Roune, and A. C. McHardy. The PhyloPythiaS web server for taxonomic assignment of metagenome sequences. *PloS One*, 7(6):e38581, 2012.

[122] A. Y. Pei, W. E. Oberdorf, C. W. Nossa, A. Agarwal, P. Chokshi, E. A. Gerz, Z. Jin, P. Lee, L. Yang, M. Poles, et al. Diversity of 16S rRNA genes within individual prokaryotic genomes. *Applied and Environmental Microbiology*, 2010.

[123] J. Peterson, S. Garges, M. Giovanni, P. McInnes, L. Wang, J. A. Schloss, V. Bonazzi, J. E. McEwen, K. A. Wetterstrand, C. Deal, et al. The NIH human microbiome project. *Genome Research*, 19(12):2317–2323, 2009.

[124] V. H. T. Pham and J. Kim. Cultivation of unculturable soil bacteria. *Trends in biotechnology*, 30(9):475–484, 2012.

[125] S. Prabhakaran, M. Rey, O. Zagordi, N. Beerenwinkel, and V. Roth. HIV-haplotype inference using a constraint-based Dirichlet process mixture model. *Machine Learning in Computational Biology (MLCB) NIPS Workshop*, pages pp. 1–4, 2010.

[126] L. Pray. Discovery of DNA structure and function: Watson and Crick. *Nature Education*, 1(1), 2008.

[127] M. C. F. Prosperi and M. Salemi. QuRe: software for viral quasispecies reconstruction from next-generation sequencing data. *Bioinformatics*, 28(1):132–133, 2012.

[128] C. Quince, T.P. Curtis, and W.T. Sloan. The rational exploration of microbial diversity. *The ISME journal*, 2(10):997–1006, 2008.

[129] C. Quince, A. Lanzén, T. P. Curtis, R. J. Davenport, N. Hall, I. M. Head, L. F. Read, and W. T. Sloan. Accurate determination of microbial diversity from 454 pyrosequencing data. *Nature Methods*, 6(9):639–641, 2009.

[130] C. Quince, A. Lanzen, R. J. Davenport, and P. J. Turnbaugh. Removing noise from pyrosequenced amplicons. *BMC Bioinformatics*, 12(1):38, 2011.

[131] D. A. Rasko, D. R. Webster, J. W. Sahl, A. Bashir, N. Boisen, F. Scheutz, E. E. Paxinos, R. Sebra, C. Chin, D. Iliopoulos, et al. Origins of the *E. coli* strain causing an outbreak of hemolytic–uremic syndrome in Germany. *New England Journal of Medicine*, 365(8):709–717, 2011.

[132] S. Reardon. Ebola treatments caught in limbo. *Nature*, 2014.

[133] W. S. Reznikoff. Tn5 as a model for understanding DNA transposition. *Molecular Microbiology*, 47(5):1199–1206, 2003.

[134] D. C. Richter, F. Ott, A. F. Auch, R. Schmid, and D. H. Huson. MetaSim - a sequencing simulator for genomics and metagenomics. *PLoS One*, 3(10):3373, 2008.

[135] S. M. Ross. *Probability models*, volume 2nd edition. Academic Press, 1980.

[136] N. Rusk. Cheap third-generation sequencing. *Nature Methods*, 6(4):244, 2009.

[137] F. Sanger and A. R. Coulson. A rapid method for determining sequences in DNA by primed synthesis with DNA polymerase. *Journal of Molecular Biology*, 94(3):441–448, 1975.

[138] F. Sanger, S. Nicklen, and A. R. Coulson. DNA sequencing with chain-terminating inhibitors. *Proceedings of the National Academy of Sciences*, 74(12):5463–5467, 1977.

[139] R. B. Sartor and S. K. Mazmanian. Intestinal microbes in inflammatory bowel diseases. *The American Journal of Gastroenterology Supplements*, 1(1):15–21, 2012.

[140] M. Schirmer, U. Z. Ijaz, L. D'Amore, N. Hall, W. T. Sloan, and C. Quince. Insight into biases and sequencing errors for amplicon sequencing with the Illumina MiSeq platform. *Nucleic Acids Research (In review)*, 2014.

[141] M. Schirmer, W. T. Sloan, and C. Quince. Benchmarking of viral haplotype reconstruction programmes: an overview of the capacities and limitations of currently available programmes. *Briefings in Bioinformatics*, page bbs081, 2012.

[142] S. C. Schuster. Next-generation sequencing transforms todays biology. *Nature*, 2008.

[143] C. L. Sears and W. S. Garrett. Microbes, microbiota, and colon cancer. *Cell Host & Microbe*, 15(3):317–328, 2014.

[144] N. Segata, L. Waldron, A. Ballarini, V. Narasimhan, O. Jousson, and C. Huttenhower. Metagenomic microbial community profiling using unique clade-specific marker genes. *Nature Methods*, 9(8):811–814, 2012.

[145] M. Shakya, C. Quince, J. H. Campbell, Z. K. Yang, C. W. Schadt, and M. Podar. Comparative metagenomic and rRNA microbial diversity characterization using archaeal and bacterial synthetic communities. *Environmental Microbiology*, 2013.

[146] N. Shimizu, T. Okamoto, E. N. Moriyama, Y. Takeuchi, T. Gojobori, and H. Hoshino. Patterns of nucleotide substitutions and implications for the immunological diversity of human immunodeficiency virus. *FEBS Letters*, 250(2):591–595, 1989.

[147] B. B. Simen, J. F. Simons, K. H. Hullsiek, R. M. Novak, R. D. MacArthur, J. D. Baxter, C. Huang, C. Lubeski, G. S. Turenchalk, M. S. Braverman, et al. Low-abundance drug-resistant viral variants in chronically HIV-infected, antiretroviral treatment-naive patients significantly impact treatment outcomes. *Journal of Infectious Diseases*, 199(5):693, 2009.

[148] D. D. Sommer, A. L. Delcher, S. L. Salzberg, and M. Pop. Minimus: a fast, lightweight genome assembler. *BMC Bioinformatics*, 8(1):64, 2007.

[149] F. Syed, H. Grunenwald, and N. Caruccio. Next-generation sequencing library preparation: simultaneous fragmentation and tagging using in vitro transposition. *Nature Methods*, 6(11), 2009.

[150] D. Thompson, P. Muriel, D. Russell, P. Osborne, A. Bromley, M. Rowland, S. Creigh-Tyte, C. Brown, et al. Economic costs of the foot and mouth disease outbreak in the United Kingdom in 2001. *Revue scientifique et technique-Office international des epizooties*, 21(3):675–685, 2002.

[151] J. F. Thompson and K. E. Steinmann. Single molecule sequencing with a HeliScope genetic analysis system. *Current Protocols in Molecular Biology*, pages 7–10, 2010.

[152] K. J. Travers, C. Chin, D. R. Rank, J. S. Eid, and S. W. Turner. A flexible and efficient template format for circular consensus sequencing and SNP detection. *Nucleic Acids Research*, 38(15):e159–e159, 2010.

[153] T. J. Treangen, S. Koren, I. Astrovskaya, D. Sommer, B. Liu, and M. Pop. MetAMOS: a metagenomic assembly and analysis pipeline for AMOS. *Genome Biology*, 12(1):1–27, 2011.

[154] S. M. Utturkar, D. M. Klingeman, M. L. Land, C. W. Schadt, M. J. Doktycz, D. A. Pelletier, and S. D. Brown. Evaluation and validation of de novo and hybrid assembly techniques to derive high-quality genome sequences. *Bioinformatics*, 30(19):2709–2716, 2014.

[155] M. Vignuzzi, J. K. Stone, J. J. Arnold, C. E. Cameron, and R. Andino. Quasispecies diversity determines pathogenesis through cooperative interactions in a viral population. *Nature*, 439(7074):344–348, 2005.

[156] M. Vos. A species concept for bacteria based on adaptive divergence. *Trends in Microbiology*, 2010.

[157] X. V. Wang, N. Blades, J. Ding, R. Sultana, and G. Parmigiani. Estimation of sequencing error rates in short reads. *BMC Bioinformatics*, 13(1):185, 2012.

[158] J. D. Watson, F. H. C. Crick, et al. Molecular structure of nucleic acids. *Nature*, 171(4356):737–738, 1953.

[159] P. M. Woollard, N. A. L. Mehta, J. J. Vamathevan, S. Van Horn, B. K. Bonde, and D. J. Dow. The application of next-generation sequencing technologies to drug discovery and development. *Drug Discovery Today*, 16(11):512–519, 2011.

[160] C. F. Wright, M. J. Morelli, G. Thebaud, N. J. Knowles, P. Herzyk, D. J. Paton, D. T. Haydon, and D. P. King. Beyond the consensus: Dissecting within-host viral population diversity of foot-and-mouth disease virus by using next-generation genome sequencing. *Journal of Virology*, 85(5):2266, 2011.

[161] Z. Yang, N. Goldman, and A. Friday. Comparison of models for nucleotide substitution used in maximum-likelihood phylogenetic estimation. *Molecular Biology and Evolution*, 11(2):316, 1994.

[162] O. Zagordi, A. Bhattacharya, N. Eriksson, and N. Beerenwinkel. ShoRAH: estimating the genetic diversity of a mixed sample from next-generation sequencing data. *BMC Bioinformatics*, 12(1):119, 2011.

[163] O. Zagordi, A. Töpfer, S. Prabhakaran, V. Roth, E. Halperin, and N. Beerenwinkel. Probabilistic inference of viral quasispecies subject to recombination. In *Research in Computational Molecular Biology*, pages 342–354. Springer, 2012.

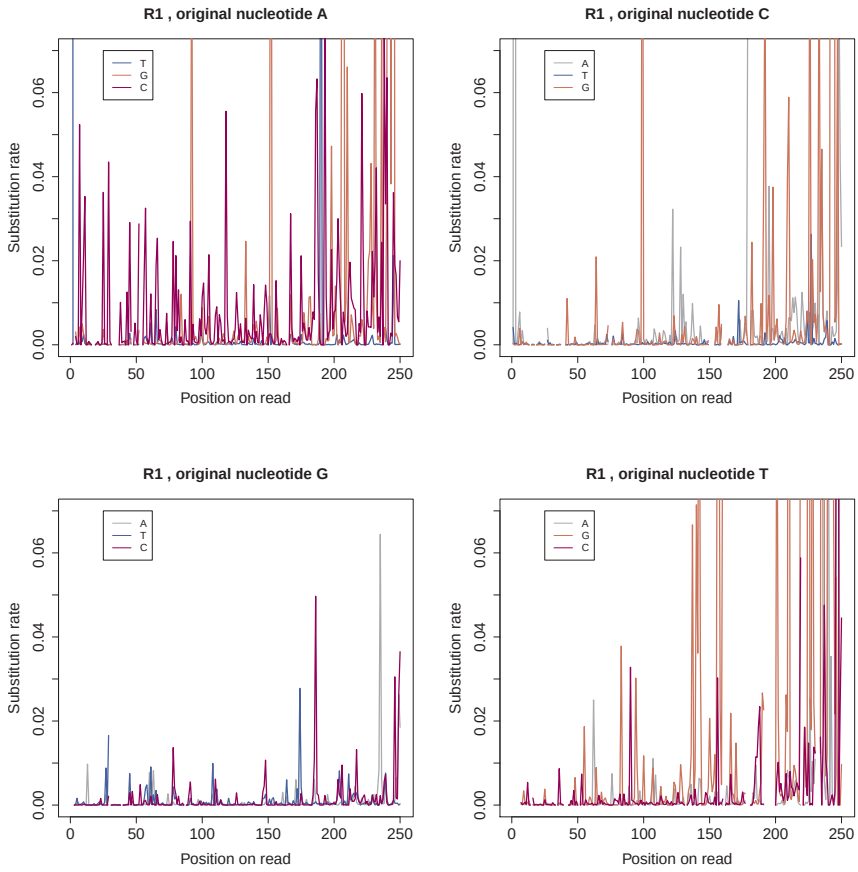
[164] D. R. Zerbino and E. Birney. Velvet: algorithms for de novo short read assembly using de Bruijn graphs. *Genome Research*, 18(5):821–829, 2008.

[165] J. Zhang, K. Kobert, T. Flouri, and A. Stamatakis. PEAR: a fast and accurate Illumina Paired-End reAd mergeR. *Bioinformatics*, page btt593, 2013.

A Appendix for Chapter 6

Error profiles for data set *DS81*

R1 Substitutions



R1 Insertions and Deletions

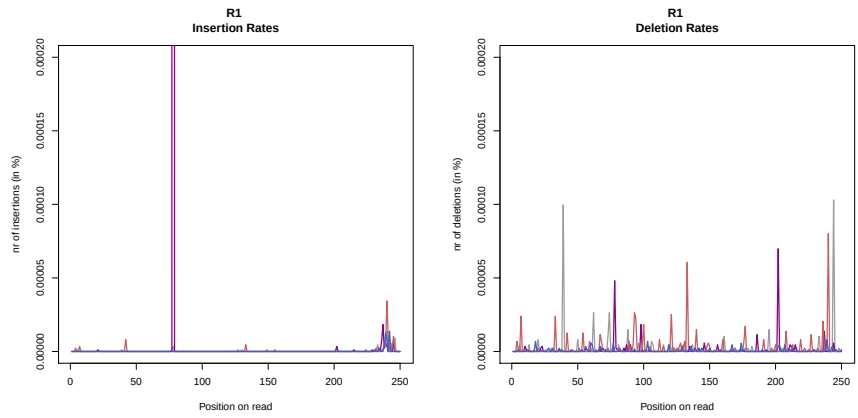
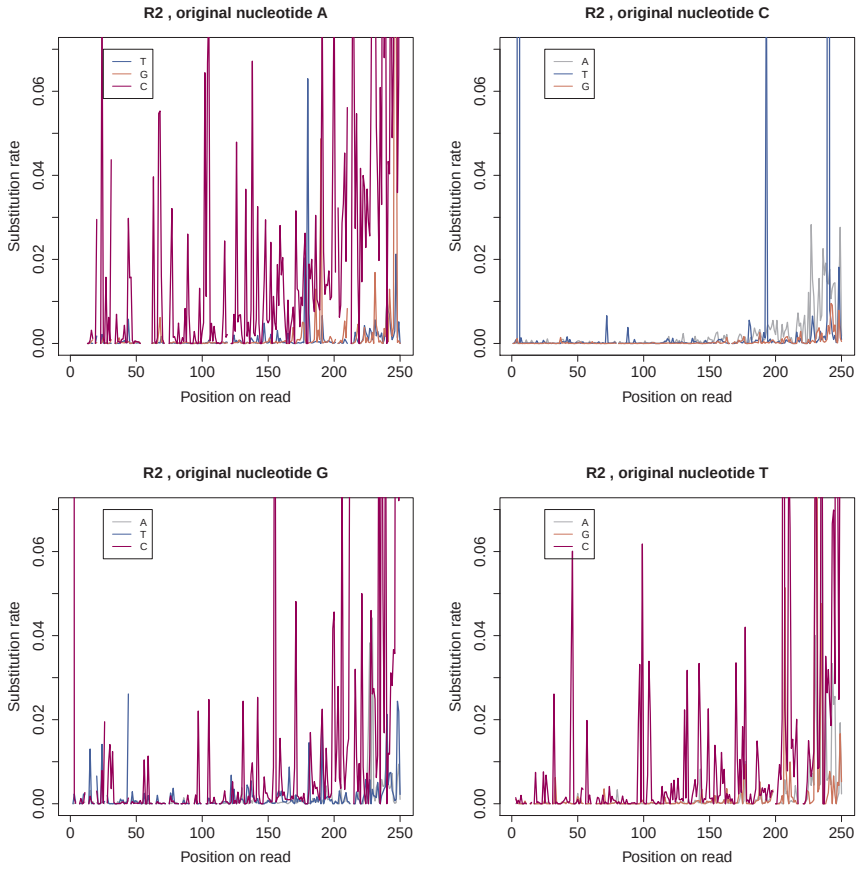


Figure 1.1: Position and nucleotide specific error profiles for the R1 reads of data set *DS81*. The V4 region of the balanced mock community was sequenced on the MiSeq and the library was prepared with the Fusion Golay method.

R2 Substitutions



R2 Insertions and Deletions

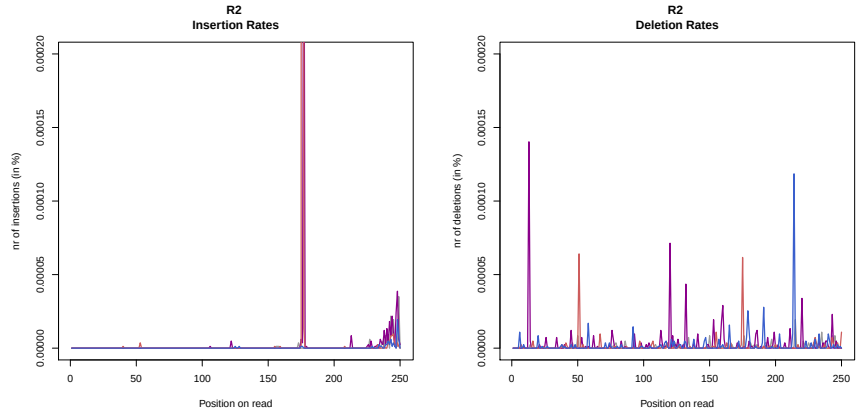
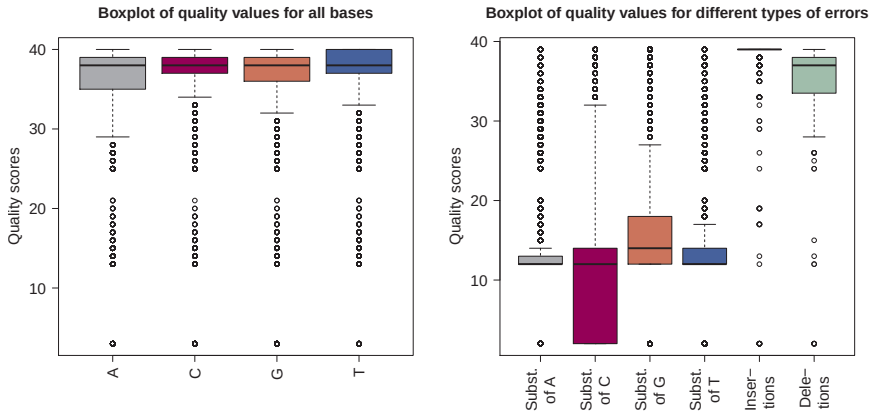


Figure 1.2: Position and nucleotide specific error profiles for the R2 reads of data set *DS81*. The V4 region of the balanced mock community was sequenced on the MiSeq and the library was prepared with the Fusion Golay method.

R1 Quality Profiles



R2 Quality Profiles

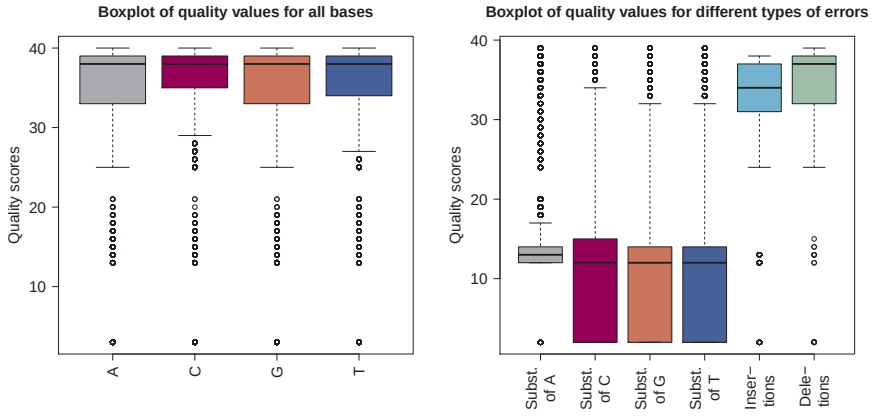
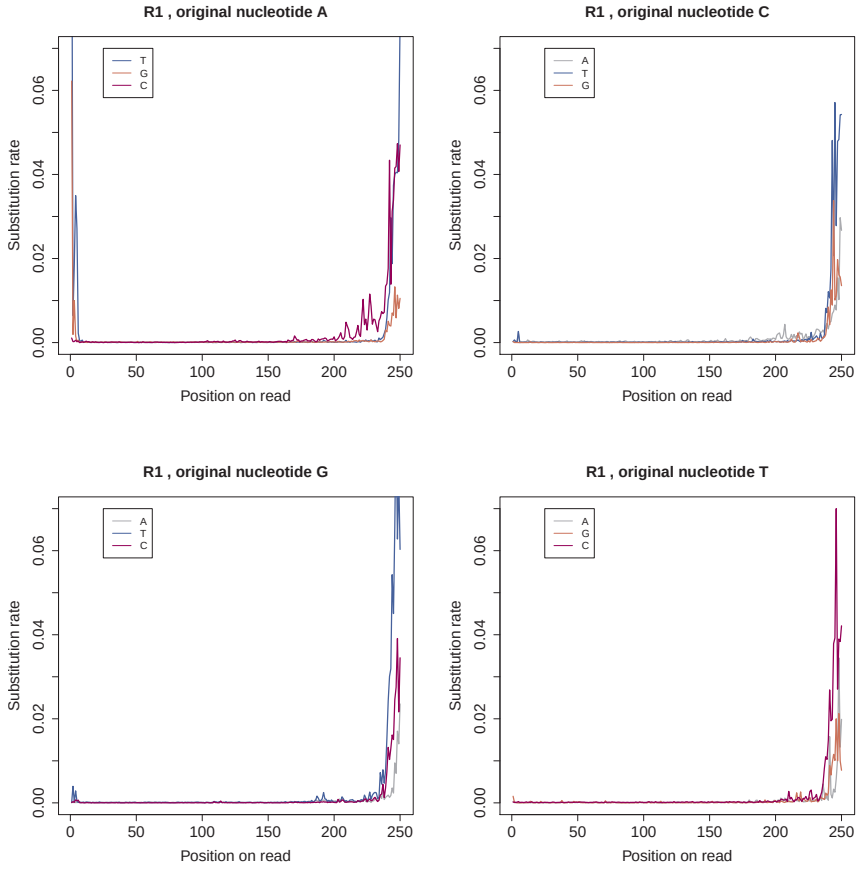


Figure 1.3: Quality profiles for R1and R2 reads of data set *DS81*. The V4 region of the balanced mock community was sequenced on the MiSeq and the library was prepared with the Fusion Golay method.

Error profiles for data set *DS34*

R1 Substitutions



R1 Insertions and Deletions

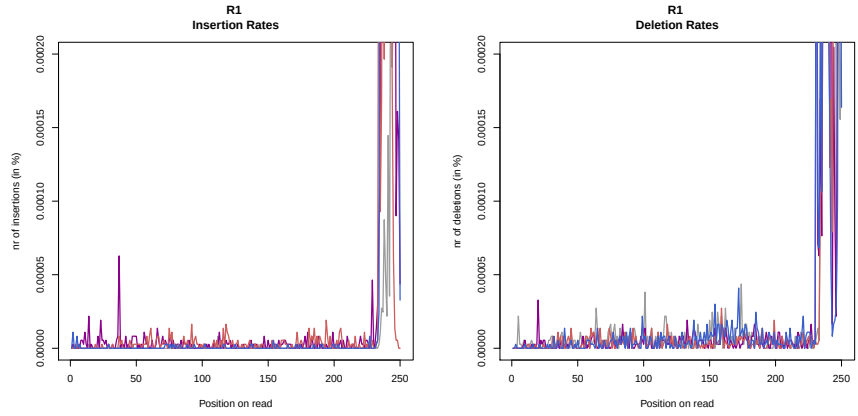
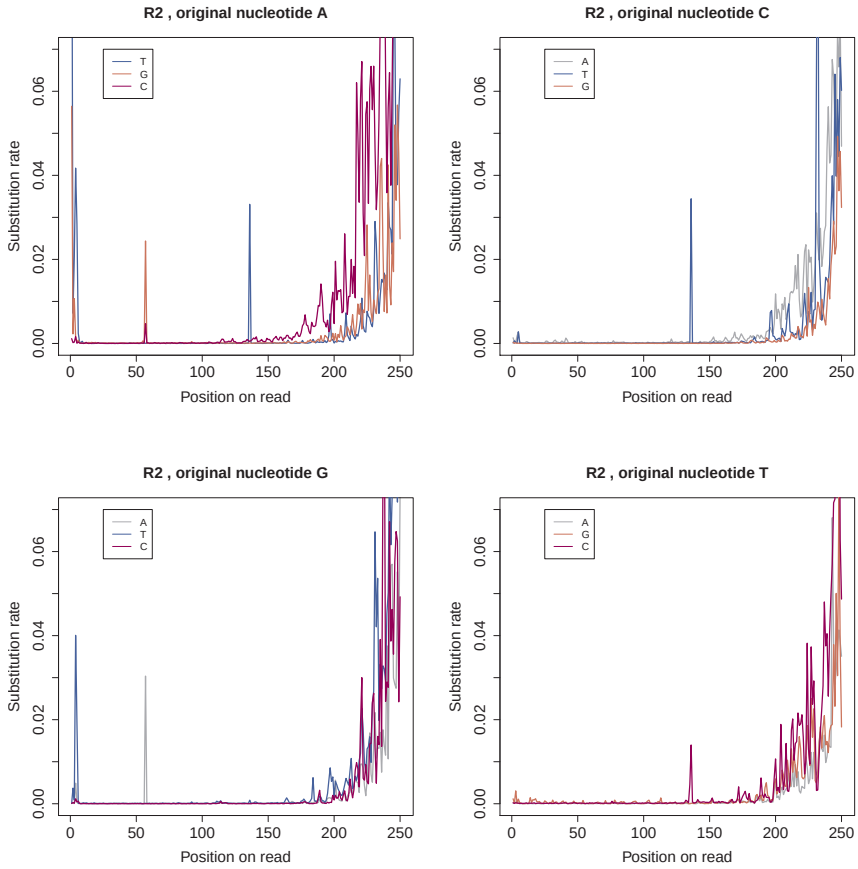


Figure 1.4: Position and nucleotide specific error profiles for the R1 reads of data set *DS34*. The V3/V4 region of *Caldicellulosiruptor saccharolyticus DSM 8903* was sequenced on the MiSeq and the library was prepared with the NexteraXT kit.

R2 Substitutions



R2 Insertions and Deletions

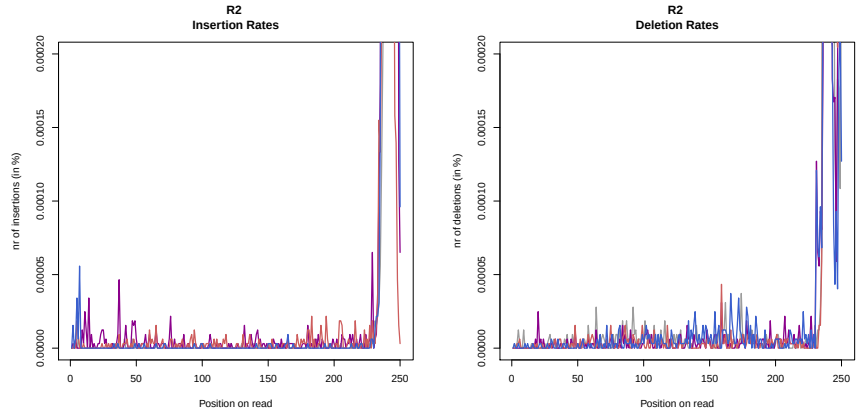
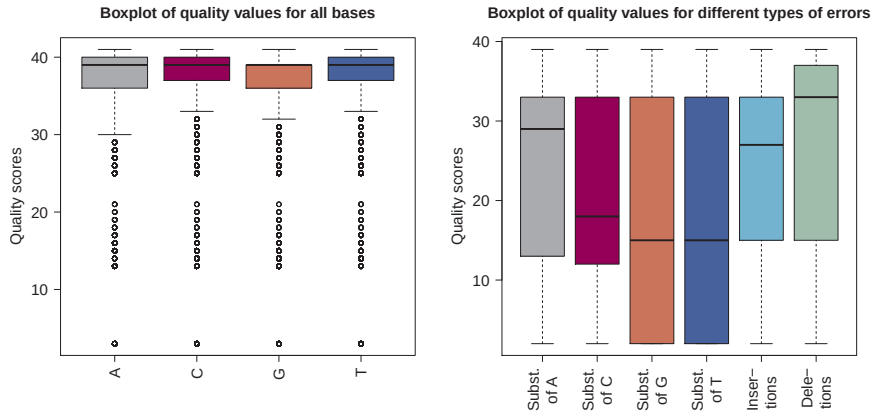


Figure 1.5: Position and nucleotide specific error profiles for the R2 reads of data set *DS34*. The V3/V4 region of *Caldicellulosiruptor saccharolyticus DSM 8903* was sequenced on the MiSeq and the library was prepared with the NexteraXT kit.

R1 Quality Profiles



R2 Quality Profiles

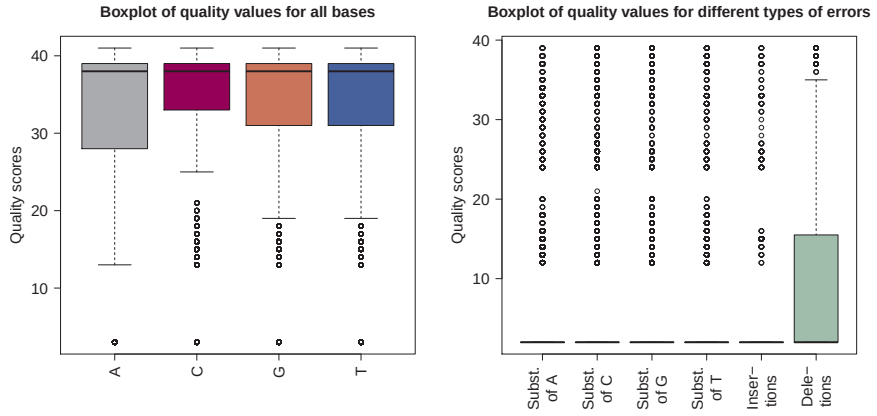
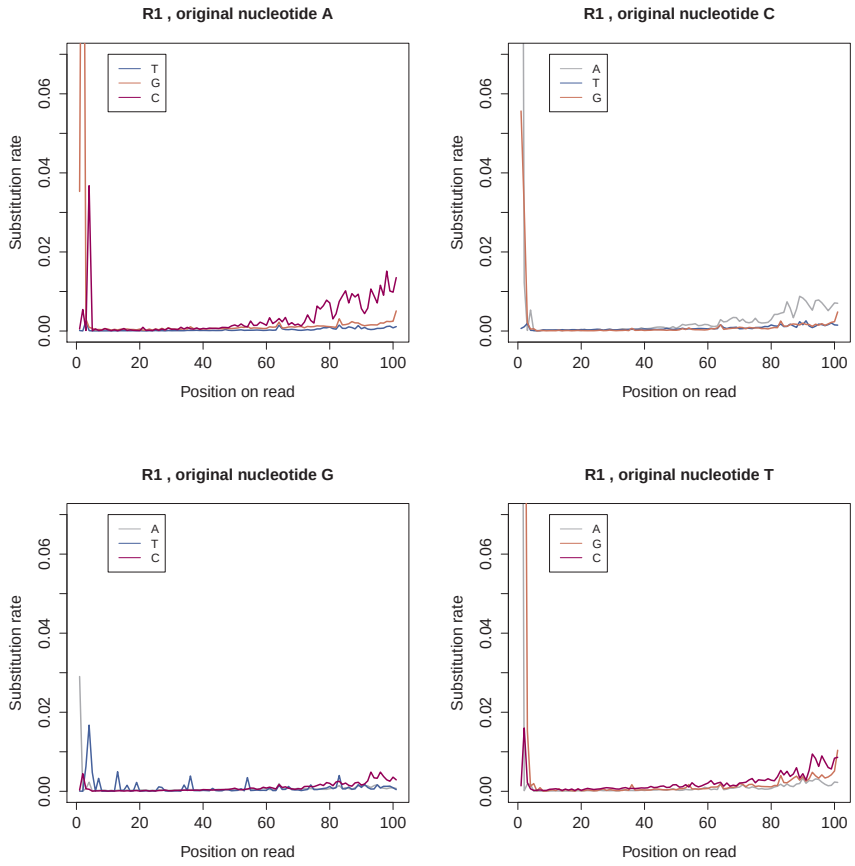


Figure 1.6: Quality profiles for R1and R2 reads of data set *DS34*. The V3/V4 region of *Caldicellulosiruptor saccharolyticus DSM 8903* was sequenced on the MiSeq and the library was prepared with the NexteraXT kit.

B Appendix for Chapter 7

Error profiles for data set *DS5*

R1 Substitutions



R1 Insertions and Deletions

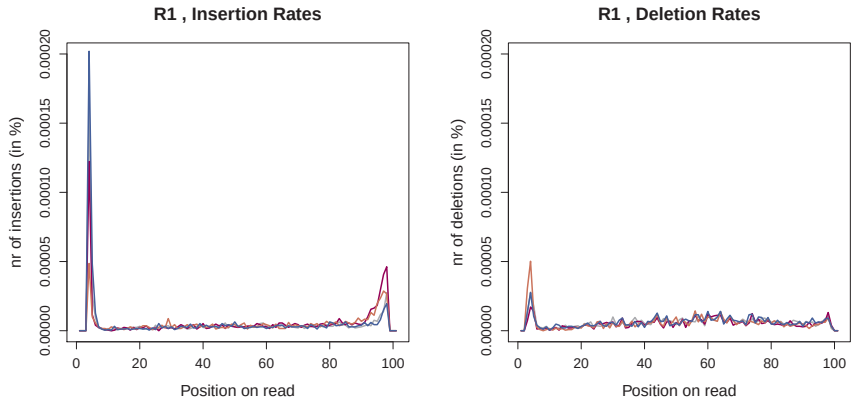
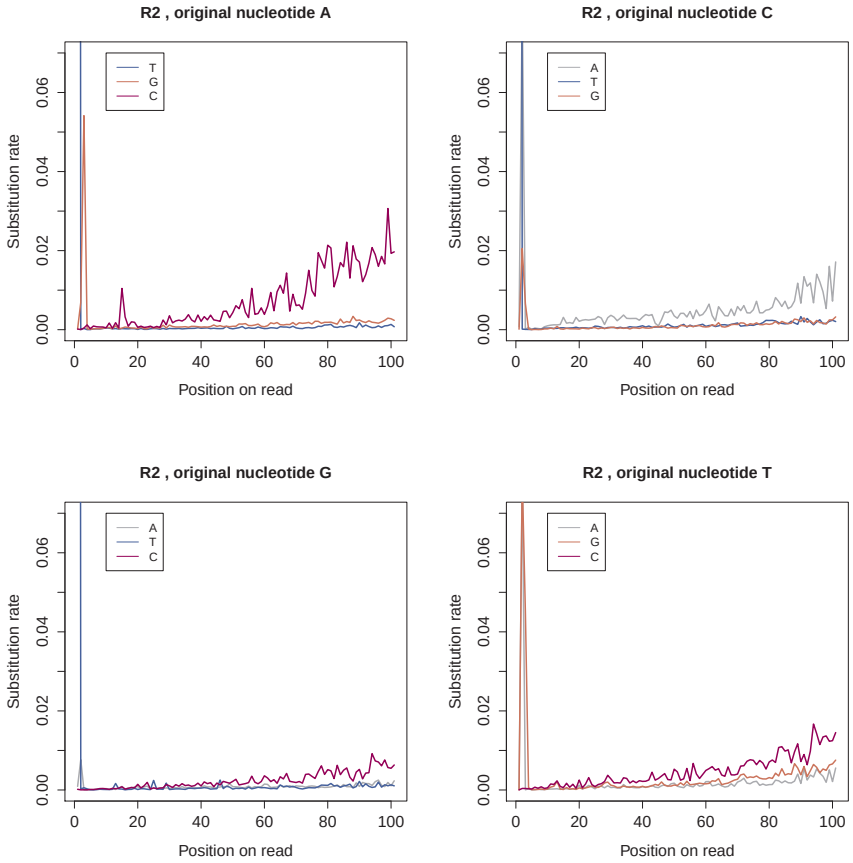


Figure 2.7: Position and nucleotide specific error profiles for the R1 reads of data set *DS5*. A sample from the balanced mock community was sequenced on the GAII and the library was prepared with the Parkinson method using 0.5ng of input DNA.

R2 Substitutions



R2 Insertions and Deletions

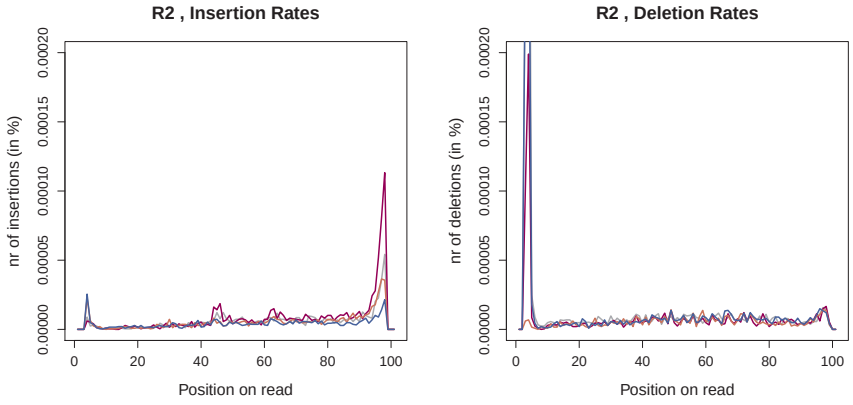
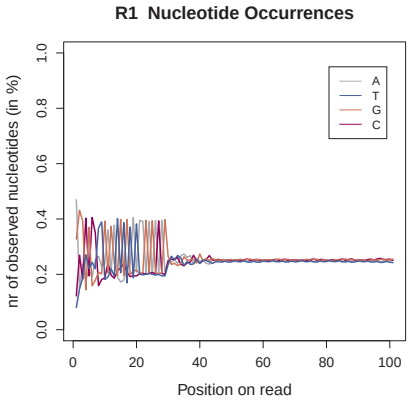


Figure 2.8: Position and nucleotide specific error profiles for the R2 reads of data set *DS5*. A sample from the balanced mock community was sequenced on the GAII and the library was prepared with the Parkinson method using 0.5ng of input DNA.

R1 Nucleotide Distribution



R2 Nucleotide Distribution

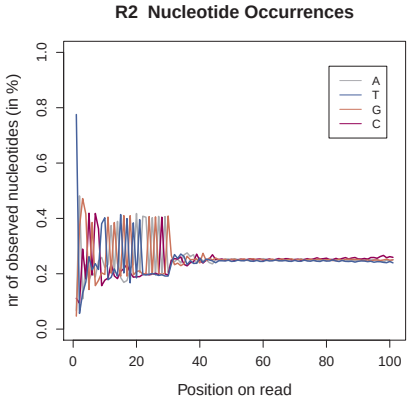
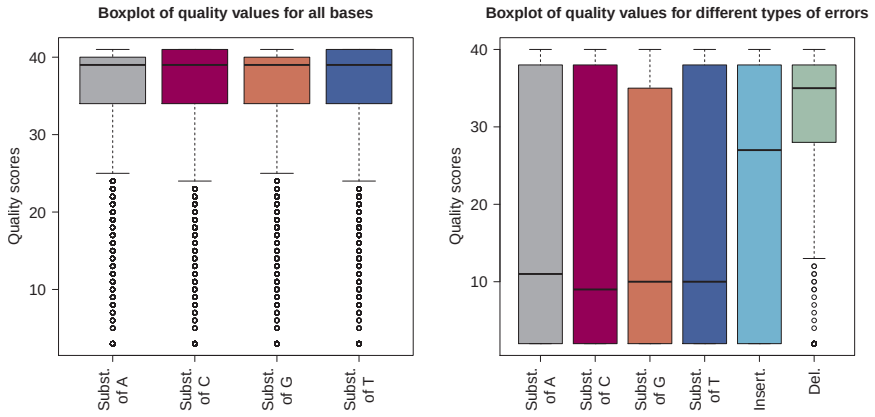


Figure 2.9: The figure displays the nucleotide distribution for the R1and R2 reads of data set *DS5*. An uneven distribution was observed at the start and end of the read as well as a bias towards C and G.

R1 Quality Profiles



R2 Quality Profiles

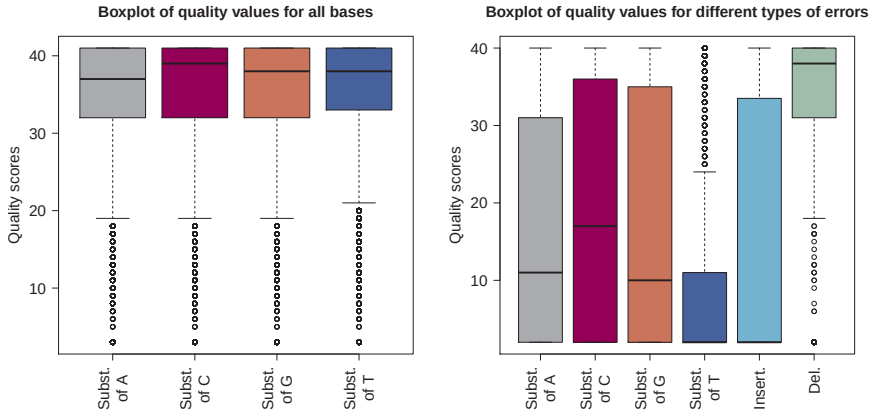
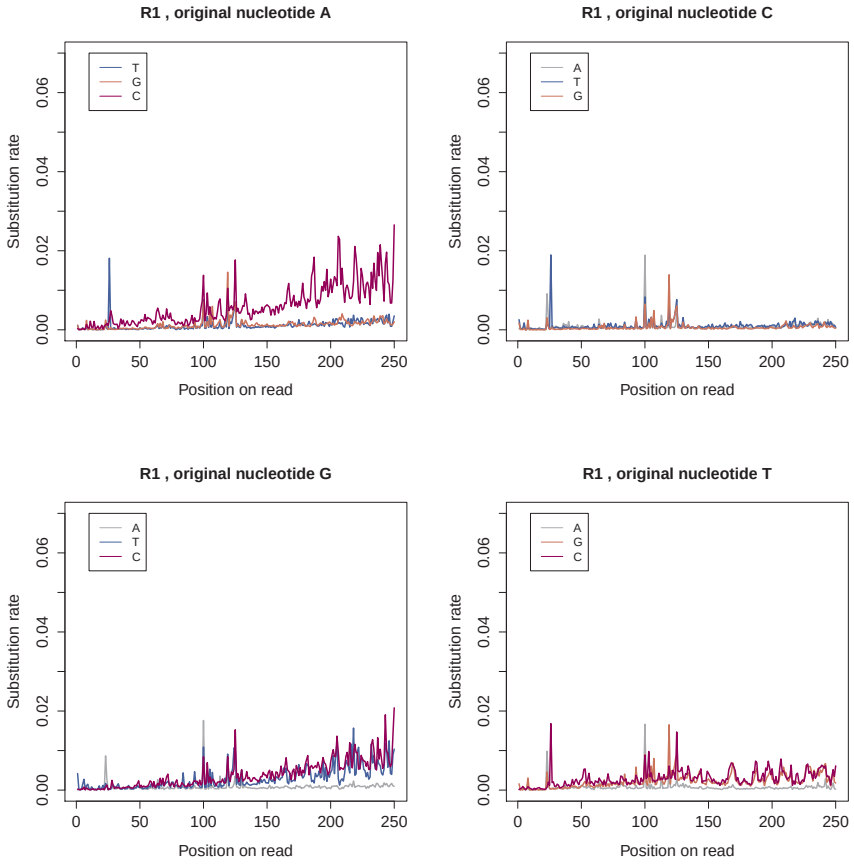


Figure 2.10: Quality profiles for R1and R2 reads of data set *DS5*. A sample from the balanced mock community was sequenced on the GAII and the library was prepared with the Parkinson method using 0.5ng of input DNA.

Error profiles for data set *DS76*

R1 Substitutions



R1 Insertions and Deletions

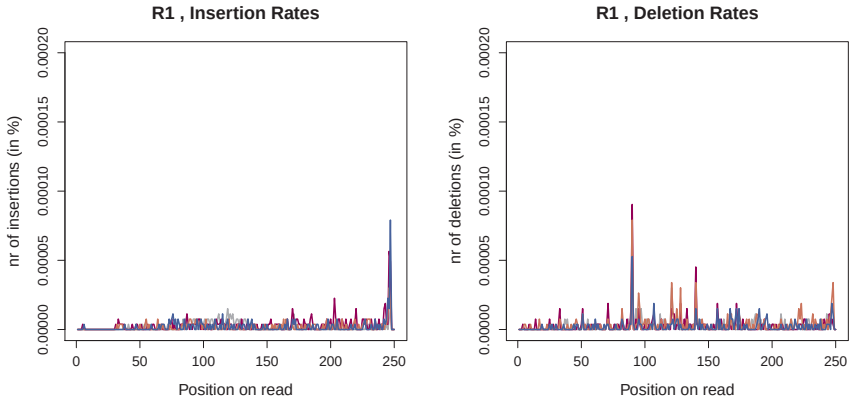
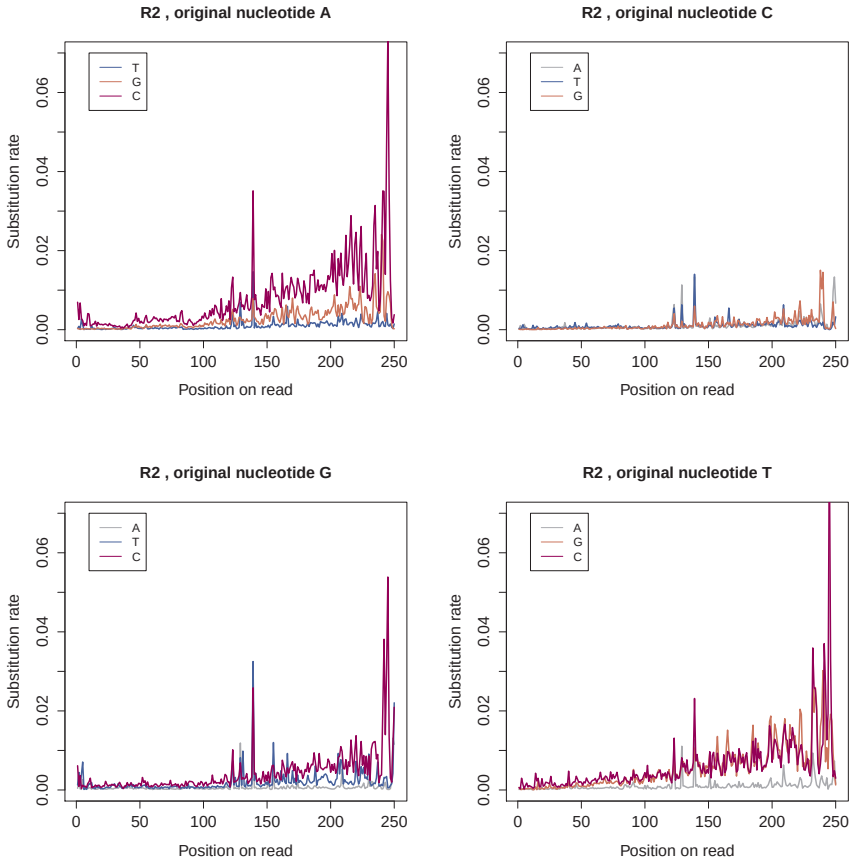


Figure 2.11: Position and nucleotide specific error profiles for the R1 reads of data set *DS76*. *Burkholderia xenovorans* was sequenced on the MiSeq and the library was prepared with the NexteraXT kit with 1ng input DNA.

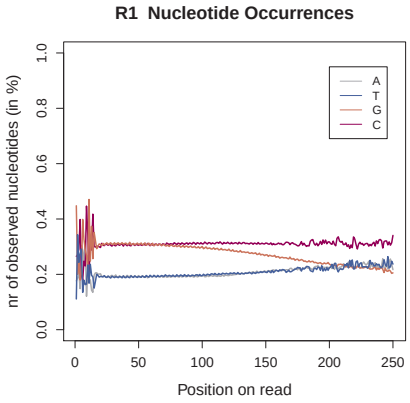
R2 Substitutions



R2 Insertions and Deletions

Figure 2.12: Position and nucleotide specific error profiles for the R2 reads of data set *DS76*. *Burkholderia xenovorans* was sequenced on the MiSeq and the library was prepared with the NexteraXT kit with 1ng input DNA.

R1 Nucleotide Distribution



R2 Nucleotide Distribution

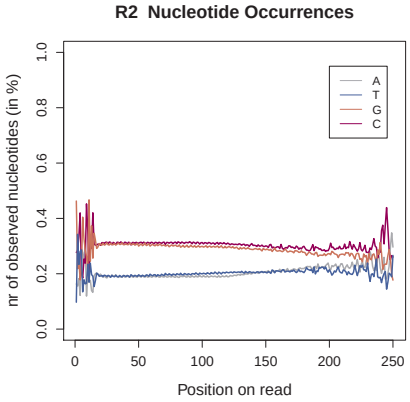
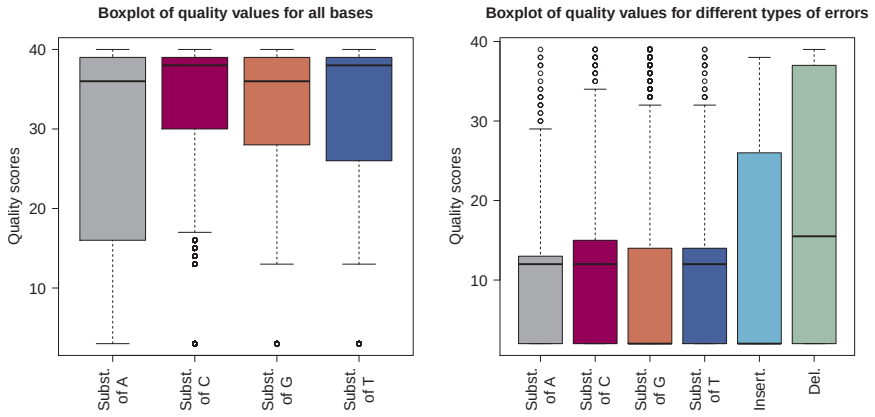


Figure 2.13: The figure displays the nucleotide distribution for the R1and R2 reads of data set *DS76*. The uneven distribution at the start of the read is characteristic for the Parkinson libraries.

R1 Quality Profiles



R2 Quality Profiles

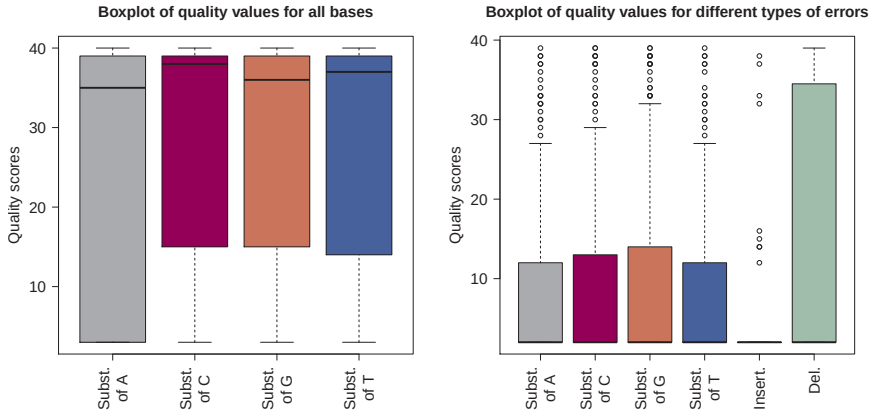


Figure 2.14: Quality profiles for R1and R2 reads of data set *DS76*. *Burkholderia xenovorans* was sequenced on the MiSeq and the library was prepared with the NexteraXT kit with 1ng input DNA.