

Development of a data processing toolkit for the analysis of next-generation sequencing data generated using the primer ID approach

Jan Phillipus Lourens Labuschagne

A thesis submitted in partial fulfilment of the requirements for the degree of
Doctor Philosophiae in the South African National Bioinformatics Institute,
University of the Western Cape.



Supervisors: Prof. Simon A. Travers

Dr. Paul T. Edlefsen

Dr. Colin Anthony

Dr. Imogen A. Wright

August 2018

Keywords

Next generation sequencing

Primer ID

PCR recombination

HIV infection timing

Amplicon Sequencing

Droplet PCR

Hypermutation

DNA Sequence Entropy

Sequencing accuracy

PCR efficiency

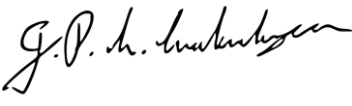


UNIVERSITY *of the*
WESTERN CAPE

Declaration

I declare that **Development of a data processing toolkit for the analysis of next-generation sequencing data generated using the primer ID approach** is my own work, that it has not been submitted for any degree or examination in any other university, and all the sources I have used or quoted have been indicated and acknowledged by complete references.

Full name: Jan Phillipus Lourens Labuschagne Date: 30 August 2018

Signed: 



Acknowledgements

I would like to thank all my supervisors for all the hours they spent explaining the various topics I was not familiar with. This project required knowledge from various fields and I would not have been able to complete it without their expert guidance.

A grant from the Gates Foundation provided a salary to me for doing work that overlapped substantially with the requirements of this thesis. The existence of this position was critical as it enabled me to dedicate enough time to this work to ensure the completion of the thesis.

Roux-cil Ferreira has been my companion throughout this entire process, bearing the brunt of the frustration induced bad moods while remaining supportive and loving. Thank you.



Abstract

Sequencing an HIV quasispecies with next generation sequencing technologies yields a dataset with significant amplification bias and errors resulting from both the PCR and sequencing steps. Both the amplification bias and sequencing error can be reduced by labelling each cDNA (generated during the reverse transcription of the viral RNA to DNA prior to PCR) with a random sequence tag called a Primer ID (PID). Processing PID data requires additional computational steps, presenting a barrier to the uptake of this method. *MotifBinner* is an R package designed to handle PID data with a focus on resolving potential problems in the dataset.

MotifBinner groups sequences into bins by their PID tags, identifies and removes false unique bins, produced from sequencing errors in the PID tags, as well as removing outlier sequences from within a bin. *MotifBinner* produces a consensus sequence for each bin, as well as a detailed report for the dataset, detailing the number of sequences per bin, the number of outlying sequences per bin, rates of chimerism, the number of degenerate letters in the final consensus sequences and the most divergent consensus sequences (potential contaminants).

We characterized the ability of the PID approach to reduce the effect of sequencing error, to detect minority variants in viral quasispecies and to reduce the rates of PCR induced recombination. We produced reference samples with known variants at known frequencies to study the effectiveness of increasing PCR elongation time, decreasing the number of PCR cycles, and sample partitioning, by means of dPCR (droplet PCR), on PCR induced recombination. After sequencing these artificial samples with the PID approach, each consensus sequence was compared to the known variants. There are complex relationships between the sample preparation protocol and the characteristics of the resulting dataset. We produce a set of recommendations that can be used to inform sample preparation that is the most useful the particular study.

The AMP trial infuses HIV-negative patients with the VRC01 antibody and monitors for HIV infections. Accurately timing the infection event and reconstructing the founder viruses of these infections are critical for relating infection risk to antibody titer and homology between the founder virus and antibody binding sites. [Dr. Paul Edlefsen](#) at the Fred Hutch Cancer Research Institute developed a pipeline that performs infection timing and founder reconstruction. Here, we document a portion of the pipeline, produce detailed tests for that portion of the pipeline and investigate the robustness of some of the tools used in the pipeline to violations of their assumptions.

1 Table of Contents

Abstract.....	5
1 Introduction	10
1.1 Thesis Outline.....	10
1.2 Central Dogma of Molecular Biology	10
1.3 Basic Overview of HIV	13
1.3.1 Overview of HIV treatment.....	16
1.3.2 HIV drug resistance	18
1.3.3 HIV vaccine.....	19
1.4 Sequencing.....	21
1.4.1 The polymerase chain reaction.....	21
1.4.2 Sanger Sequencing.....	25
1.4.3 Single Genome Amplification and Sequencing	27
1.4.4 High throughput sequencing	27
1.5 The PID approach.....	32
1.6 Complications of Primer ID	36
1.6.1 Skewed template resampling.....	36
1.6.2 PID Collisions	37
1.6.3 Sequencing error with in the PID region.....	39
1.6.4 Sequencing and PCR errors in the final consensus sequences	40
1.7 Other PID processing toolkits	42
1.7.1 MT-Toolbox.....	43
1.7.2 Ruby scripts of (S. Zhou et al., 2015).....	44
2 MotifBinner.....	45
2.1 Design and Implementation.....	45
2.1.1 Locate and identify the PIDs in each sequence	45
2.1.2 Group sequences into bins based on their PIDs	47
2.1.3 Discard bins based on invalid PIDs.....	47
2.1.4 Determine which bins are chimeric	50
2.1.5 Find and remove sequences that were given an incorrect PID	51
2.1.6 Align the sequences in each bin.....	52
2.1.7 Construct a consensus sequence for each bin.....	52
2.1.8 Produce a report detailing the steps that were taken.....	52
2.1.9 Save all the results and the report.....	53
2.2 Materials and Methods.....	53

2.3	Results and Discussion	58
2.3.1	Validation of MotifBinner's design.	58
2.3.2	Evaluating the benefit of using the PID approach	79
2.3.3	Conclusions and future work	92
3	Protocol Optimization for PID sample preparation	95
3.1	Methods.....	95
3.1.1	Processing Raw Sequences	97
3.1.2	Generating Consensus Sequences	98
3.1.3	Constructing an array of accuracy and quality score tallies	98
3.1.4	Detecting Recombination	99
3.2	Results and Discussion	100
3.2.1	Protocols that contained dPCR amplified less input templates.....	102
3.2.2	Biased amplification occurred in all protocols.....	104
3.2.3	The Primer ID approach reduces the effect of sequencing errors.....	105
3.2.4	Prevalence estimates based on PID data are less variable.	111
3.2.5	PCR recombination is influenced by sequence homology, but effectively removed by the PID approach.	119
3.2.6	The loss of minority variants in consensus sequences at the 5' end was minimal.....	128
3.3	Conclusion and Future Work	133
4	Infection Timing	135
4.1	Overview and Design	135
4.2	hypermurR	138
4.2.1	Algorithm	138
4.2.2	Implementation	139
4.2.3	Tests	141
4.2.4	Benchmarks and Comparisons.....	142
4.2.5	Future Work and Conclusions.....	146
4.3	Entropy Calculation.....	147
4.3.1	Definition and Explanation of Shannon Entropy.....	147
4.3.2	Implementation	149
4.3.3	Tests and Examples.....	151
4.3.4	Future Work and Conclusions.....	155
4.4	PhyML	156
4.4.1	Overview of PhyML.....	156
4.4.2	Implementation Details	156
4.4.3	Test Procedure	157

4.4.4	Data simulation	159
4.4.5	Test results and discussion	167
4.4.6	Future work and Conclusions.....	185
4.5	Poisson Fitter.....	187
4.5.1	Implementation Details	187
4.5.2	Tests and Examples	192
4.5.3	Test results and discussion	192
4.5.4	Future work and Conclusions (Poisson Fitter)	204
4.6	Future work and Conclusions (Infection Timing).....	205
5	Future work and conclusions	207
6	Bibliography	212
7	Appendices.....	229
7.1	Binning Report (for the 006wpi dataset)	229
7.1.1	Input Sequences and Motifs Found	229
7.1.2	Sequence Lengths	230
7.1.3	Bin Sizes.....	231
7.1.4	Consensus Cutoff	232
7.1.5	Chimeric bins.....	232
7.1.6	Outlier Removal	232
7.1.7	Degeneracies in Consensus Sequences.....	234
7.1.8	Relatedness of Final Consensuses.....	235
7.1.9	Running Time	237
7.1.10	Parameters.....	238
7.1.11	MotifBinner version	239
7.2	Binning Report (for the 193wpi dataset)	240
7.2.1	Input Sequences and Motifs Found	240
7.2.2	Sequence Lengths	241
7.2.3	Bin Sizes.....	242
7.2.4	Consensus Cutoff	243
7.2.5	Chimeric bins.....	243
7.2.6	Outlier Removal	243
7.2.7	Degeneracies in Consensus Sequences.....	244
7.2.8	Relatedness of Final Consensuses.....	246
7.2.9	Running Time	248
7.2.10	Parameters.....	249
7.2.11	MotifBinner version	250

7.3	Detailed methods for section 3.2.....	251
7.4	hypermurR installation instructions for Ubuntu.....	262
7.5	hypermurR usage instructions	262
7.6	Patching PhyML.....	264



2 Introduction

2.1 Thesis Outline

The main goal of this project was to develop methods that will improve the accuracy with which the genetic information of populations of organisms can be obtained. Sequencing genetic information is an error-prone process. Robust techniques exist for correcting these errors when sequencing an individual organism, but the approaches for populations of organisms are more complex.

The first chapter provides background information and reviews some of the existing approaches for obtaining the genetic information from populations of organisms. A very basic introduction to molecular biology is provided. Using these concepts, the human immunodeficiency virus (HIV) is described since it was used as a case study in this work. Next, the traditional approach and its limitations to sequencing HIV is reviewed motivating the application of next generation sequencing (NGS) for sequencing HIV. Modifications to the standard NGS protocols that enable the sequencing of the HIV quasispecies are described in the subsections about the PID approach. The first chapter concludes with a discussion of an approach for computing the time since HIV infection based on sequence data.

MotifBinner implements a complex algorithm to process the sequence data. Each step in this algorithm, and the reasoning that led to the inclusion of that step, together with a detailed example of the use of MotifBinner is detailed in Chapter 2. An experiment comparing variations in the protocols for preparing a sequence library with the PID approach is detailed in Chapter 3. The final chapter presents work that tests and investigates a software pipeline that estimates the time since infection based on datasets like those produced by MotifBinner.

2.2 Central Dogma of Molecular Biology

Molecular biology is a large and complex subject. This section briefly summarizes a few concepts drawn from (Fairbanks & Andersen, 1999). In simple terms, the central dogma of molecular biology states that genetic information is used to construct proteins. Proteins are large molecules that are essential to all living organisms (F. H. Crick, 1958).

Genetic information is stored in long linear molecules called deoxyribonucleic acid (DNA). DNA consists of a long series of nucleotides (also referred to as bases or residues) that are linked together. Four different kinds of nucleotides can be found in DNA: adenine (A), cytosine (C), guanine (G) and thymine (T) (Watson & Crick, 1953). These four nucleotides form a four-letter alphabet and the order in which they occur encodes the genetic information. When DNA is sequenced, the order of the nucleotides is determined.

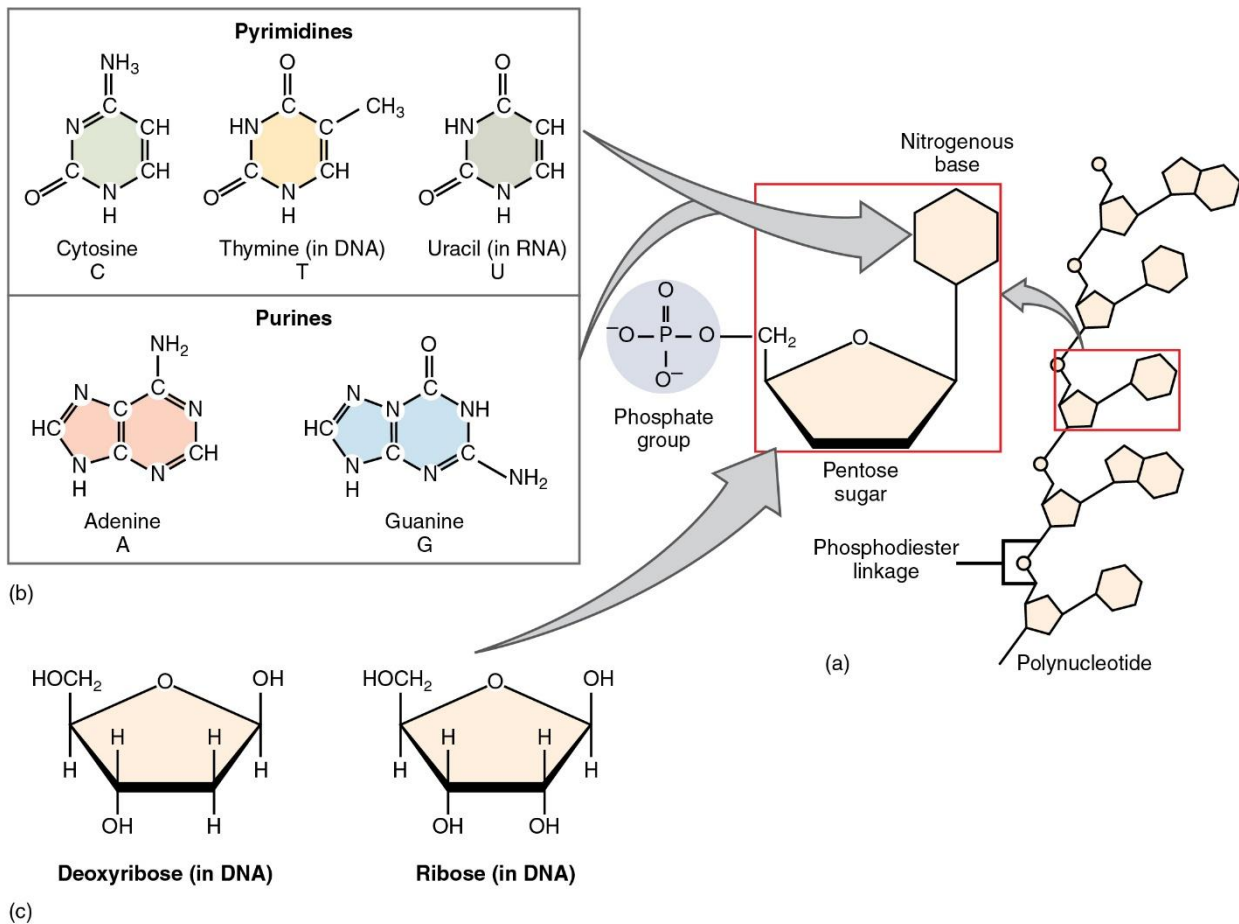


Figure 1: Structure of nucleotides. (a) The building blocks of all nucleotides are one or more phosphate groups, a pentose sugar, and a nitrogen-containing base. (b) The nitrogen-containing bases of nucleotides. (c) The two pentose sugars of DNA and RNA. (OpenStax-College, 2015); Download for free at <http://cnx.org/contents/14fb4ad7-39a1-4eee-ab6e-3ef2482e3e22@7.30>.

A nucleotide is composed of a five-carbon sugar molecule with a phosphate group at one end (the 5' carbon) and a nitrogenous base at the other (the 1' carbon), see Figure 1. The phosphate group of one nucleotide can bind to the 3' carbon of another nucleotide, allowing long chains to form. This also provides the orientation of the DNA, with DNA being read from the 5' to the 3' end. When a gene or region is towards the 5' end of the DNA molecule, it is said to be upstream and downstream when it is towards the 3' end.

DNA exists as double stranded helices in which the nitrogenous bases facing each other (Figure 2). The facing nitrogenous bases bind to each other with the following restrictions: 1) Adenine can only bind to thymine and; 2) Cytosine can only bind to guanine (Watson & Crick, 1953). Adenine binds to thymine with two hydrogen bonds while guanine binds to cytosine with three hydrogen bonds, thus the bonds between guanine and cytosine is stronger than the bonds between adenine and thymine. Additionally, for the bases to bind to each other, the two strands must run in opposite directions in the helix. These restrictions mean that any strand in a double stranded DNA molecule is uniquely

determined by the other strand. The two strands are said to be reverse complements of each other. The formal terminology for the orientation of the two strands is to refer to them as either sense or antisense based on their relationship to the messenger RNA for which they encode. However, in the Primer ID literature, the strands are referred to as either the forward or reverse strand, with the forward strand being the strand that is the same as the RNA version of the sequence.

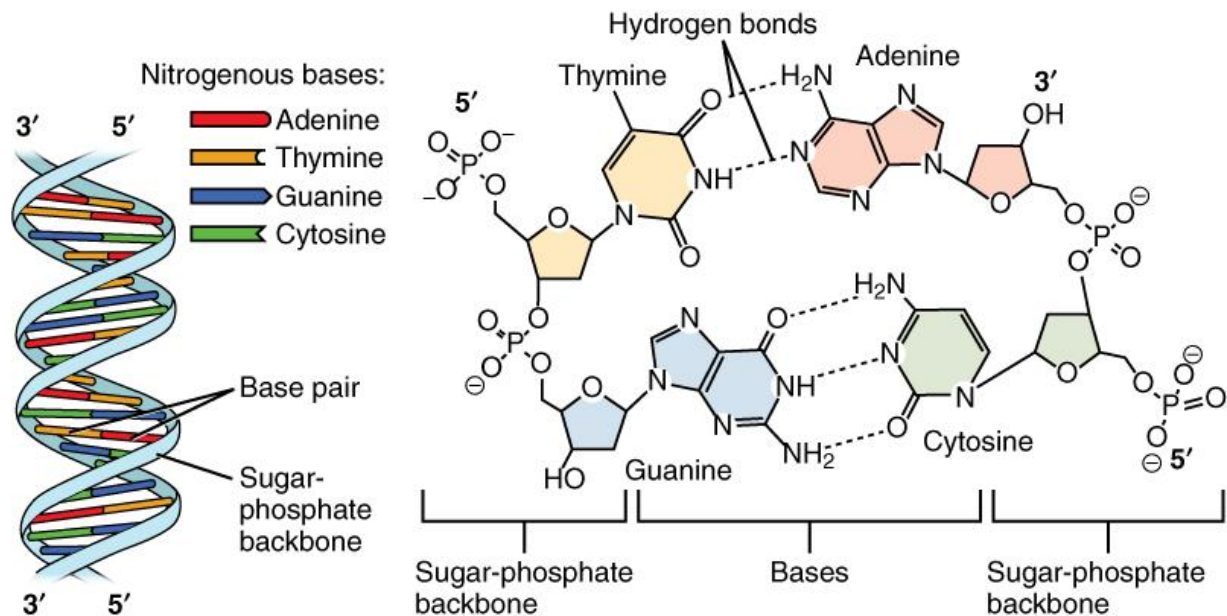


Figure 2: In the DNA double helix, two strands attach via hydrogen bonds between the bases of the component nucleotides. (OpenStax-College, 2015); Download for free at <http://cnx.org/contents/14fb4ad7-39a1-4eee-ab6e-3ef2482e3e22@7.30>.

A protein is a long chain of amino acids (AA). The order of the amino acids together with the way the protein is folded determines the function of the protein. The order of the amino acids in proteins are encoded into regions on DNA molecules called genes.

There are 20 amino acids, but only 4 letters occur in DNA. Hence, consecutive nucleotides in DNA are grouped together into units of three, called codons. There are $64 = 4 \times 4 \times 4$ unique codons, so that each amino acid is encoded by multiple codons. This mapping from groups of three nucleotides to amino acids is called the genetic code.

The information in DNA is used build a protein in two steps called transcription and translation. Transcription copies the gene from the DNA that encodes the protein into a ribonucleic acid (RNA) molecule. Translation builds the protein from the RNA molecule.

In order to transcribe a gene, an RNA polymerase binds to the DNA just before the start of the gene (upstream of the gene). The RNA polymerase proceeds along the gene towards the 3' end in the DNA and builds up a strand of RNA by building up a chain of nucleotides that are complementary to the nucleotides in the DNA. RNA is very similar to DNA except that at the 2' carbon of their sugars they

have an extra oxygen molecule attached, and that thymine does not occur in RNA, it is replaced with uracil.

Translation is the process by which protein is constructed by a molecule called a ribosome using an RNA molecule as a template. The process is initiated when the ribosome binds to the RNA molecule. The ribosome proceeds towards the 3' end of the RNA and builds up a chain of amino acids in the process. The amino acids added into the chain are determined by the codons present in the RNA molecule.

Using the terminology and processes defined in this section, the central dogma of molecular biology can be more precisely stated as: DNA codes for RNA, which is translated into protein. Additionally, such information cannot be transferred back from protein to either protein or nucleic acid (F. Crick, 1970). Information flow between DNA and RNA is complex in that DNA (RNA) can code for other DNA (RNA) and that DNA can also be constructed from an RNA template.

2.3 Basic Overview of HIV

Human immunodeficiency virus (HIV) infects humans, primarily targeting the CD4+ T cells of the host's immune system. Once these cells become infected they are destroyed which, as a result, weakens the immune system of the infected individual. This in turn can lead to infection by opportunistic pathogens, which ultimately results in death (Gottlieb et al., 1981; Masur et al., 1981). The advanced stage of the disease, where the immune system of the infected individual is severely compromised, is referred to as acquired immunodeficiency syndrome (AIDS)

HIV is a retrovirus, implying that its genetic information is encoded as RNA rather than DNA. The structure of an HIV virion is shown in Figure 3 A. The HIV replication cycle, shown in Figure 4, starts when a virion binds to a host cell (generally CD4+ T cells) (Doms & Trono, 2000). The envelope glycoprotein on the surface of the virion binds the surface protein, CD4, expressed on CD4+ T cell among others. This initiates a series of conformational rearrangements in the envelope protein which fuse the virion on CD4+ T cell membranes. Upon fusion the HIV capsid, which contains the viral RNA and enzymes, is injected into the host cell cytoplasm. Once inside, the viral RNA is reverse transcribed to DNA by the viral enzyme reverse transcriptase (M. D. Miller, Farnet, & Bushman, 1997) and the DNA is subsequently incorporated into the DNA of the host cell (Andrake & Skalka, 1996; LaFemina et al., 1992). The viral DNA incorporated into the host cell is called provirus which replicates by using the host cell's translation machinery to produce HIV proteins (Q. Zhou, Chen, Pierstorff, & Luo, 1998; Q. Zhou & Sharp, 1995). As part of this translational process, the HIV envelope protein is also covered in the same glycans that cover many of the host's cells. This "glycan shield" hides the surface features of HIV, allowing it to evade the immune system (Figure 4 B). These viral proteins assemble at the plasma

membrane of the infected cell after which a new virion buds from the host cell in order to find other host cells to invade (Kohl et al., 1988).

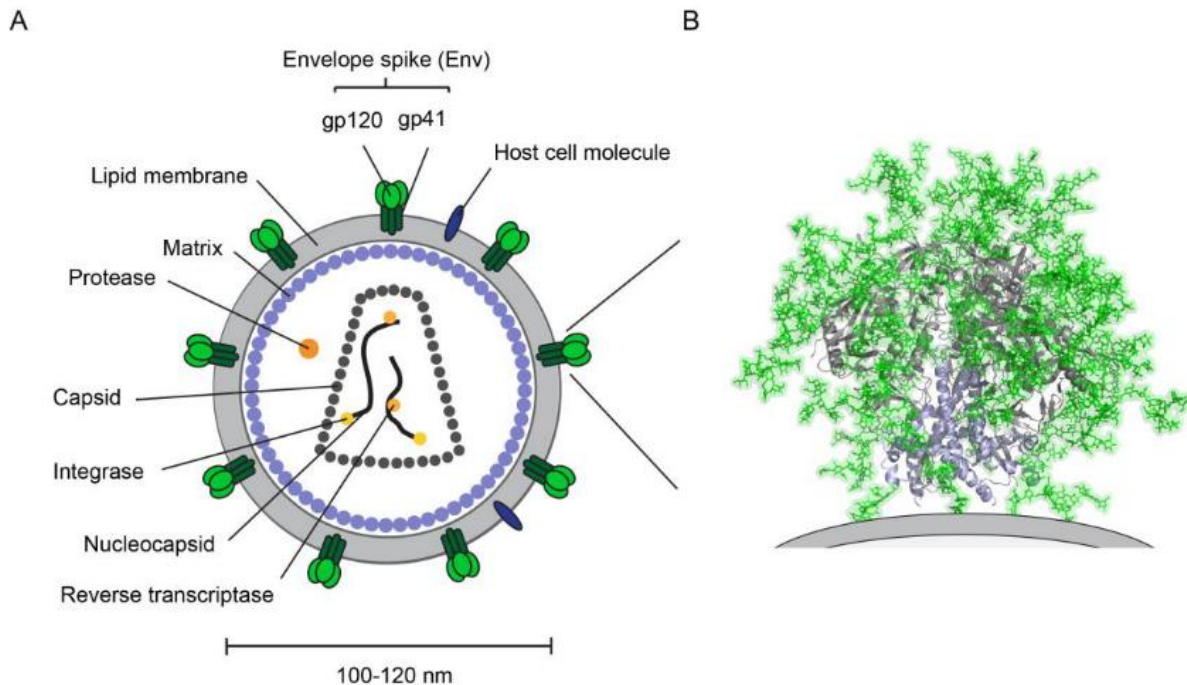


Figure 3: Figure 1.1 from (Behrens, 2017). The structure of the virion (A). Two copies of the viral RNA, the integrase enzyme and the reverse transcriptase enzyme are surrounded by the capsid. Structural integrity of the virion is provided by the matrix. The virion is surrounded by a membrane and the envelope spikes form protuberances on its outside. The envelope spikes are covered in glycans (B).

As described above, the HIV replication cycle requires the reverse transcription of the viral RNA to DNA which is facilitated by a viral enzyme, reverse transcriptase. While transcription and translation of non-viral DNA and RNA is very accurate due to various proofreading and error correction mechanisms (Fairbanks & Andersen, 1999), the reverse transcription process employed by HIV is less accurate since it lacks these mechanism (Bebenek, Abbotts, Wilson, & Kunkel, 1993). Inaccurate reverse transcription introduces many mutations (approximately 2.16×10^{-5} mutations per base per replication cycle (Mansky & Temin, 1995)), which could result in changes to the viral proteins. Since these mutations occur randomly, changes may yield a virion that is either less fit than the original or non-functional (Gao et al., 2004). However, mutations also occur that do not affect the fitness significantly, or which confers an evolutionary advantage (Rambaut, Posada, Crandall, & Holmes, 2004) Moreover, the replication cycle is short, lasting only 2.5 days (Perelson, Neumann, Markowitz, Leonard, & Ho, 1996). This high level of replication and the low level of fidelity of the reverse transcription process leads to a diverse population of virions in a single host (Rambaut et al., 2004) which is termed the viral quasispecies (Nowak, 1992).

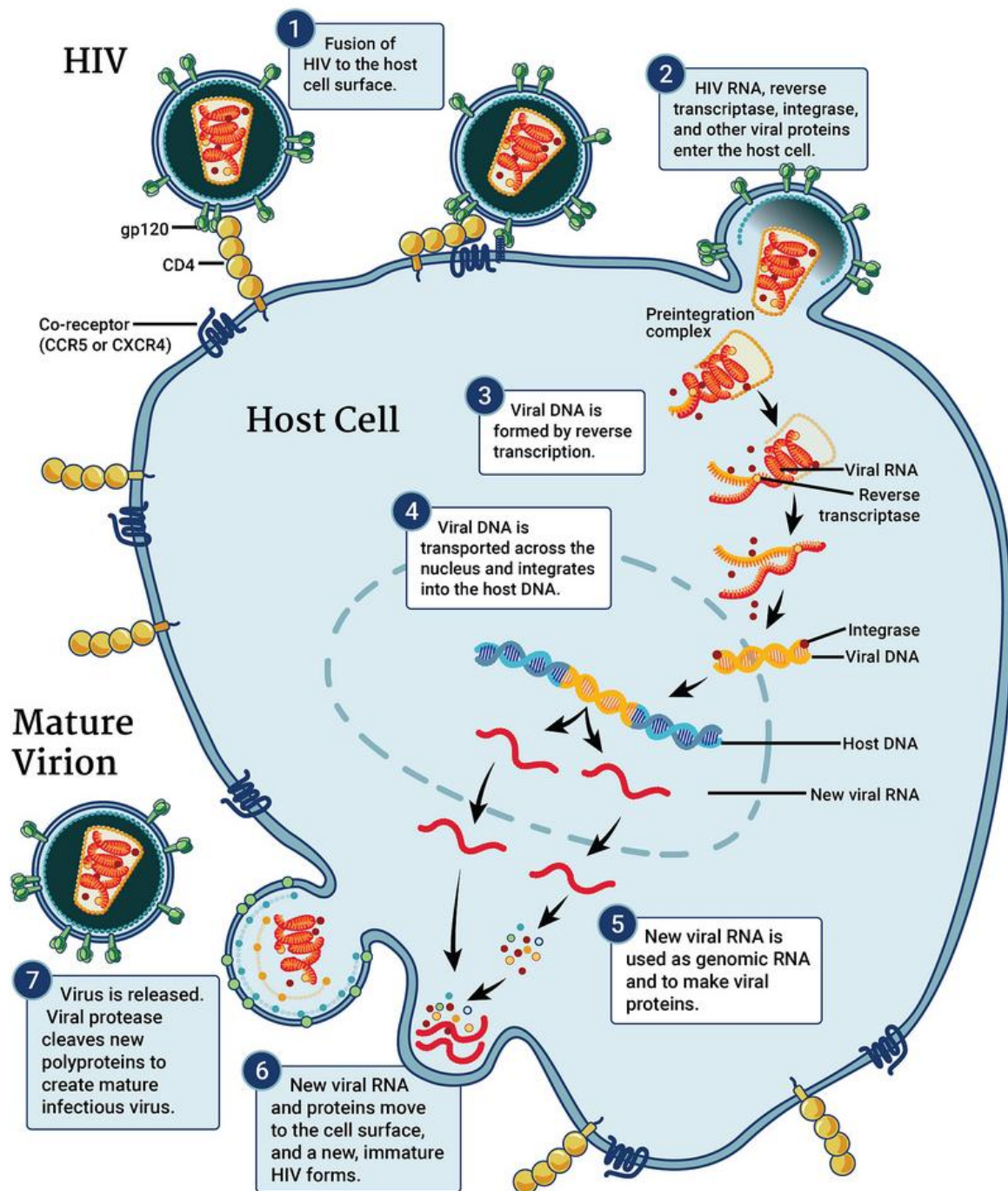


Figure 4: Figure 1 from (Pau & George, 2014). The HIV replication cycle.

Even though infections eventually become diverse, in the vast majority of cases the infection is founded by a single virus, defined as the transmitted founder (Keele et al., 2008). The initial phase of infection, during which so few viral RNA particles are present that they are undetectable is called the eclipse phase. After approximately 10 days of replication, sensitive RNA assays can detect the viral RNA (Figure 5). During this early stage, the immune response is limited and poorly targeted to HIV

allowing rapid replication. The amount of viral RNA present in the blood of the infected individual, called the viral load, increases rapidly leading to a phase of acute infection during which the patient may experience flu-like symptoms and the risk of spreading HIV is greatly increased (Simon & Ho, 2003).

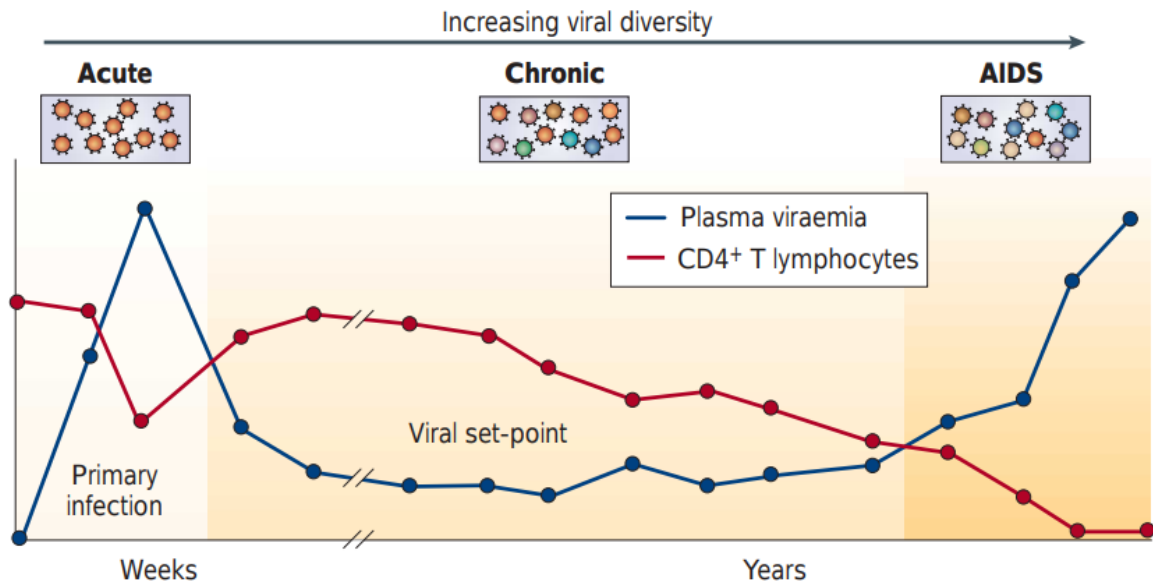


Figure 5: Figure 2 from (Simon & Ho, 2003). The natural course of HIV-1 infection on the basis of the longitudinal evolution of the two key surrogate markers — viral load (plasma viraemia) and CD4 count (CD4+ T-lymphocyte count).

During this acute phase, the immune system will start to produce antibodies targeted to HIV and the CD8+ cells will start to respond to the infection. Together, these responses reduce the severity of the infection and reduces the levels of ongoing viral replication. Eventually, the viral load stabilizes around a set point and few, or even no, symptoms are experienced. This phase may persist for many years, but the low level ongoing replication steadily reduces the amount of CD4+ cells in the infected individual. Eventually, the CD4+ cell count is reduced to such a low level that the immune system can no longer effectively protect against other pathogens, resulting in a multitude of symptoms related to opportunistic infections. An AIDS diagnosis is based on the number and severity of these opportunistic infections or a CD4 count below 200 cells per cubic millimeter of blood (Girard, Osmanov, Assossou, & Kieny, 2011).

2.3.1 Overview of HIV treatment

Various drugs have been developed to treat HIV infected individuals. In March 1987, azidothymidine (AZT), originally explored as a treatment for cancer during the 1960s, was approved by the FDA as a treatment for AIDS. Soon after, clinical trials showed that AZT effectively delayed the progression of HIV into AIDS, thus becoming the first treatment for HIV. Due to large investments in the subsequent 30 years an extensive suite of drugs that treat HIV have been developed. These drugs are referred to

as antiretroviral drugs with treatment commonly referred to as antiretroviral therapy (ART) (Pau & George, 2014). The website of the Food and Drug Administration of the United States of America that lists the available antiretroviral drugs for the treatment of HIV listed 40 drugs on the 2nd of August 2018. The drugs are grouped based on their function and disrupt one of the stages of the HIV replication cycle. The HIV drug groups and their functions are summarized in Table 1.

Table 1: Classes of antiretroviral drugs.

Name (Abbreviation)	Description	Number of drugs
Multi-class Combination Products	Combines a number of different drugs into a combination product that simultaneously targets more than one stage in the replication cycle.	5
Nucleoside Reverse Transcriptase Inhibitors (NRTIs)	Inhibits the activity of reverse transcriptase, preventing the viral RNA from being converted into DNA. NRTIs mimic the natural ACGT nucleotides, but lack some of the structures that attach different nucleotides to each other. Hence they compete with the normal nucleotides for incorporation and when one of them is used by the reverse transcriptase enzyme, then the process stops.	13
Nonnucleoside Reverse Transcriptase Inhibitors (NNRTIs)	Like NRTIs, they interfere with the reverse transcription step. These drugs directly bind to the reverse transcription enzyme disabling it.	6
Protease Inhibitors (PIs)	Binds selectively to viral proteases and prevents them from producing the precursor molecules needed to produce viral particles.	11
Fusion Inhibitors	Prevents the merging of the HIV envelope and the membrane of the CD4 cell.	1
Entry Inhibitors	Antagonizes a receptor on the surface of a CD4 Cell (the CCR5 receptor), preventing the HIV virion from forming the bond with the CD4 Cell that is required for the envelope and membrane to fuse.	1
HIV integrase strand transfer inhibitors	Blocks the incorporation of the viral DNA into the host's genome by interfering with the HIV enzyme called integrase.	3

Although the use of a single drug can control the infection, a high dose is required and it is frequently unsuccessful. Additionally, resistance to a single drug can easily arise in a patient. Therefore a number of drugs are combined, reducing the required dose of any individual drug required leading to reduced side effects, substantially increasing efficacy and reducing the risk of resistance emerging. This multi-drug treatment regime is referred to as highly active antiretroviral therapy (HAART). Proper adherence to an HAART routine result in a life expectancy of infected individuals which is similar to that of uninfected individuals (Pau & George, 2014).

Some of the cells infected with HIV are not actively producing HIV and are said to be in a latent state. All the drugs listed in Table 1 inhibit ongoing replication of HIV, however they do not destroy these inactive infected cells. Thus, HAART only reduces the number of active HIV virions present as well as the rate at which cells are infected without impacting the latent cells. These latent cells are distributed throughout the body, and can remain latent for many years after which they can become active. Therefore, even if treatment is effective enough to reduce the viral load of a patient to undetectable levels, treatment interruption results in the infection rebounding (Chun et al., 1997).

2.3.2 HIV drug resistance

Apart from treatment interruption, drug resistance can also result in a rebound of the infection. After prolonged treatment drug resistance often arises in the quasispecies. Due to the increased fitness of these strains in the presence of treatment, these variants become the dominant subpopulation in the quasispecies, leading to treatment failure. In addition to suppressing the level of replication, HAART reduces the chances of drug resistance arising by requiring that multiple mutations need to arise simultaneously to confer resistance to the treatment regimen. However, various factors such as poor access to medication in resource limited settings, poor adherence and various other political and personal factors leads to imperfect treatment (Nachega et al., 2011).

Low adherence to treatment leads to higher levels of viral replication in the presence of low levels of various drugs. In these scenarios, the higher replication levels allow for high numbers of mutations to arise and the sub-therapeutic drug concentrations confers a modest fitness advantage to the resistant variants. This allows drug resistant variants to arise at a much faster rate than what would have been possible with proper treatment. Furthermore, drug resistance strains can be transmitted, leading to cases where an individual's baseline infection is a drug resistant strain. Epidemiological studies of drug resistance are finding increasing rates of first time infections with drug resistance and they predict that drug resistance will become an increasingly serious problem. According to the World Health Organization's 2017 HIV drug resistance report, a survey of 11 poor counties, between 2014 and 2016, found that in 6 of these counties more than 10 percent of cases were drug resistant.

Treatment failure due to drug resistance requires that the patient be switched to a different treatment regimen. However, first-line drugs are cheaper and have fewer side effects than these alternative treatments. The challenges associated with drug resistance is further exacerbated in resource limited settings where access to second and third-line treatment strategies can be limited (Nachega et al., 2011).

2.3.3 HIV vaccine

While the availability of HAART has greatly reduced the spread of HIV and markedly improved the health outcomes of infected individuals on treatment, it is not a cure. Additionally, the complex treatment regimens and side effects causes low adherence which in turn drive the development of drug resistance. Thus, there is continued interest in the development of a vaccine. A vaccine exposes the immune system to inactivated or weakened (attenuated) portions of a pathogen, inducing the production of neutralizing antibodies targeting that vaccine. If a vaccine can successfully elicit potent antibodies, then the immune system can rapidly respond to the pathogen the next time an infection occurs. This rapid response is critical since it enables the immune system to stop the infection before there is a large number of virions and before the latent reservoirs can be seeded (McMichael & Koff, 2014).

A key requirement for a vaccine to work is that the immune system must be able to produce antibodies that effectively target the pathogen. However, the antibodies produced by most HIV positive individuals are non-neutralizing and target regions of the virion that mutate rapidly. Thus, vaccines that expose the body to specific portions of certain HIV variants are unlikely to be effective since the antibodies they induce are either not potent enough, can be easily evaded by a small number of mutations, or there are already a high enough percentage of circulating virus in the human population that contains escape mutations to that vaccine (Girard et al., 2011).

In addition to the difficulty of eliciting effective antibodies, the search for a vaccine faces another major obstacle. Conducting HIV vaccine trials are extremely expensive and time consuming. To conduct any vaccine trial, a number of subjects must be enrolled and divided into two groups. One group is given the vaccine and the other is given a placebo. The two groups are monitored for infection events and if the number of infections occurring in the placebo group is statistically greater than in the vaccinated group then the trial is considered successful. In the general population, the incidence of HIV is very low. Thus, to obtain enough infection events in the placebo arm, a very large number of subjects must be enrolled and they must be followed for a long time, greatly increasing the cost.

The high cost of vaccine studies, combined with the difficulty of designing a vaccine that is likely to be effective, means that very few HIV vaccine trials have been conducted to date. Out of the five large

scale vaccine efficacy trials that has been conducted, only one showed a possible protective effect. The RV144 trial injected 16,402 subjects with a prime-boost recombinant vaccine and demonstrated a 31.2% efficacy in preventing HIV infection. The mode of protection is still unknown, but theorized to be related to the antibodies that were elicited (Girard et al., 2011).

During natural infection, a small number of HIV infected individuals develop potent antibodies that target regions of the virion that are conserved and are capable of neutralizing a broad spectrum of viral strains. One such antibody, VRC01, targets the conserved region on Env that binds to the CD4 receptor on T cells (T. Zhou et al., 2010). VRC01 is both potent and broad, neutralizing 90% of HIV-1 isolates across all clades of HIV-1. The theorized protective effect of the antibodies motivated the launch of the AMP study.

The AMP study will infuse HIV-negative patients via intravenous drip with either VRC01 antibodies or a placebo every two months and monitor for HIV infections in both arms. The primary endpoint of the study is to compare the number of infections that occurred between the two treatment arms. If successful, this will demonstrate that the presence of a potent broadly neutralizing antibody is effective at reducing the risk of infection (Gilbert et al., 2017). Since the investigation product is an expensive infusion that needs to be administered every two months, the AMP trial will not result in a feasible public health intervention.

The value of the trial, however, will be in establishing a correlate of protection. If a vaccine can be derived that induces a person's immune system to produce the VRC01 antibody, then the result of the AMP study (if successful) will provide strong evidence that such a vaccine will be effective, provided the antibody is elicited at a high enough concentration. This will greatly reduce the cost and time required to test vaccine candidates, since a modest number of subjects can be injected with the vaccine candidate and the concentration of broadly neutralizing antibody in their blood can be measured soon after. Additionally, a successful result from the AMP trial will provide evidence that if a vaccine candidate can elicit any potent broadly neutralizing antibodies, then the vaccine candidate may be efficacious. If this result can be confirmed by duplicating the AMP study with another antibody, then the cheaper and faster trials may be performed for any broadly neutralizing antibody instead of only for VRC01 (Gilbert et al., 2017).

The concentration of the VRC01 antibody in the patient's blood increases rapidly to a high level during the infusion. The antibody is continuously cleared from the body of the patient meaning that the concentration will start declining immediately upon cessation of the infusion and will reach a lowest level just prior to the next infusion. Thus, in order to relate the protective effect of the antibody to the

concentration of the antibody, it is useful to know the time of HIV infection with as little uncertainty as possible (L. Zhang, Gilbert, Capparelli, & Huang, 2018).

As previously described, upon infection the virus starts to replicate producing increasing levels of viral RNA in the patient's blood (viral load). As the viral load increases, the body also recognizes the infection and produce antibodies. HIV requires a period of time after infection to replicate to such levels that the RNA can be detected. This period of time is different from the amount of time it takes the immune system to produce enough of these antibodies to be detectable by an assay. Thus by using both tests (RNA based and antibody based), information about the time of infection can be gleaned. For example, if the person tests positive on the RNA test but not on the antibody test, then the infection was so recent that enough antibodies were not yet produced, but far enough into the past that enough replication could have happened for the RNA test to be positive (L. Zhang et al., 2018).

A possible approach to increase the accuracy of the infection timing is to analyze the sequences of a sample of the HIV population in a patient. Poisson Fitter available from the LANL website implements such an approach (Elena E Giorgi et al., 2010). Poisson Fitter is reviewed in detail in section 5.5. Briefly, by comparing the number of sequences to each other, the amount of mutation that has occurred since infection can be tallied. Using previously published parameters of the mutation rate of HIV, the number of mutations can be converted into a number of generations of HIV replication that has occurred since HIV infection. Using knowledge of the replication cycle of HIV, the number of generations can be converted into a time estimate. This approach and other approaches to investigating the diversity of an HIV population are explored in Chapter 4.

2.4 Sequencing

The process by which the sequence of nucleotides in a molecule of DNA are identified is called sequencing. The proliferation of DNA sequence data led to the creation of the field of bioinformatics to process and analysis all of the resulting data (Ouzounis & Valencia, 2003).

2.4.1 The polymerase chain reaction

The two sequencing techniques discussed in this work both require large amounts of target DNA to be present. Hence the region of interest must first be amplified using polymerase chain reaction (PCR) (R. Saiki et al., 1988).

The reaction occurs in a solution containing the DNA templates of interest; a polymerase which copies DNA; deoxynucleotide triphosphate (dNTP) and primers. Primers are short DNA sequences (oligonucleotides) which are designed to be complementary to two specific positions of a DNA sequence. The region between these two specific positions will be amplified by the reaction. PCR is a

cyclical process in which three steps are repeated a number of times. Figure 6 illustrates a single cycle of PCR. First the DNA double helix is separated (denatured) by heating the solution to over 90°C. Next the solution is cooled to under 60°C allowing the primers to anneal to their complementary regions. The last step is to heat the solution to 72°C where the polymerase enzyme functions best, allowing the new strands of DNA to be synthesized. The synthesis process is described in detail in the next paragraph. The most commonly used polymerase is the Taq polymerase derived from the organism *Thermus aquaticus* (R. K. Saiki et al., 1985). Exact details of the protocols differ slightly between laboratories (Brodin et al., 2015; Jabara, Jones, Roach, Anderson, & Swanstrom, 2011; Kinde, Wu, Papadopoulos, Kinzler, & Vogelstein, 2011; Kou et al., 2016; S. Zhou, Jones, Mieczkowski, & Swanstrom, 2015).

Synthesis occurs when a primer will anneals to these specific position and the polymerase binds to the DNA sequence where a primer annealed. The polymerase will then copy the DNA sequence forming a new molecule that starts with the primer and ends at some position towards the 3' end from primer binding site. The position where the molecule ends is determined by the DNA sequence and the parameters of the reaction. This position is either the end of the DNA sequence, or the position where the polymerase detached from the DNA sequence interrupting the process or the position where the polymerase was when the PCR cycle was stopped. Since DNA molecules have an orientation and the polymerase can only copy DNA in one direction, a newly copied strand will be duplicated in the opposite direction in the next PCR cycle. Thus in the following cycle, the other primer will anneal to the molecule and it will be copied towards the region to which the previous primer was bound. As this process is repeated only the region between the two primers is amplified.

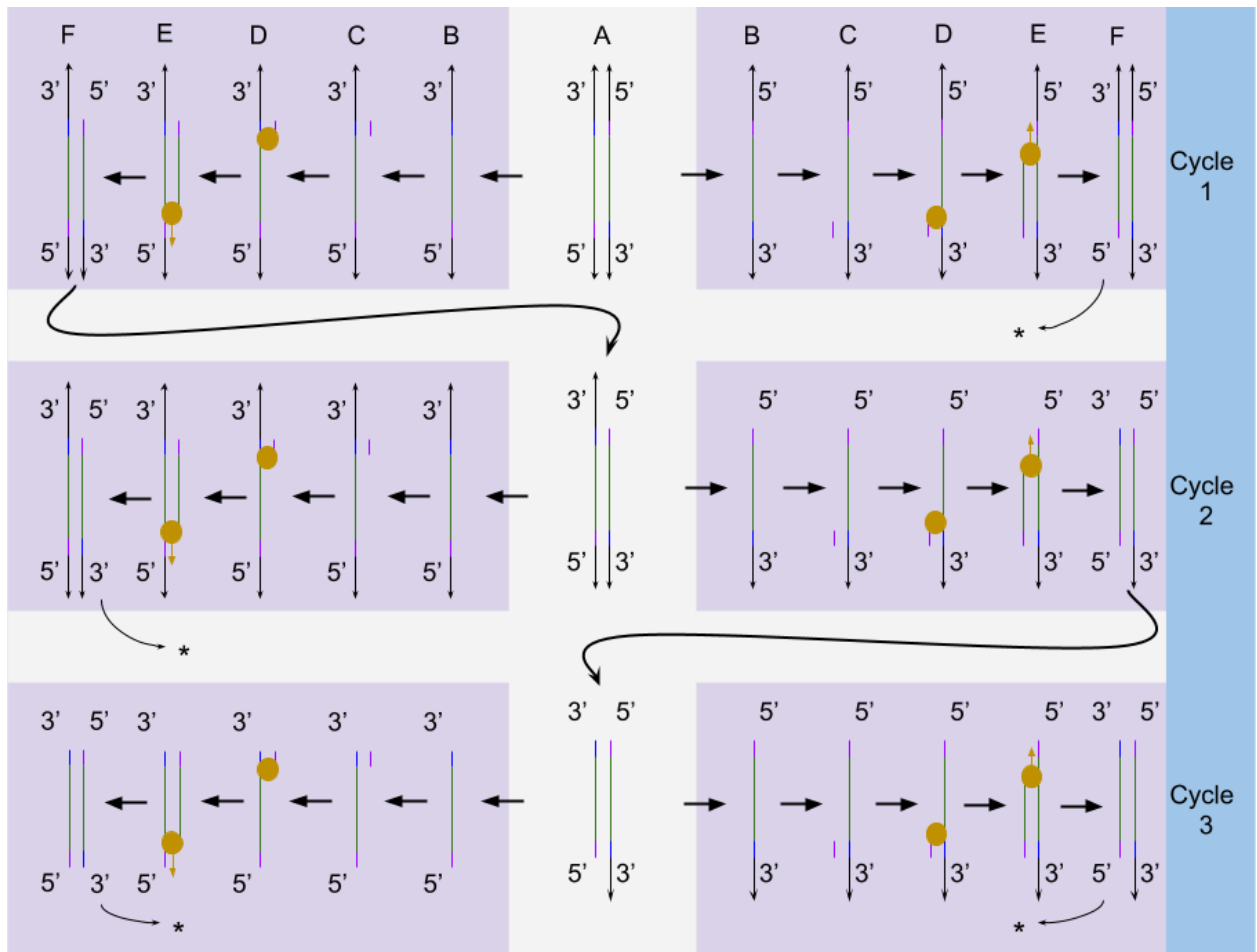


Figure 6: The first three cycle of PCR. The process is initiated with a double stranded DNA molecule (A) which is denatured by heating the solution (B). Cooling the solution allows primers to bind to the denatured DNA (C) producing an attachment site for the polymerase enzyme (D). Heating the solution again allows the polymerase to synthesize a complementary strand (E) yielding a double stranded DNA molecule (F) which can be denatured again to initiate the next cycle of PCR (* or the subsequent row in the figure). Note how the size of the molecule that is amplified is reduced by the restrictions placed on it by the primers.

It is important to note that PCR amplifies DNA in a biased way. At each cycle of PCR only a portion of the DNA molecules are copied. This random sampling of molecules at each cycle leads to a final distribution of molecules that is different from the distribution in the original sample. If the match between the primer and the target region on the genome is poor, the rates at which the primers anneal is much lower during the initial rounds, suppressing amplification efficiency. If two similar templates are amplified together in the same reaction and their primer binding regions are identical except that at one position one template has a adenine instead of a cytosine in the forward primer and a thymine instead of a cytosine in the reverse primer, the template with the cytosine amplified between 40% and 120% more efficiently due to the higher binding energy between guanine and cytosine than between adenine and thymine (Polz & Cavanaugh, 1998).

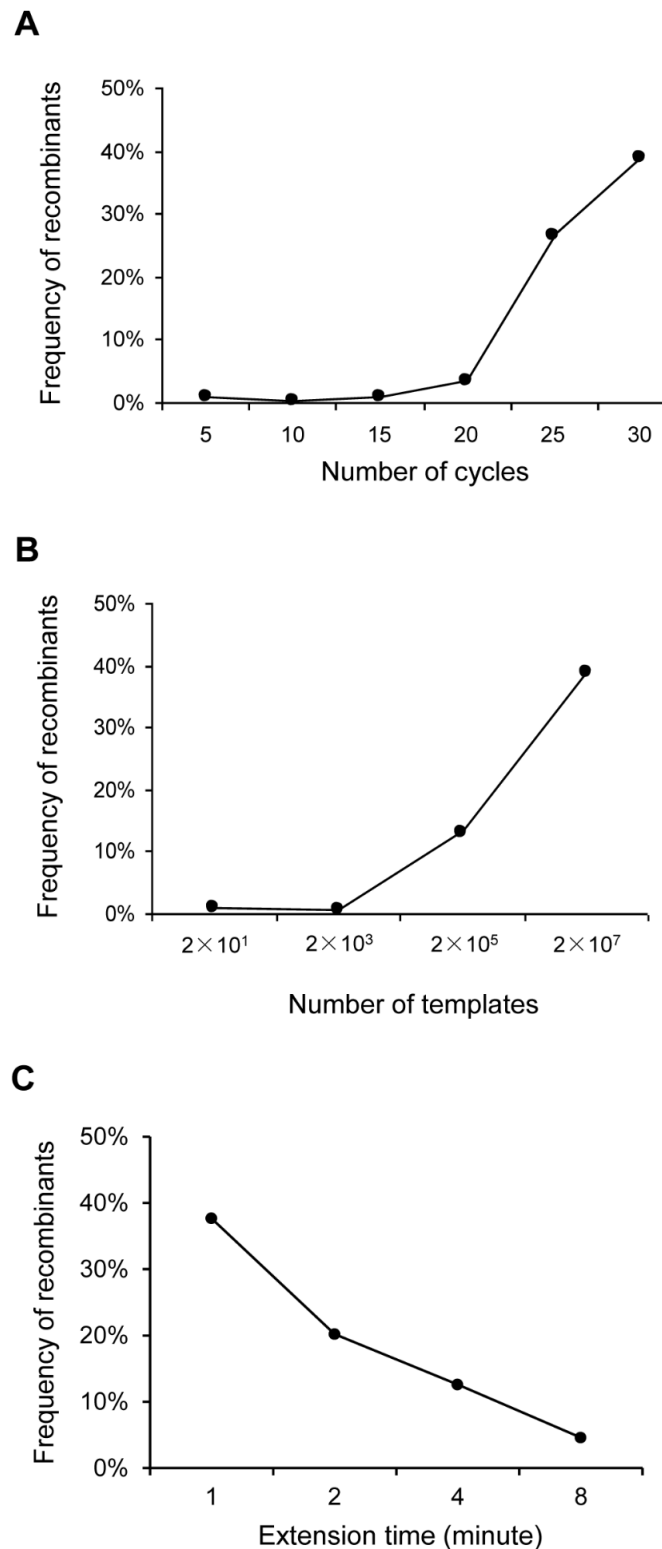


Figure 7: Figure 2 from Liu et al 2014. Recombination frequencies at different conditions during PCR. (A) Recombination frequencies were determined at different thermal cycles. Equal amount of NL4-3 and 89.6 plasmids (107 copies per template) were mixed together and co-amplified. The PCR was carried with 5, 10, 15, 20, 25 or 30 thermal cycles. (B) Recombination frequencies were determined with different numbers of templates. Equal amount of NL4-3 and 89.6 plasmids (101,103,105 or 107 copies each) was mixed together and co-amplified by 30 cycles of PCR. (C) Recombination frequencies were determined with different extension time. Equal amount of NL4-3 and 89.6 plasmids (107 copies per template) were mixed together and co-amplified. The PCR was carried with different extension time (1, 2, 4 or 8 minutes). The PCR products were analyzed by the PASS assay and the recombination frequency at each condition was determined by linkage analysis of six bases. doi:10.1371/journal.pone.0106658.g002

Another artifact that degrades the quality of product produced by PCR is that chimeric molecules can be generated. In such molecules the product built up by the polymerase contains the genetic material from one template in certain regions and the genetic material from another template in other regions. The most common version is when one side of the molecule is from template one and then a switch occur such that the other end of the molecule is from another template. (Kanagawa, 2003) describes multiple processes that can produce such chimeric molecules. ~~We will only focus the single process that produces the majority of chimeric molecules~~Of all the processes that generate PCR recombination, we will restrict our discussion to a single one which is responsible for the majority of the recombination (J. Liu et al., 2014). The polymerase may fall off from the molecule that it is busy copying, yielding a partially synthesized molecule. This incomplete sequence can then act as a primer for the next cycle of PCR. A chimeric sequences will be produced since the partially copied product will be extended with the polymerase adding the genetic information from another template onto the partial product. The rate at which recombination occurs during PCR is strongly influenced by the PCR parameters. Most chimeric sequences are produced in the latter stages of the reaction when the concentration of primers are lower and the concentration of amplified products are higher (Kanagawa, 2003). Thus recombination rates are increased when the number of cycles are higher (Figure 7 A) or when the number of molecules with which the reaction is seeded are higher (Figure 7 B). Furthermore, increasing the duration of the elongation step reduces the number of partially synthesized molecules also reducing the rate at which recombinant sequences are produced (Figure 7 C).

2.4.2 Sanger Sequencing

The first widely adopted approach used to sequence DNA was described in (Sanger, Nicklen, & Coulson, 1977). Originally, to perform Sanger sequencing, a PCR solution was prepared and divided into four equal samples. To each of the samples, one type of dideoxynucleotide triphosphates (ddNTP) is added, so that one sample contains ddATP, the next contains ddCTP, and so forth. After incorporating a ddNTP instead of a dNTP into a new copy of a DNA molecule, the polymerase enzyme is unable to continue extending the nucleotide chain.

Since only a small amount of ddNTP is added, each of the four reactions will produce copies of the input DNA templates that are randomly terminated at one of the positions containing the base corresponding to the ddNTP added to that sample. By determining the lengths of the DNA molecules in each of the four samples, the positions of the bases in the DNA sequencing can be inferred. The lengths of the molecules are assessed using polyacrylamide gel electrophoresis. The four samples are run through the same gel in separate channels causing bands to appear in each channel at positions

that are indicative of the relative lengths of the DNA molecules (Figure 8). When the information is transferred from the gel to digital storage, the sequence of digitally stored letters is called a read.

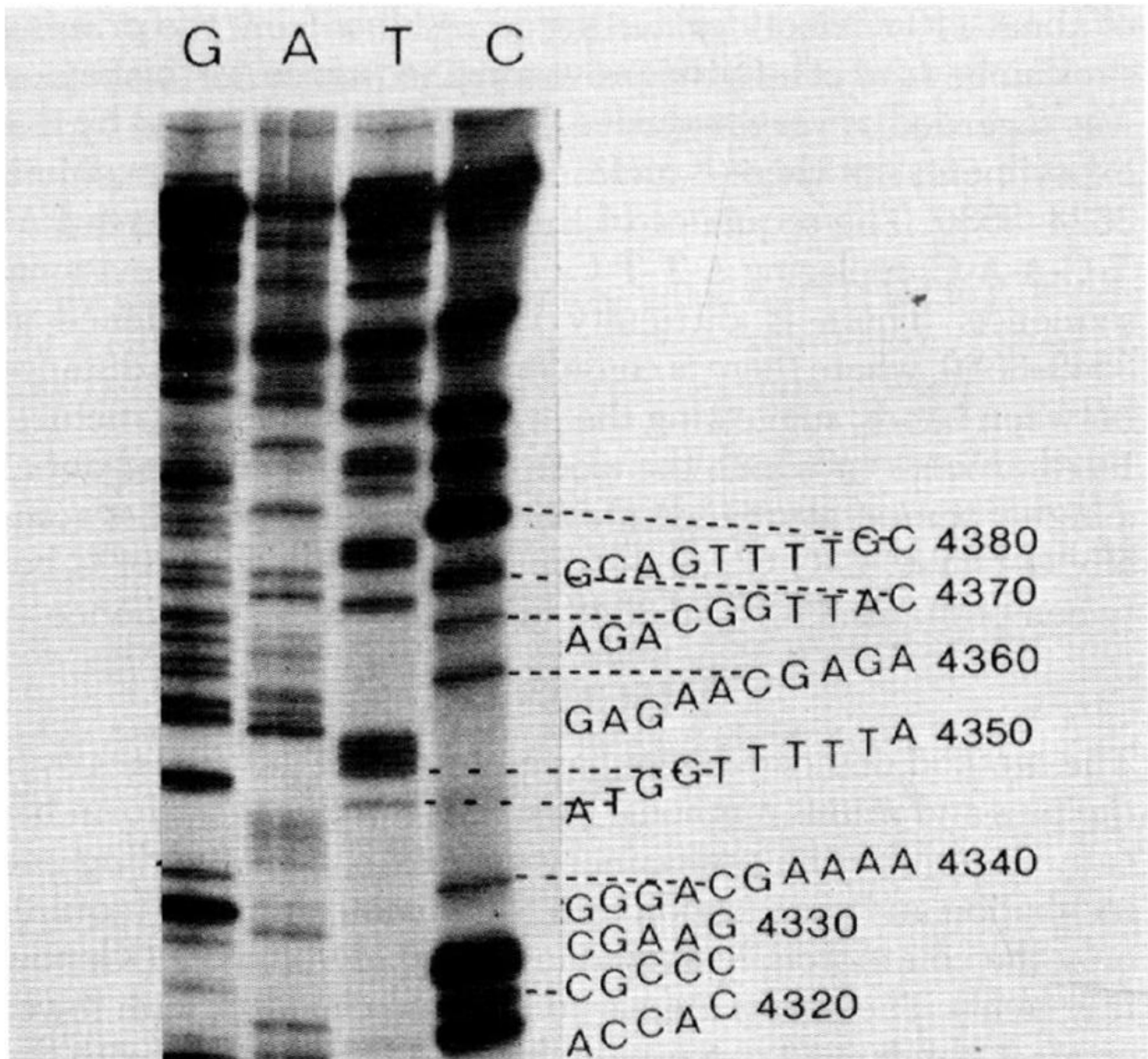


Figure 8: Example of gel obtained after performing Sanger sequencing. The DNA sequence is written from left to right and upwards. The position of the nucleotides in the genome of the organism is also provided. The dotted lines connect bases in the sequence with the bands from which they were read. Figure from (Sanger et al., 1977).

As reviewed in (Dovichi & Zhang, 2000), optimizations to the original process includes the attachment of fluorescent dyes to the ddNTPs and replacing the manual gel electrophoresis step with an automated step. These improvements increased the throughput of this sequencing technique enough to allow sequencing of the entire human genome. While slow and expensive, Sanger sequencing is still used because it produces long high quality reads. The top-of-the-line 3730xl DNA Analyzer machine produced by Applied Biosystems produces reads between 400 and 900 bases in length with an accuracy of 99.999% (L. Liu et al., 2012).

2.4.3 Single Genome Amplification and Sequencing

In order to apply Sanger sequencing to viral RNA, it must first be converted to DNA. DNA derived from RNA is referred to as complementary DNA (cDNA) (Palmer et al., 2005). The viral RNA isolated from the blood plasma contains the RNA of many different individual virions from the quasispecies. If Sanger sequencing is applied directly to cDNA produced from such a sample, then the diversity of the quasispecies will not be reflected in the final dataset since only a single sequence will be obtained for the entire sample. This sequence will consist of the bases that occur most frequently at each position in the region that was sequenced and is called the consensus sequence (Thomas et al., 2006).

In order to obtain DNA sequences of individual members of the quasispecies, single cDNA molecules must be isolated from the sample before they are amplified with PCR. This is performed with a time consuming and expensive process called single genome amplification (SGA). SGA is performed by diluting the sample and running separate PCR reactions on sub-samples from the dilutions (Butler, Pacold, Jordan, Richman, & Smith, 2009).

The degree of dilution is varied until the PCR reaction in only 30% of the diluted sub-samples successfully amplifies the DNA in that sub-sample. The most probable reason for the other 70% of the sub-samples failing to amplify is that there is no DNA in those sub-samples due to the high level of dilution. The number of DNA molecules in each sub-sample follows the Poisson distribution. The probability mass function of a Poisson distributed random variable with a rate parameter of 0.35 is 0.70 and 0.25 at the values of 0 and 1 respectively. Hence if 70% of the sub-samples did not contain DNA, then of the sub-samples with DNA, approximately 80% ($\frac{0.25}{1-0.70} = 0.83 \approx 0.8$) of the sub-samples will contain DNA from a single input DNA template. The DNA from the successfully amplified sub-samples can be sequenced with Sanger sequencing (Butler et al., 2009).

2.4.4 High throughput sequencing

Newer sequencing technologies, referred to as next generation sequencing (NGS) were developed that produce more data in a single run than Sanger sequencing (L. Liu et al., 2012). These technologies enable the sequencing of individual viral templates at a lower cost while being less labor intensive than SGA (Jabara et al., 2011). While several NGS platforms exist, this work will only discuss the approach used by Illumina for their MiSeq machines as this is the primary sequencing technology used with the PID approach due to the large amount of sequence data they produce, the length of their reads and favorable error profiles. (Bhiman et al., 2015; Kinde et al., 2011; Kou et al., 2016; Lundberg, Yourstone, Mieczkowski, Jones, & Dangl, 2013; S. Zhou et al., 2015).

The Illumina process was reviewed in (Voelkerding, Dames, & Durtschi, 2009) and for the applications discussed in this work starts by amplifying the sample with PCR (Brodin et al., 2015; Lundberg et al., 2013; S. Zhou et al., 2015). The primers used in the PCR step contain an adaptor sequence that enables the amplified DNA fragments to be anchored to a slide whose surface is covered in oligonucleotides that are complementary to the region added during PCR (Figure 9).

After attaching the DNA to the slide, it is amplified again so that copies of each sequence is made and attached close to the originally attached sequence. This process is called bridge amplification and produces a slide covered in spots each with identical sequences. Sequencing is performed by attaching primers to sequences on the slide, synthesizing strands complementary to the sequences on the slide and recording each base that is incorporated into the newly synthesized strands.

This process, called sequencing by synthesis, is performed by using special nucleotides that contain a fluorescent marker that emits colored light when excited by a light source and a removable modification that blocks the synthesis process. Each type of nucleotide emits a different color. After attachment of the primers, the slide is washed with a solution that contains these special nucleotides. Only one nucleotide can be incorporated into the newly synthesized strands because of the modification that blocks addition of more nucleotides. The solution containing the special nucleotides is washed away and the fluorescent markers are read. For each spot on the slide this reveals which nucleotide was incorporated first and hence what the sequence is at the first position for each spot.

To read the next position, the slide is washed in a solution containing enzymes that remove the modification that blocks the synthesis process and the fluorescent marker from the previously synthesized nucleotides. The slide is washed in the in the solution with the special nucleotides again to begin the process of reading the second position in the sequence. By repeating these steps, the sequence of the DNA molecules attached to each of the spots can be determined.

As this process continues, some sequences fail to incorporate a nucleotide in some cycles. When this occurs, this sequence becomes out of phase with the other sequences at the same spot, lagging 1 or more positions behind. At the n th cycle, the lagging sequence will incorporate the nucleotide for the $(n-x)$ th position in the sequence, where x denotes the number of failed incorporations for the sequence in question. The desynchronization degrades the quality of the signal produced when the fluorescent dyes are excited and ultimately limits the number of positions that can be read. Read lengths on the MiSeq machines with reagent kit v3 used to generate the example data for this work are 300 bases long (Illumina, 2015).

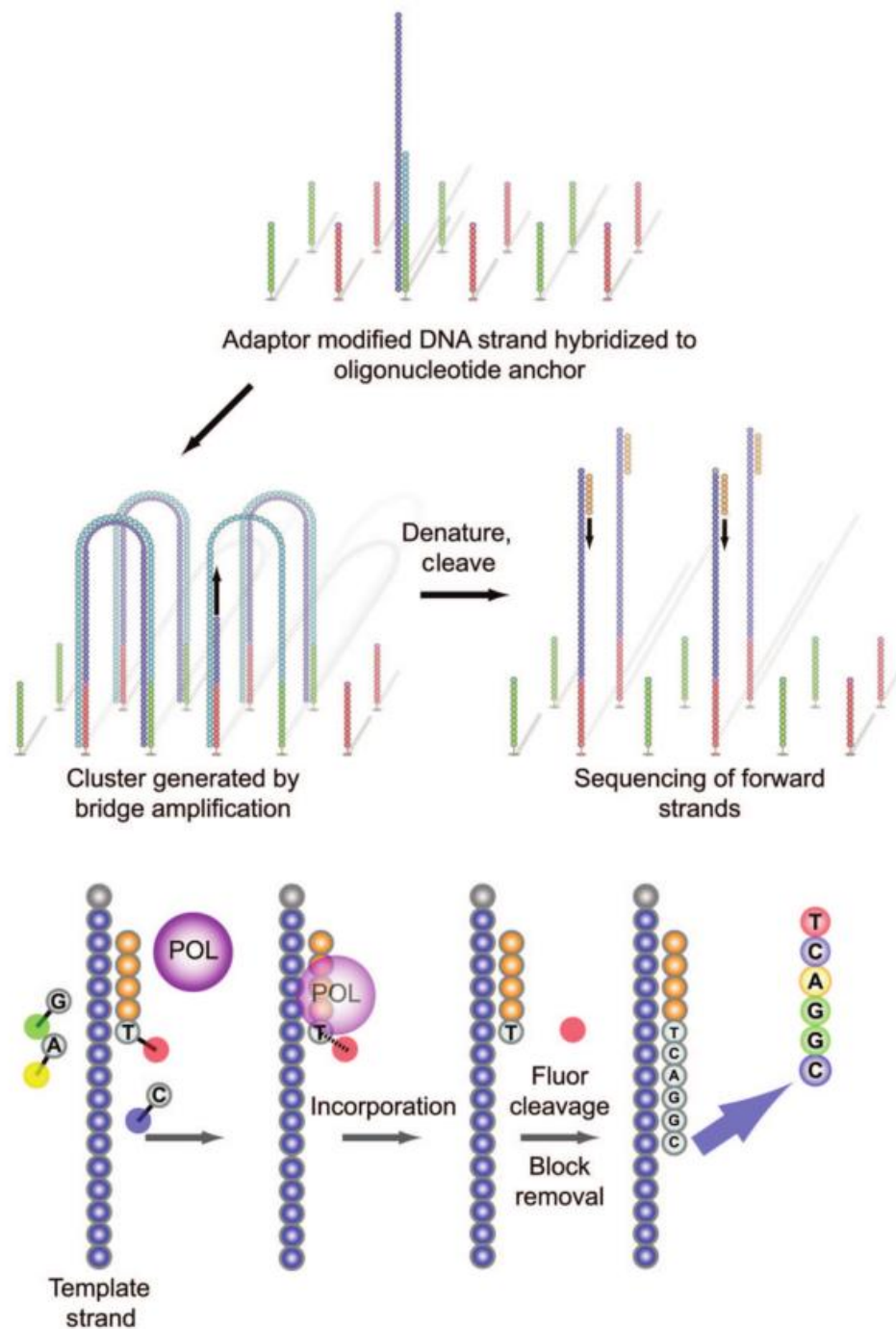


Figure 9: The Illumina sequencing process. Single stranded DNA with a sequencing platform specific region are attached to the slide. Bridge amplification produces spots of identical sequences. The bridge amplified sequences are denatured and sequencing is initiated with the addition of primer, polymerase (POL) and ddNTPs with the removable blocking modification and the fluorescent dye. After incorporation the fluorescence is recorded. The blocking modification and fluorescent dye are removed before the next synthesis cycle. Figure from (Voelkerding et al., 2009).

It is possible to attach and bridge amplify sequences longer than 300 bases to an Illumina slide. Benefit can be derived from using longer sequences, since the attached sequences can be read from either end. The first sequencing run reads the sequences from the 5' end for 300 bases in the direction of

the 3' end. A second run can be performed starting at the 3' end, reading 300 bases in the region of the 5' end. In cases where the attached sequences are shorter than 600 base pairs, the two reads will overlap in the central part of the sequence yielding overlapping paired end reads.

Illumina sequencing is more error prone than Sanger sequencing. In contrast to Sanger sequencing's accuracy of 99.999% (L. Liu et al., 2012), the MiSeq platform's accuracy is around 99% depending on various factors and processing strategies. The most common error type is a substitution error which occurs when a base is miscalled as another base. Insertions (when a base is spuriously inserted into a sequence) or deletions (when a base in the sequence is skipped) occur roughly 2 orders of magnitude less frequently than substitutions. Together insertion and deletions are referred to as indels. A detailed investigation of the sequencing error rates of the MiSeq platform, (Schirmer et al., 2015), reported substitution error rates ranging between 0.00157 and 0.0187 (Figure 10 A), insertion rates from below 0.000002 to 0.00123 (Figure 10 B) and deletion rates ranging from smaller than 0.000006 up to a maximum of 0.000712 (Figure 10 C).

A key observation reported in (Schirmer et al., 2015) is that the errors are non-random. From Figure 10 it is clear that library preparation has a significant influence on the errors. Additionally, the location in the sequence can affect the error rates with the first 10 bases and the last bases having higher error rates. Certain errors occur very frequently, for example, in one dataset, 25% of all substitution errors occurred when the A at position 226 was misread as a G. Under the assumption that the error is independent of position and nucleotide, this specific error is expected to account for only 0.133% ($\frac{1}{250} \times \frac{1}{3}$; sequence length of 250) of all errors. When the 3mers occurring directly before an error was analyzed, it was observed that the same 3mers directly preceded up to 75% of the errors in the datasets. However, the 3mers that preceded the errors were not consistent across all datasets (Figure 11).

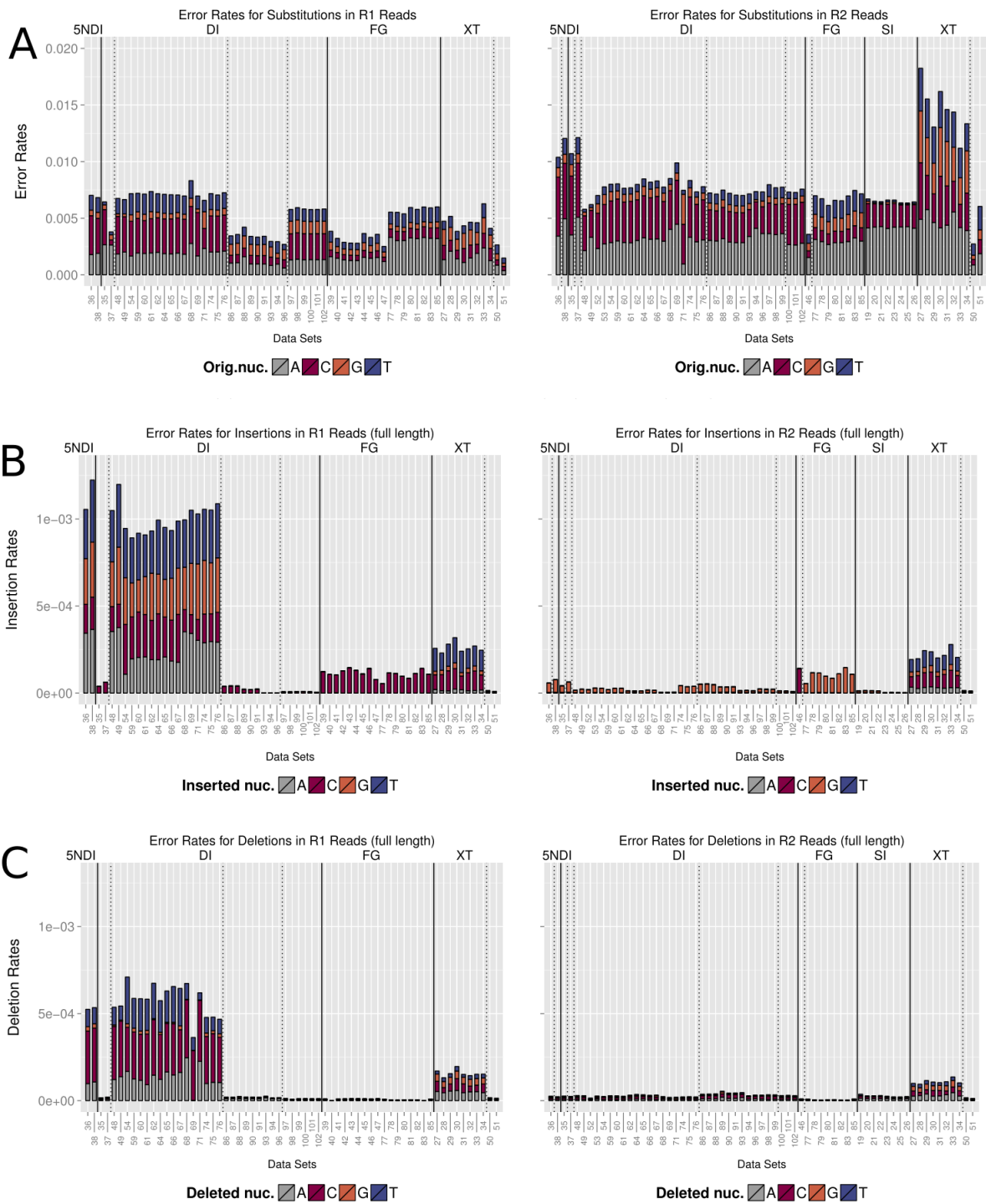


Figure 10: Comparison of substitution (A), insertion (B) and deletion (C) error rates for a number of datasets, library preparation techniques and primers. The lower x-axis indicates the number of the dataset used, the upper x-axis indicates the library preparation method grouped by the solid vertical black lines. The dotted vertical black lines blocks of datasets that utilized the same primers. The heights of the bars shows the amount error in the dataset and the colors allocate the errors to the nucleotide at the position where error occurred. Figure from (Schirmer et al., 2015).

When sequencing an HIV quasispecies directly using the MiSeq platform, it is challenging to deconvolute the diversity inherent in the quasispecies from the sequencing error (Zagordi, Bhattacharya, Eriksson, & Beerenwinkel, 2011). Additionally, the bias introduced during the PCR amplification step required to generate enough DNA obscures the true relative abundance levels for different variants. The Primer ID (PID) approach was developed to address this problem.

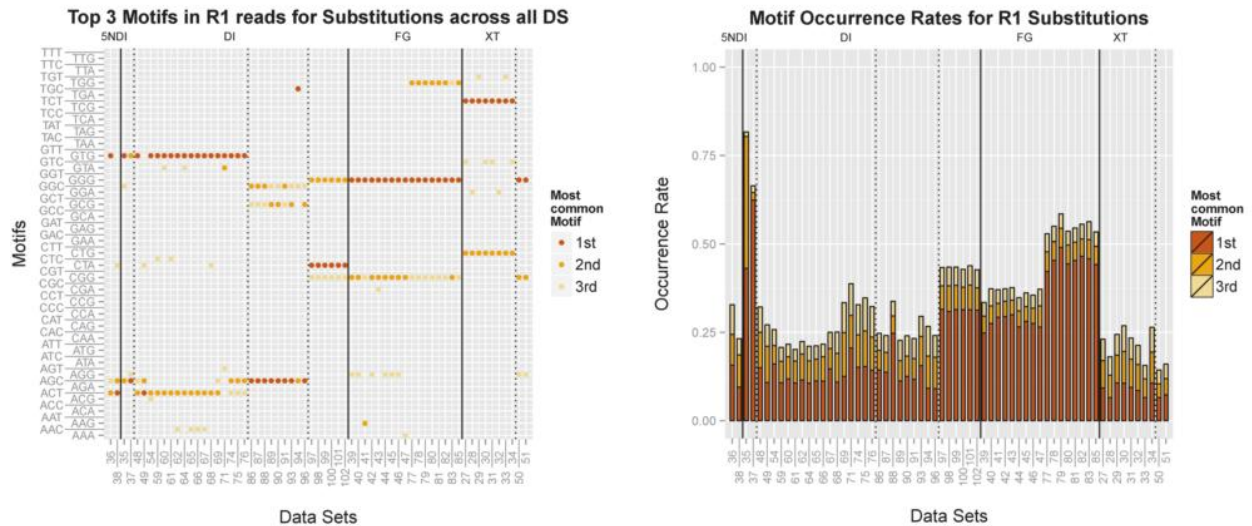


Figure 11: The most frequently occurring 3mers directly preceding substitution errors. The left side of the figure displays the three most common motifs for each data set and the right side illustrates the percentage of errors that were associated with the respective motif. The lower x-axis indicates the number of the dataset used, the upper x-axis indicates the library preparation method grouped by the solid vertical black lines. The dotted vertical black lines blocks off datasets that utilized the same primers. Figure from (Schirmer et al., 2015).

2.5 The PID approach

In 2011 and 2012 numerous publications proposed tagging DNA molecules with unique identifiers to address the issue of PCR induced bias (Casbon, Osborne, Brenner, & Lichtenstein, 2011; Fu, Hu, Wang, & Fodor, 2011; Kivioja et al., 2012) or both the bias and inaccuracy of NGS (Jabara et al., 2011; Kinde et al., 2011; Shiroguchi, Jia, Sims, & Xie, 2012). The PID approach allows highly accurate sequencing of a significant proportion of individuals from a population. Most PID approaches focus on sequencing a short amplicon, but (Hong et al., 2014) proposed a modification potentially allowing full length genome sequencing.

As illustrated in Figure 12, the first step when sequencing HIV RNA is to reverse transcribe the RNA into cDNA. When the primers for the reverse transcription step are synthesized, random letters of a set length are added to each primer so that each primer contains the region that binds to the RNA,

together with a random sequence that is unique¹ to that individual primer molecule. These primers together with reverse transcriptase are added to the RNA causing a reaction that yields double stranded molecules in which one strand is RNA and the other cDNA.

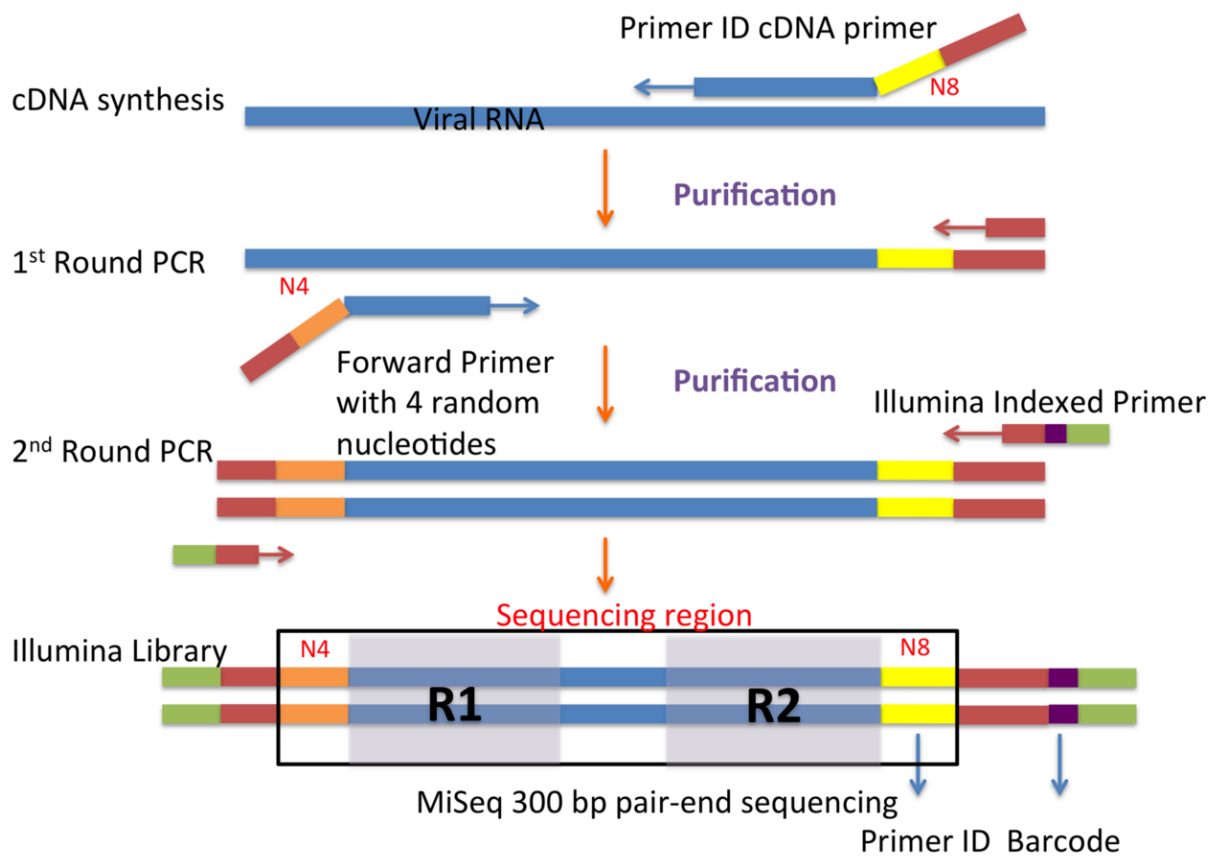


Figure 12: PID approach for the MiSeq Platform used by (S. Zhou et al., 2015). The PID (yellow, N8) is included in the cDNA primer along with a PCR primer site (brown) and the upstream primer includes four randomized bases to add diversity to the initial sequencing read (required for calibration; orange N4). The platform specific primers (green with purple barcode) are included in the last round of PCR. Paired-end sequence of region 1 (R1) and region 2 (R2) are indicated which may or may not overlap in the middle. Figure from (S. Zhou et al., 2015).

The RNA is removed from the sample in a purification step before PCR is used to amplify the cDNA. When the cDNA is amplified, the unique labels are copied together with the rest of the sequence of interest. Thus after PCR, each input copy of cDNA that was successfully amplified yields a set of identical molecules, except for PCR induced errors, with the same PID.

The amplified DNA is sequenced using next generation sequencing technologies like Illumina’s MiSeq platform. The reads obtained from the machine will contain a PID, which can then be used to group together all reads that were generated from the same input cDNA molecule. By comparing these reads

¹ This is not strictly true due to the large number of molecules and a limited number of possible combinations, see section 2.6.2 for more details.

to each other, any differences can be attributed to either sequencing error or an error made by the polymerase used during PCR. A consensus sequence is constructed for each bin in the final dataset by first aligning all the sequences with the same PID and then constructing a sequence by letting each position be the base that occurs most frequently at that position in the alignment.

The construction of the consensus sequences reduces the PCR bias and the sequencing error significantly. The PCR bias differs between samples and laboratories. Jabara and colleagues generated 2,213 consensus sequences from 27,075 reads (Table 2) and no PID was sequenced more than 96 times (Figure 13) (Jabara et al., 2011). In contrast, a team from Sweden and Germany generated 14 consensus sequences from 47,387 reads (Table 3) with the most frequently occurring PID occurring over 9,000 times (Figure 14) (Brodin et al., 2015). By working with the sequences grouped on their PIDs instead of the reads from the machine, the bias induced by PCR amplification is removed. Researchers based at the university of North Carolina measured the substitution error rate after applying the PID approach to be between 0.011% and 0.002% (Table 4) (S. Zhou et al., 2015) which is less than the 0.5% reported in (Schirmer et al., 2015) for the Illumina MiSeq platform.

Table 2: Number of reads and consensus sequences obtained by (Jabara et al., 2011) for three datasets.

Sample	T1	T2	T3
Ritonavir	-	-	+
Total reads	20,429	24,658	27,075
Consensus sequences	857	1,609	2,213

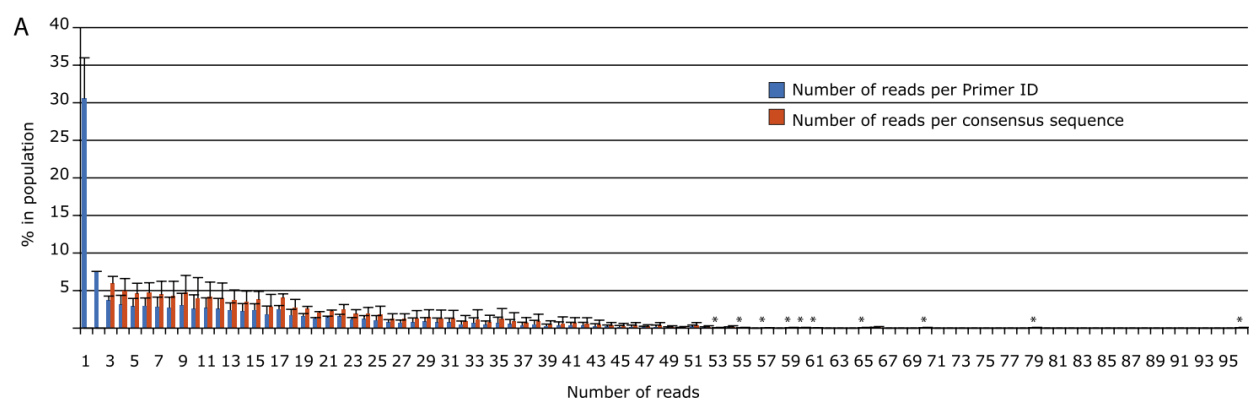
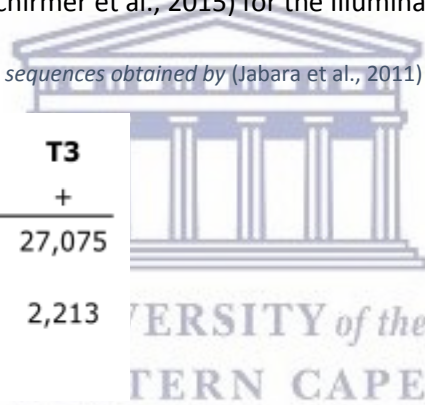


Figure 13: Distribution of the number of reads per PID (blue) or consensus sequence (red). Three datasets were used in the construction of this figure, the height of the bar indicates the average value over the three datasets and the error bars the standard deviation. Stars indicates cases where a single sequences was resampled a high number of times. Figure from (Jabara et al., 2011).

Table 3: Number of reads and consensus sequences obtained by (Brodin et al., 2015) for four datasets.

Sample	No. of input template molecules	No. of reads	No. of reads with PIDs observed at least 3 times	No. of consensus template sequences		
				Uncorrected	Corrected for PID substitutions	Corrected for PID substitutions and indels
Clone	10,000	47,387	47,225	97	23	14
Patient A	18,900	104,597	102,192	2,103	2,000	1,786
Patient B	24,000	57,159	56,317	263	200	184
Patient C	5,850	20,089	19,816	120	103	99

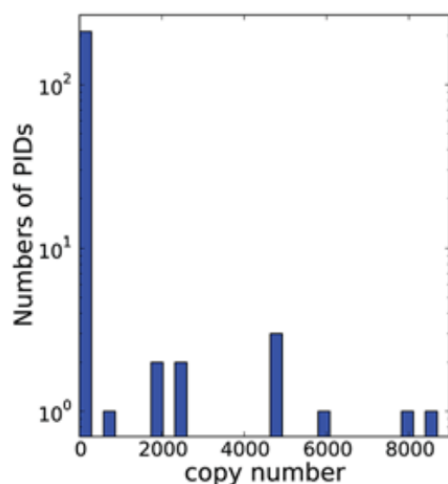


Figure 14: Distribution of the number of reads (copy number) per PID for one sample (the "Clone" sample from Table 3). Figure from (Brodin et al., 2015).

Table 4: Summary of the error rates after application of the PID approach from (S. Zhou et al., 2015). The different sets were produced using different enzymes.

	ENV (Set1)*		ENV (Set2)		ENV (Set3)		Protease (Set1)		
	V1/V2	C2/V3	V1/V2	C2/V3	V1/V2	C2/V3	R1	R2	combined
Control									
Consensus sequences	23,385	23,385	18,408	18,408	15,205	15,205	14,778	14,778	14,741
Mis-priming	6	41	8	40	4	5	8	15	7
In-frame deletion	5	12	0	8	1	2	0	0	0
Frameshift	138	134	151	87	184	25	43	42	55
Consensus sequences (no in/del)	23,236	23,198	18,249	18,273	15,016	15,173	14,727	14,721	14,679
Length	265	256	265	256	265	256	265	256	340
Substitutions	206	748	73	412	158	311	426	488	565
Substitution rate	0.003%	0.013%	0.002%	0.009%	0.004%	0.008%	0.011%	0.013%	0.011%
Substitutions (excluding first 1 and last 2 positions)		0.009%		0.005%		0.008%			

2.6 Complications of Primer ID

A number of complications associated with the PID approach have been highlighted:

- 1) Skewed template resampling with a small number of input templates accounting for a large proportion of the total sequences (Brodin et al., 2015; Fu et al., 2014; Kou et al., 2016; Lundberg et al., 2013).
- 2) PID collisions leading to chimeric bins (Sheward, Murrell, & Williamson, 2012; Yourstone, Lundberg, Dangl, & Jones, 2014; S. Zhou et al., 2015).
- 3) Sequencing or PCR error in the PID region (Kou et al., 2016; Lundberg et al., 2013; S. Zhou et al., 2015).
- 4) Sequencing and PCR error after applying the PID technique (Brodin et al., 2015; Jabara et al., 2011; Kinde et al., 2011; Kou et al., 2016; S. Zhou et al., 2015).

Each of these issues are explored in this section.

2.6.1 Skewed template resampling

As noted at the end of section 2.5, the ability of the PID approach to correct for PCR induced bias is one of its main features. However, in cases where the bias is severe the final number of consensus sequences produced might be very small. As shown in Table 3, as few as 14 final consensus sequences can be obtained from 47225 input reads. Skewed template resampling leading to a low number of consensus sequences was also reported by (Fu et al., 2014) and (Kou et al., 2016) see Table 5 and Table 6 respectively.

Table 5: Number of consensus sequences and raw reads for different samples from (Fu et al., 2014). The column titled "Unique transcripts" lists the number of consensus sequences obtained for each sample from the number of raw reads given by the "Paired-end reads" column.

RNA	Length	Copies of input RNA in library	Paired-end reads	Unique start/stop sites*	Unique transcripts [†]	Yield [‡]
ERCC130	1,059	9,000,000	1,059,847	11,206	41,331	0.0046
ERCC136	1,033	562,500	310,315	2,255	3,579	0.0064
ERCC108	1,022	281,250	76,479	1,198	1,603	0.0057
ERCC116	1,991	140,625	40,592	157	181	0.0013
ERCC092	1,124	70,314	36,347	263	308	0.0044
ERCC095	521	35,157	5,565	41	42	0.0012
ERCC019	644	8,790	3,080	15	17	0.0019

*The number of sequenced clones of different start/stop sites and overlapping by at least a single nucleotide.

[†]The number of sequenced clones of different start/stop sites with distinct molecular indexing and overlapping by at least a single nucleotide.

[‡]The ratio of resulting transcripts in the library to the total number of copies added to the sample used for library preparation.

Table 6: Number of consensus sequences and raw reads for different samples from (Kou et al., 2016). Five different samples were prepared which was composed of a majority variant (99% or 99.9% of the sample) and a minority variant (1.0% or 0.1% of the sample). Each variant is reported on a separate row. The column titled "UID #" lists the number of consensus sequences obtained for each variant from the number of raw reads given by the "Read#" column.

Template	Input	Read#	Output (based on reads)	UID #	Output (based on UID)
FGFR3-E7 WT	99.9%	6669968	100%	177998	100%
FGFR3-E7 R248C(Chr4:1803564 G>A)	0.1%	1	0%	0	0%
FGFR3-E9 Y373C(chr4:1806099 T >C)	99.0%	70722	69.0%	271	96.4%
FGFR3-E9 WT	1.0%	19336	21.5%	10	3.6%
HRAS-E1 WT	99.0%	265986	97.9%	1259	97.7%
HRAS-E1 G13V(Chr11:534285G>T)	1.0%	5753	2.1%	29	2.3%
PIK3CA-E20 WT	99.9%	45767	81.7%	768	96.6%
PIK3CA-E20 H1047L(Chr3:178952085A>G)	0.1%	10231	18.3%	27	3.4%
PIK3CA-E9 WT	99.9%	13390	100%	211	100%
PIK3CA-E9 E542K(Chr3:178936083 C>T)	0.1%	65	0%	0	0%

A possible explanation for the skewed resampling is that some of the randomly generated primers has a higher probability of being amplified during the PCR steps. (S. Zhou et al., 2015) explored this hypothesis by applying the PID approach to the same sample twice independently of each other. The most frequent PIDs from the one sample were randomly distributed in the other sample suggesting that the PID itself is not the cause of the bias amplification.

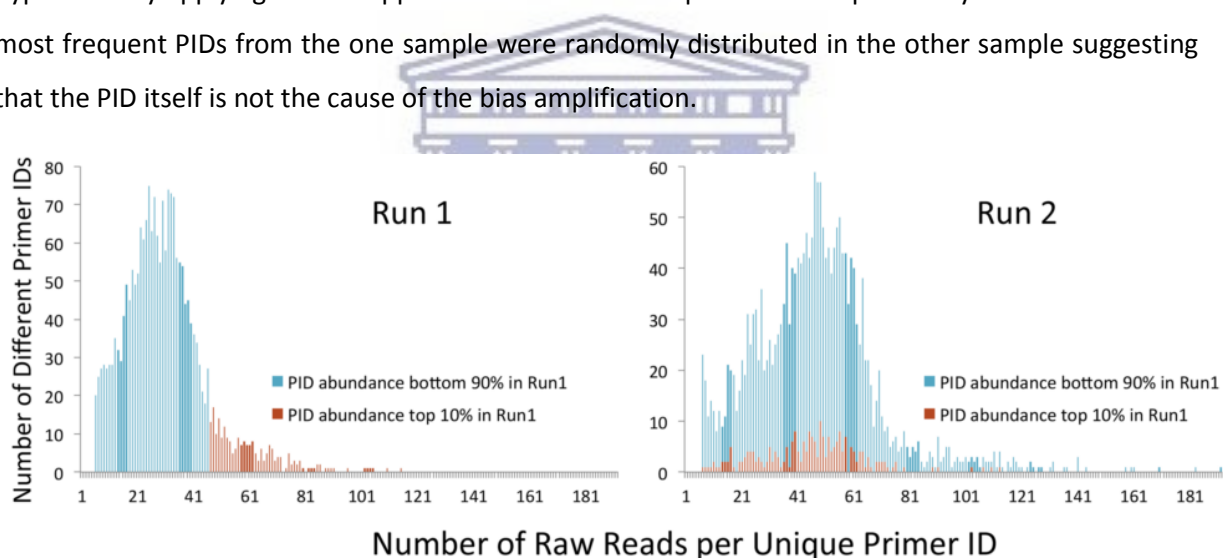


Figure 15: The number of reads per unique PID. The PIDs that occurred the most frequently (top 10%) in Run 1 is colored red and all other are colored blue. Figure from (S. Zhou et al., 2015).

2.6.2 PID Collisions

(Sheward et al., 2012) introduced the birthday paradox in the context of the PID approach. The birthday paradox states that in a room of only 23 people there is a larger than 50% chance that two people will share a birthday. If for example, the block of degenerate nucleotides is 8 bases long and 10,000 templates were reverse transcribed into cDNA molecules, then only 8585 of the templates are expected to have unique PIDs. However, the need to assign a unique PID to each template must be balanced with the probability of making sequencing and/or PCR errors in the PID region. The longer the PID, the larger the chance of making a sequencing error while sequencing the PID region (S. Zhou

et al., 2015). The relationship between the length of the PID and the probability of a sequencing error in the PID is illustrated in Figure 16.

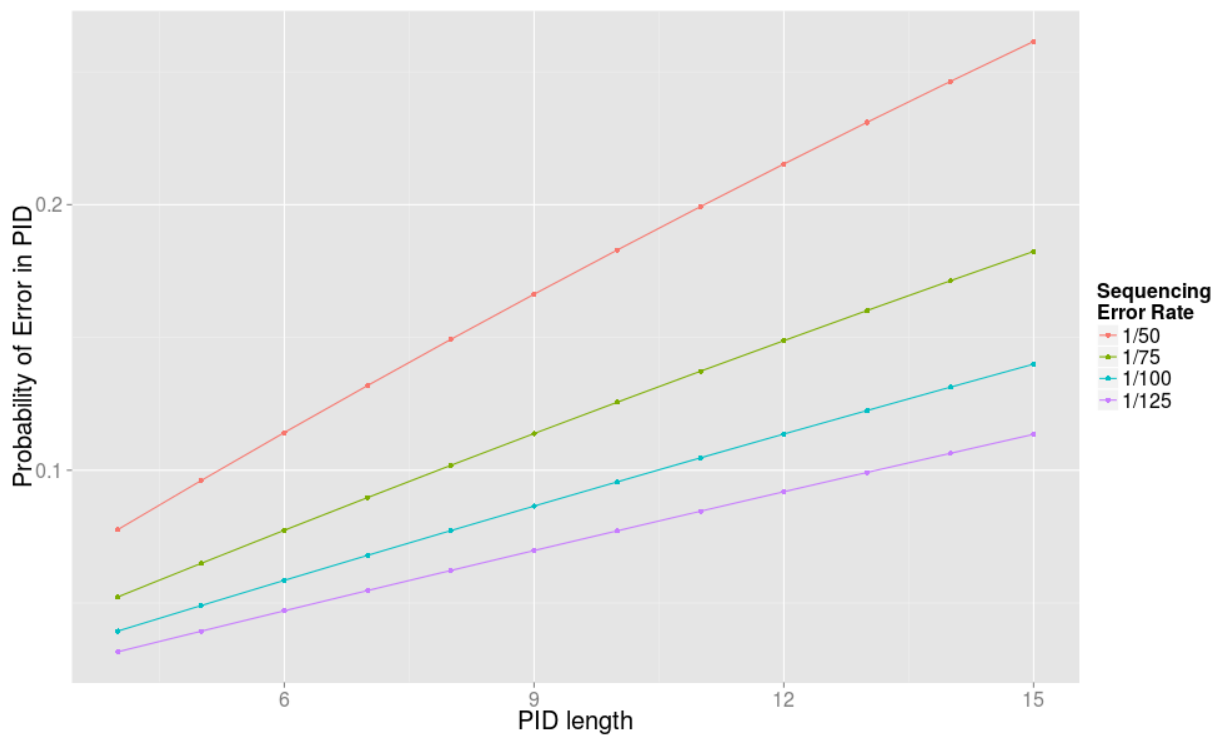


Figure 16: The relationship between the length of a PID and the probability of an error occurring while sequencing the PID.

When two different input templates are assigned the same PID, it is called a PID collision. A PID collision will lead to a chimeric bin. Chimeric bins can have one of the following compositions:

- 1) The sequence for the different input templates may be identical,
- 2) The one template may occur more frequently than the other template in the bin, or
- 3) Both templates may occur at similar frequencies and have different sequences.

The first two cases will not lead to complications in the generation of the consensus sequence, but the third case may lead to ambiguous nucleotide calls occurring in the consensus sequence at the positions where the two templates differ. However, the rate at which bins satisfying case 3 will occur should be very low since a number of events must occur in sequence:

- 1) Two input templates must be given the same PID,
- 2) The sequences of the templates must be different, and
- 3) Both of these templates must occur at very similar frequencies in the final dataset.

A measure of the chimerism in a bin was developed in (Yourstone et al., 2014). When different sequences with the same PID have different bases at the same position, this is due to either a sequencing or PCR error in one of the sequences or the two sequences were from different input templates. If the quality scores associated with these bases are high, then it is more likely that two

different input templates were tagged with the same PID than if the quality scores of either or both of these bases were low. Based on this (Yourstone et al., 2014) defined the consensus score (c-score). Each column in the multiple sequence alignment (MSA) is first assigned a score by multiplying the mode base percentage (i.e. mode base count / total base count) by the mode base average quality score. Subsequently, these column scores are averaged across all columns to generate the c-score.

2.6.3 Sequencing error with in the PID region

Sequencing errors in the PID are very frequent. Consider an example where the PID is 8 nucleotides long, the sequencing error rate is 1 in 100 and there is a bin of size 500. In order to sequence all the nucleotides in the PID region of this bin, 4000 nucleotides need to be sequenced. Under the assumed error rate, around 40 sequencing errors is expected within the PID. This means that this one bin of size 500 will generate around 40 offspring bins each of a very small size. The exact same sequencing error may happen in more than one of the offspring bins, yielding offspring bins with more than one template. Sequencing error in the PID region was first mentioned in (Lundberg et al., 2013). They noted that there was an unexpectedly high number of singletons (PID sequenced only once) in the dataset and that the average sequence quality of singletons were lower than that of the rest of the data. MT-Toolbox (Yourstone et al., 2014), the software used in (Lundberg et al., 2013), allows the exclusion of bins of size one, but does not further address the issue.

(S. Zhou et al., 2015) performed a detailed investigation of the sequencing error in the PID region. They used a dataset with a low input copy number (370 input templates). 11,208 unique PIDs were present in this dataset of which 8,121 occurred only once. First, they showed that the sequencing quality scores in PID region of those sequences belonging to bins with only one sequence is lower than for bins with more than 53 sequences (Figure 17 a). Next they compared the PIDs in the bins with less than 23 sequences to the PIDs of the bins with more than 53 sequences and found that over 80% of bins sized 4 to 22 have PIDs that differ by only one nucleotide from a PID found in a large bin (there were only 121 large bins in this sample) (Figure 17 b). An offspring bin is defined as a bin whose PID resulted from a sequencing error in the PID region.

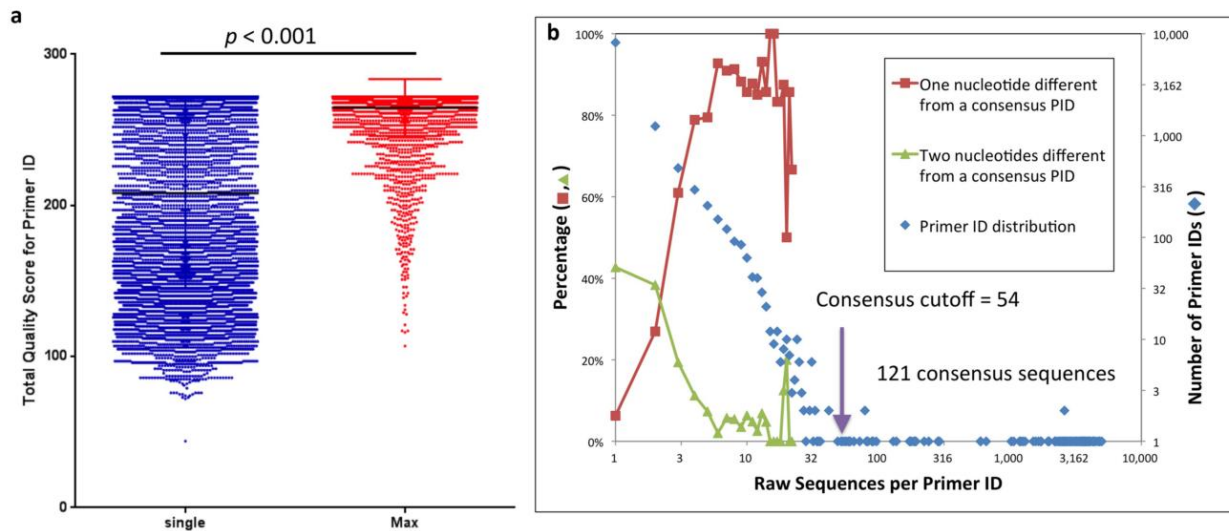


Figure 17: Assessment of sequencing errors in the PID region. The quality scores for the PID region in sequences with PIDs that occur only once in the dataset (singles) compared to the quality scored for more frequently occurring PIDs (more than 53 times) (a). The PID distribution and percentages of PIDs at low abundance (less than 23 occurrences) that differ by one or two nucleotides from the frequently occurring PIDs (more than 53 times). Figure from (S. Zhou et al., 2015).

To combat the problems resulting from sequencing errors in the PID, (S. Zhou et al., 2015) performed a simulation to compute a cutoff (named the consensus cutoff) so that all bins with less sequences than this cutoff are likely to be offspring bins. All offspring bins are discarded. The key assumption of their simulation is that the largest offspring bins will most likely result from sequencing errors in the largest bin in the dataset. Hence they simulated sequencing a PID as many times as the largest bin in the dataset. The most frequently occurring incorrectly sequenced PID determines the size of the largest offspring bin. By repeating the simulation a large number of times, a distribution of sizes for the largest offspring bin was built. The consensus cutoff was chosen as the mean of the distribution plus 1.96 times the standard deviation. To make this approach easier to implement, they simulated the expected size of the largest offspring bin for a large range of bin sizes and fitted a 6th order polynomial regression model through the data points. Hence one can use this formula to quickly determine the cutoff for any dataset with similar assumptions.

2.6.4 Sequencing and PCR errors in the final consensus sequences

(Brodin et al., 2015; Jabara et al., 2011; Kinde et al., 2011; Kou et al., 2016; S. Zhou et al., 2015) reported that the error rates after applying the PID approach are very low, but not zero. Different reasons for the residual errors have been proposed. Kinde and colleagues measured error rate per cycle of PCR of Phusion polymerase and obtained a figure (0.00000045 errors/bp/PCR cycle) similar to that claimed by the manufacturer (0.00000044 errors/bp/PCR cycle) (Kinde et al., 2011). However they were not able to differentiate between sequencing error that still exists after applying the PID

approach and the error rate of the polymerase. The error rate per cycle of 0.00000045 errors/bp/PCR cycle reported in (Kinde et al., 2011) translates to an error rate of 0.00099% meaning that in the final sequences each base has a 1 in 101,010 chance of being incorrect. This is lower than the rates reported in other publications, but is explained by the stringent criteria used in (Kinde et al., 2011) to construct consensus sequence which required that 95% of sequences in a bin must be the same at each position for a consensus sequence to be constructed.

Zhou and colleagues measured the error rates in a number of different samples (Table 4) and reported the approximate error rate 0.01% (S. Zhou et al., 2015). They noted that this rate is close to the reported error rate for reverse transcriptase in an enzyme reaction.

Brodin and colleagues presented a scenario where the targeted region contained a hotspot for sequencing error (Brodin et al., 2015). It is a homopolymer stretch which presents a significant challenge for the 454 sequencing technology they used (Shao et al., 2013). Even though they achieved very deep coverage in a bin of this position (8,222 reads), the homopolymer error was still present in the final consensus sequence. A single substitution error is also shown where a T was misread as a C in a bin with 4,748 sequences.



Primer ID	No. of reads	Sequence	91	101	111	121	131	141	151	161
1	TTGGTACACC 8846	AAAAATAGAG	GA	AACTGAGAC	AACATCTGTT	AAGGTGGGGA	TTTACCACAC	CAGACAAAAA	ACATCAGAAA	GAACCTC
2	TAGACTTCTC 6086	-----	-----	-----	-----	-----	-----	-----	-----	-----
3	CCTAAGCAGC 4978	-----	-----	-----	-----	-----	-----	-----	-----	-----
4	TCCACTGTAG 4692	-----	-----	-----	-----	-----	-----	-----	-----	-----
5	AGATGGCCTG 2029	-----	-----	-----	-----	-----	-----	-----	-----	-----
6	GAATTATCCC 771	-----	-----	-----	-----	-----	-----	-----	-----	-----
7	TAAGCGAAAG 3	-----	-----	-----	-----	-----	-----	-----	-----	-----
8	TAAGCGATTG 8222	-----	-----	-----	-----	-----	-----	-----	-----	-----
9	CCGAGAGTGT 2625	-----	-----	-----	-----	-----	-----	-----	-----	-----
10	TTTTGCTAGG 3177	-----	-----	-----	-----	-----	-----	-----	-----	-----
11	TTTTGCTAGG 2460	-----	-----	-----	-----	-----	-----	-----	-----	-----
12	CACTGCTATT 1851	-----	-----	-----	-----	-----	-----	-----	-----	-----
13	TTACATTAAG 40	-----	-----	-----	-----	-----	-----	-----	-----	-----
14	CTCGGCCTGG 4748	-----	-----	-----	-----	-----	-----	-----	-----	-----

Figure 18: Errors in consensus sequences. An alignment of 14 partial template consensus sequences from (Brodin et al., 2015). Errors compared to the correct sequence are highlighted. Figure from (Brodin et al., 2015).

A detailed investigation of the error rates of the PID approach was presented in (Kou et al., 2016). They produced figures exploring the per position error rates in their datasets and measured early stage PCR errors by flagging bins in which more than 95% of the sequences differed from the know input template as errors that occurred during the first two cycles of PCR (Figure 19). The average error rate of stage 2 PCR (25 cycles) and Illumina sequencing ranged from 0.17% to 0.28% when using a less accurate polymerase (Platinum Taq) and 0.02% to 0.006% when using an accurate polymerase (Q5 with Platinum Taq). The average error rate of stage 1 PCR (2 cycles) was about one order of magnitude lower when using Platinum Taq and ranged from 0.04% to 0.05% (Table 7). The difference between

stage 1 PCR error and stage 2 PCR error combined with sequencing error was not as pronounced when using Q5 with Platinum Taq (.003% to .009%) (Table 7).

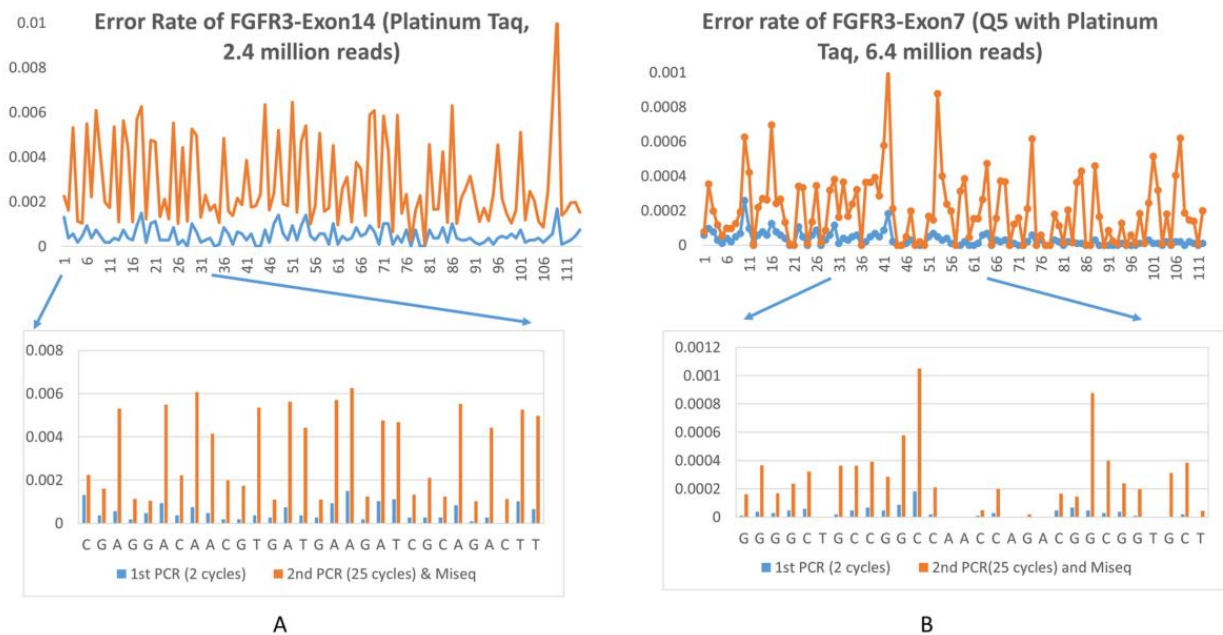


Figure 19: Error rates at each nucleotide in two datasets from (Kou et al., 2016). (A) Error rates plotted for all 114 nucleotides of the FGFR3-Exon14 sample which was amplified with Platinum Taq, nucleotides 1 to 30 are magnified. (B) Error rates plotted for all 112 nucleotides of the FGFR3-Exon7 sample which was amplified with Q5 enzyme, nucleotides 31 to 60 are magnified. Figure from (Kou et al., 2016).

Table 7: Average and standard deviation of the PCR and sequencing error rate on three of the samples from (Kou et al., 2016). The 2nd PCR rows include error from both the 2nd round of PCR and the Illumina sequencing step. Table from (Kou et al., 2016).

	Platinum Taq		Q5 with Platinum Taq		
	FGFR3-E14	FGFR3-E9	FGFR3-E14	FGFR3-E9	FGFR-E7
1st PCR Average	4.83E-04	4.14E-04	9.40E-05	3.39E-05	3.02E-05
2nd PCR Average	2.85E-03	1.68E-03	9.08E-05	6.22E-05	2.01E-04
1st PCR Stdev	3.61E-04	1.31E-03	5.34E-04	2.86E-04	3.97E-05
2nd PCR Stdev	1.92E-03	3.83E-03	4.80E-04	3.29E-04	1.99E-04
Read#	2400000	90059	62400	54638	6440196
UID#	4910	281	997	1589	104094

2.7 Other PID processing toolkits

The extra data processing steps introduced by the PID approach are non-trivial and MotifBinner aims to address them. During the course of this project, two solutions to the data processing steps were released. The first solution by (Yourstone et al., 2014) was used in the field of metagenomics and is a suite of perl modules and scripts with tests and a graphical user interface (GUI). The second

solution was used to detect drug resistance in HIV (S. Zhou et al., 2015) and is distributed as a set of ruby scripts that the user should edit and run.

2.7.1 MT-Toolbox

The Molecular Tag Toolbox (MT-Toolbox) is a suite of perl modules and scripts that processes data produced with a PID approach (Yourstone et al., 2014). It is a high quality implementation build with perl's package management and testing tools. MT-Toolbox is the first step in a bigger software system called MT-MT-Toolbox which performs metagenomics analyses on a PID dataset.

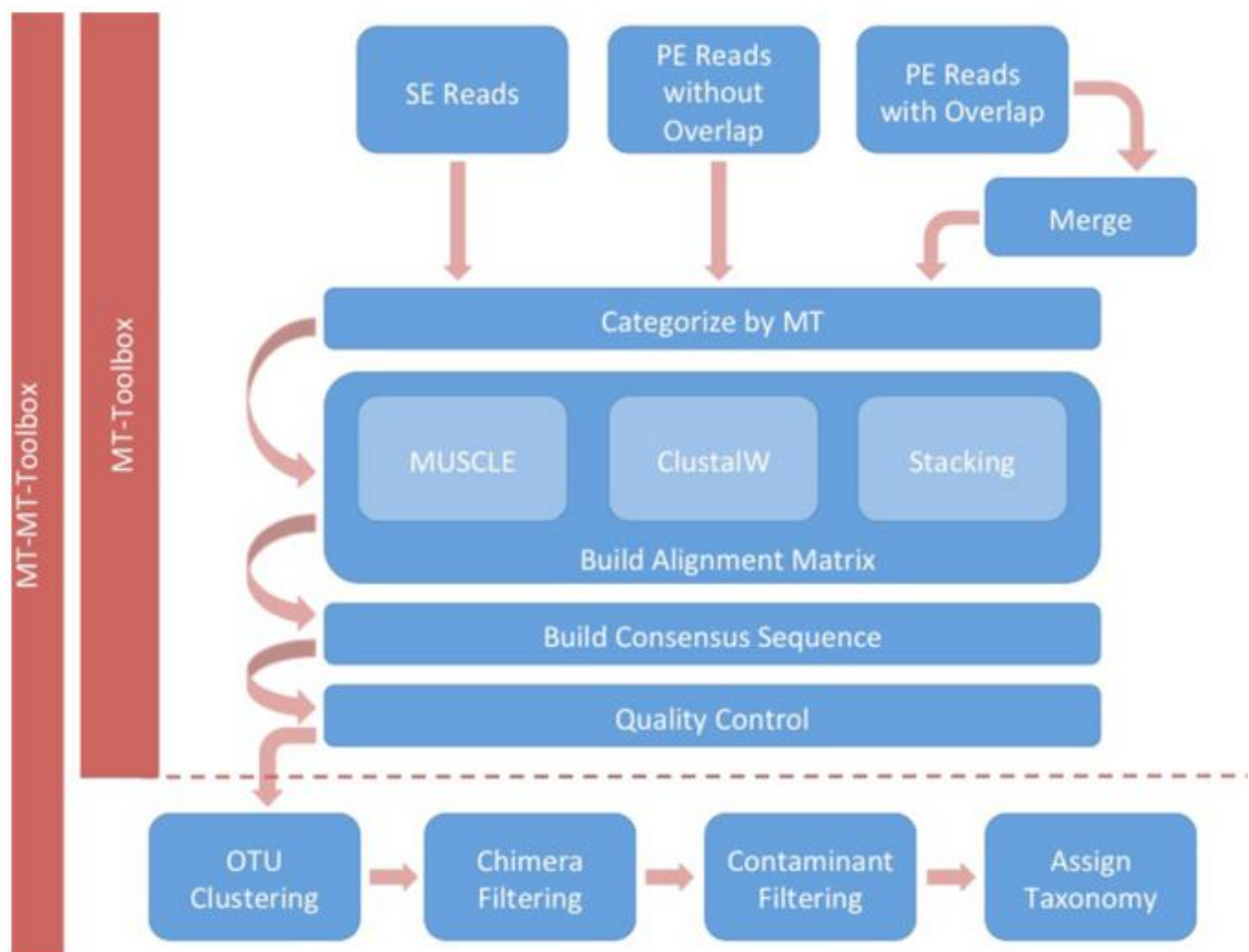


Figure 20: MT-Toolbox workflow. Single-end or paired-end (overlapping or non-overlapping) reads can be input into MT-Toolbox. Overlapping paired-end reads are merged after which all reads, regardless of their type, are categorized by their MT (molecular tag referred to as the PID in this work). Next a square alignment matrix is created for each MT (PID) category using either a multiple sequence alignment algorithm or by stacking the reads. From these matrices, consensus sequences are built and quality control measures remove low quality consensus sequences. The MT-MT-Toolbox extension performs additional metagenomics analyses. Figure from (Yourstone et al., 2014).

The workflow is described in Figure 20. A choice is offered for whether the sequences for each PID should be aligned or just stacked without alignment prior to consensus sequence construction. Due

to the low frequency of indels in Illumina reads, the stacking strategy yields acceptable results. A sample dataset described in the publication contained 449,676 reads of which 447,175 (99.45%) were 253 bases long.

When constructing the consensus sequences for each PID, ties are broken using the quality scores. For example, if in a bin of size four, two sequences has an A at position 1 and two sequences a T, then the consensus will contain the base which was read with the highest average quality. If using the quality scores does not resolve the tie, then an IUPAC ambiguity character is inserted. No provision is made for sequencing errors in the PID beyond exclusion of bins of size 1.

Chimerism is measured using a custom metric they called the consensus score (c-score). It measures conflicts within the sequences with the same PID. If sequences with the same PID has different bases at the same position and those bases have very high quality scores, then it is a strong indication that the sequences are in fact from different input templates. To compute the c-score, each column in the MSA is first assigned a score by multiplying the mode base percentage (i.e. mode base count / total base count) by the mode base average quality score. Subsequently, these column scores are averaged across all columns to generate the c-score. The user may specify a threshold c-score and all PIDs whose associated c-scores are below this threshold are excluded from the final dataset.

MT-Toolbox allows for a complex primer design in which the lengths of the primers can be staggered to overcome problems in earlier versions of the Illumina MiSeq platform. This flexibility complicates the application of this tool to datasets produced with more basic primer designs.

2.7.2 Ruby scripts of (S. Zhou et al., 2015)

The (S. Zhou et al., 2015) publication was accompanied by a set of ruby scripts. Two of these scripts are of interest when processing PID data. The one script computes the consensus cutoff which is an input to the script that processes the data. The data processing script can be run from the command line and accepts the length of the PID and the input file name as arguments. The script must be edited to set the sequences for the flanking the PID which will allow the extraction of the PID for each sequence.

No alignment step is performed, taking advantage of the very low frequency of indels in the Illumina sequencing process. Using a simulation process, a cutoff is computed and all bins smaller than this cutoff are discarded due to the possibility that they are the result of sequencing error inside the PID region itself.

3 MotifBinner

As part of a project to develop and optimize protocols for sample preparation and sequencing of HIV samples we developed a software package called MotifBinner. It processes the raw reads producing consensus sequences and various metrics that aid in troubleshooting samples that did not produce usable data. The first part of this chapter details the design and implementation of the software package. The parameters and algorithms are explained here. The formats in which output are produced are described in this first section. The second section of this chapter presents the simulations and investigations that were used to justify the design decisions. The chapter closes by carefully examining the results of processing two datasets with MotifBinner.

3.1 Design and Implementation

MotifBinner is designed to process high-throughput/next generation sequencing (NGS) data generated using the Primer ID (PID) approach, as described in (Jabara et al., 2011). In this method, a random sequence tag is included in the initial cDNA synthesis primer, such that each input template is tagged with a unique block of nucleotides (PID). Following amplification with PCR, all sequences amplified from the same original cDNA template will theoretically have the same PID. MotifBinner utilizes the following steps to process NGS data that contains PIDs:

1. Locate and identify the PIDs in each sequence
2. Group sequences into bins based on their PIDs
3. Based on a model of sequencing error rate, discard bins which have a high probability of originating from sequencing error within the PID
4. Determine whether a bin contains non-uniquely labelled sequences (PID collisions).
5. Filter out sequences from each bin that are the result of PID collisions.
6. Align the sequences in each bin
7. Construct a consensus sequence for each bin
8. Produce a report detailing the steps that were taken
9. Save all the results and the report

3.1.1 Locate and identify the PIDs in each sequence

According to the PID method, the primer used for cDNA synthesis consists of a known, gene-specific binding region, a sequencing platform specific region and a randomly assigned PID of a known length. A common design, shown in Figure 21, is to have the primer ID between two known regions in the primer. A region in the primer that comes before (upstream) the primer ID is referred to here as a

'prefix'. Likewise, a region in the primer that comes after (downstream) the primer ID is referred to as the 'suffix'.

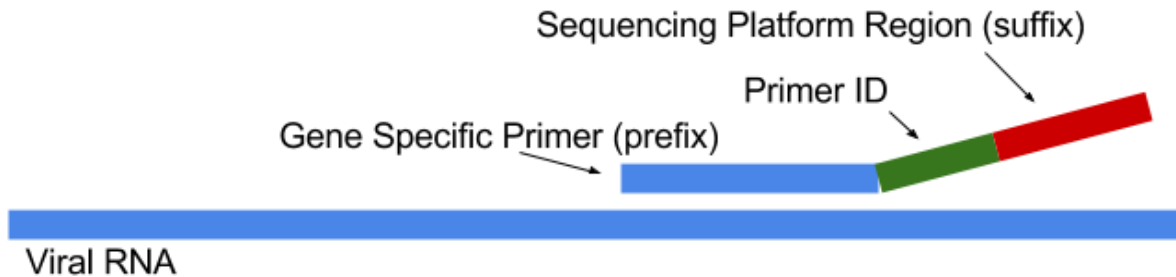


Figure 21: cDNA synthesis primer design assumed for use with MotifBinner.

A search pattern is constructed by concatenating the prefix sequence, a series of 'N's, and the suffix sequence. The 'N's represent a match to any base and the number of 'N's inserted is equal to the length of the PID. Either the prefix or suffix may be NULL, allowing researchers flexibility in their primer design. Each sequence is then searched for this pattern using the `vmatchPattern` function from the Biostrings (Pages, Aboyou, Gentleman, & DebRoy, 2017) package. The number of mismatches to the search pattern are relaxed iteratively to account for sequencing error within the prefix and suffix, while ensuring that each match is obtained at the highest specificity. Since an 'N' will match any base, mismatches are only allowed in the prefix and suffix and not in the PID itself. Figure 22 shows an example of the search for the PIDs in the sequence data. This process is repeated until the maximum number of allowed mismatches is reached (controlled via the `max.mismatch` parameter).

	Prefix								Primer ID								Suffix					Mismatch					
Primer Design	C	A	C	A	T	G	G	A	A	T	N	N	N	N	N	N	N	N	N	N	N	C	T	G	A	G	-
Sequence 1	C	A	C	A	T	G	G	A	A	T	A	A	T	T	T	T	T	A	G	C	T	G	A	G	0		
Sequence 2	C	A	G	A	T	G	G	A	A	T	A	A	T	A	A	G	G	A	A	C	T	G	A	G	1		
Sequence 3	C	A	C	A	T	G	G	T	A	T	A	C	C	A	T	G	T	A	C	C	T	G	A	-	2		
Sequence 4	C	A	C	T	T	G	C	A	A	T	A	C	C	A	T	G	T	A	C	G	T	G	A	G	3		

Figure 22: An example of an alignment of the primer region in sequence data. The first row shows the design of the primer which is the pattern used to search for PIDs in the sequences. A number of example sequences from raw data is included. The last column shows how many mismatches must be allowed in the search before the search pattern will be found in the sequence.

The current approach does not allow for insertions or deletions (indels) in the search pattern. If a sequence contains an indel in the prefix, suffix or primer ID it will not be returned as a match. A possible exception is that if the indel is very close to either of the terminal regions of the search motif,

then a match might be returned, depending on the number of additional mismatches present. For example, if there was an insertion at the third position on the left of the prefix, such that the sequence is AACCGT instead of AACGT, a match will be returned if the `max.mismatch` parameter was ≥ 3 .

The motif scanning procedure is implemented in parallel, using the `foreach` (Revolution Analytics & Weston, 2015) library. The number of CPU cores to use is specified by the `ncpu` parameter. A further processing time optimization allows the motif searching to start with a given number of mismatches allowed, as specified with the `max.mismatch_start` parameter.

The raw sequence names are replaced in the output by an optional user input string, concatenated to the PID that was found for the sequence. Additionally, a `data.frame` linking the new sequence name, with the original raw sequence name is produced. This `data.frame` is especially useful when processing paired-end data, allowing both pairs to be given the same PID, even though the PID was only contained in one of the two reads.

Sequences in which no primer ID could be located are removed from the primary dataset. These sequences are written to a file with the suffix `'_pid_not_found.fasta'` when the results of the binning process are written to disk.

3.1.2 Group sequences into bins based on their PIDs

The sequence data set is broken up into a series of smaller data sets. Each of these data sets is called a bin, where each bin will only contain sequences that have the same primer ID. In this way, each bin will theoretically contain sequences that are from a single input template.

3.1.3 Discard bins based on invalid PIDs

An inherent problem with current NGS technologies is their high error rates. Thus, there is the possibility that sequencing errors may occur within the PID. In order to combat this problem, `MotifBinner` implements an approach proposed by Zhou and colleagues (S. Zhou et al., 2015) based on estimating a cutoff and discarding all sequences where the bins contain fewer sequences than this cutoff. This is due to the fact that these small bins are likely to be 'offspring bins', bins that arise as a result of sequencing errors within the PID as opposed to true bins containing a uniquely generated PID. Assuming an error rate, specified via the `sequencing_error_rate` parameter, one can estimate the size of the offspring bins produced from the largest bin. This number of sequences in the largest expected offspring bin is then used as the cutoff for the minimum bin size allowed. The simulation approach implemented in `MotifBinner` is described in the following sections.

3.1.3.1 *Simulating the sequencing of Primer IDs*

A function called `sim_one_parent_main_off` was written to perform the simulation of the sequencing for a single bin. For a given bin size, primer ID length and sequencing error rate, this function will simulate sequencing a PID the number of times specified by the bin size argument under the assumed sequencing error rate. A number of PIDs will then be produced, of which the majority will be the original PID sequence (called the parent PID). All other PIDs produced (those with sequencing errors in them) will be referred to as offspring PIDs. The PID that occurs at the second highest frequency (the most frequent of the offspring PIDs), will be referred to as the main offspring PID. The function returns the number of parent PIDs produced and the number of main offspring PIDs.

If this process is repeated a large number of times, a distribution can be built showing the relationship between the number of parent PIDs produced and the number of main offspring PIDs produced. This distribution can be used to guide the choice for the consensus cutoff associated with a parent bin of a specified size. Three parameters influence the size of the main offspring bin and a range was considered for each of them, namely:

- The true size of the bin (which would result if PIDs were sequenced with perfect accuracy), ranging from 100 to 15000 in steps of 20
- The length of the primer ID, ranging from 5 to 15
- The sequence error rate for which four values are considered, 1/50, 1/75, 1/100 and 1/125.

3.1.3.2 *Computing a consensus cutoff from simulated data*

Once you have a distribution of main offspring bin sizes simulated for a given parent bin size, the question remains as to how to choose a cutoff value from that distribution. The mean value is not a good choice, since roughly 50% of the offspring bins would still be passed into the final dataset. The maximum offspring bin size is not a valid choice either as it is effectively determined by the number of simulations you perform (since the chance exists that all of the reads for a given PID contains a sequencing error in the PID). We chose to use the 99th percentile of the distribution.

To accurately estimate the 99th percentile of a distribution a large sample size is required. It is impractical to simulate a large sample for each possible parent bin size, therefore we used a sliding window of size 20 over the parent bin size. All observations within the sliding window are pooled and a 99th percentile was computed for them. This 99th percentile was then taken as the size of the main offspring bin for a parent bin whose size is equal to the midpoint of the sliding window. To simulate the data, the true bin size is incremented from 100 to 15000 in steps of 20 and 2000 observations are

simulated for each true bin size. A sliding window yields acceptable results since the relationship between parent bin size and the main offspring bin size is very linear over the width of the sliding window.

3.1.3.3 Modeling the relationship between parent and main offspring bin sizes

The relationship between the parent bin and main offspring bin sizes behaves like a power function with an exponent smaller than one for small parent bin sizes and like a straight line for larger parent bin sizes. Hence the relationship can be summarized with the following model: $y = a + bx + dx^e + \varepsilon$ where y denotes the size of the main offspring bin and x the size of the parent bin. The `nls2` (Grothendieck, 2013) package was used to fit this model to the data by assuming that error terms, ε , are normally distributed.

To ensure reliable fitting of the models, a linear model regressing the main offspring bin size on the parent bin size as a straight line is first fitted. The residual sum of squares is computed from this simplistic model. Next the `nls2` (Grothendieck, 2013) package is used to fit 1000 versions of the nonlinear model each with small perturbations to the starting conditions. Of the 1000 fitted models, the model with the smallest residual sum of squares is selected. As a basic check the residual sum of squares is compared to the residual sum of squares of the basic linear model to ensure that the fitting procedure was successful. The starting conditions for the nonlinear models are perturbed with the following scheme:

- The a parameter is uniformly selected from -10 to 10
- The b parameter is uniformly selected from 0.001 to 0.01
- The d parameter is uniformly selected from 0.2 to 0.7
- The e parameter is uniformly selected from 0.2 to 0.6

3.1.3.4 Distribution and usage

The cutoff computed from the simulation will only be applicable to the dataset if the simulation parameters match those used in the biological design. Thus, a number of datasets were simulated and models were fit to these datasets. The coefficients of these models are packaged into `MotifBinner` so that the user can quickly obtain a consensus cutoff for the current dataset without the need to perform simulations.

After extracting PIDs from sequences and sorting the sequences into appropriate bins a consensus cutoff is computed by `MotifBinner` using these stored models; the largest bin that was found and the user supplied sequencing error rate. All bins smaller than the consensus cutoff are discarded and are not included in the subsequent steps.

3.1.4 Determine which bins are chimeric

While bins should theoretically only contain sequences from a single template, bins containing sequences from multiple templates have been observed, (Brodin et al., 2015; Yourstone et al., 2014; S. Zhou et al., 2015). Such bins are referred to as chimeric bins. This occurs when different input templates are tagged with an identical PID. PCR recombination causes a similar problem in which a portion of a sequence is labelled with the incorrect PID. The effect of chimeric bins on the overall quality of the data is small, but their presence indicates a potential flaw in the PID approach. Hence it is important to detect when a data set contains a large amount of chimeric bins.

If a bin contains n aligned sequences, each of length m , and we hypothesize that the bin was produced from a single input template, then the sequencing error for the bin can be approximated by dividing the total number of non-consensus letters in the alignment across all positions by the total number of letters in all sequences of the bin. We denote the estimated sequencing error by \hat{p} . Under the hypothesis of a single input template, the number of sequencing errors at each position follows a binomial distribution of size n and a success probability of \hat{p} .

For any position (i), the number of observed non-consensus letters ε_i can be used to construct a test for the null hypothesis that the given number of non-consensus bases are due to sequencing error only, against the alternative hypothesis that bases observed at the position is not the result of random sampling from a binomial distribution with a success probability of \hat{p} . The p-value for this test is given by $\theta_i = 1 - \text{binom}(\varepsilon_i, n, \hat{p})$ where binom denotes the cumulative distribution function (cdf) of a binomially distributed random variable. One can then specify a tolerance for type I error by choosing an α . The null hypothesis is rejected if $\theta_i > \alpha$. If the null hypothesis is rejected, then one may infer that the number of mutations at position i is unlikely to result from sequencing error with a single input template.

This computation must be done for the entire sequence. This means that m such hypothesis tests must be performed. Multiple hypothesis testing has well documented effects on the type I error of the test (R. G. J. Miller, 2012). To preserve the type I error, a correction must be performed. In *MotifBinner*, the Holm-Bonferroni correction is used (Holm 1979). After computing the p-values for all the positions and applying the correction to them, a bin is said to be chimeric if any of the corrected p-values exceed 0.999. Hence α is equal to 0.001.

Using a correction for multiple hypothesis testing frequently leads to severe power loss of the hypothesis test. However, for this application, the probability of a sequencing error is so low (when interpreted as a success probability in a binomial distribution), that the p-values returned from the binomial cdf gets extremely small for even a small number of non-consensus bases. Hence, the power

after applying the correction is still acceptable. Consider for example that $P(X \leq 10)$ for a $\text{binom}(30, \frac{1}{100})$ distributed random variable is equal to 0.999999999999995.

A limitation of the approach outlined above is that it only uses the information from the position with the largest number of mutations. Better performance is achieved by constructing a compound rejection criteria for the null hypothesis:

- Reject H_0 if the largest p-value exceeds 0.999, or
- Reject H_0 if the two largest p-values exceed 0.995.

While a statistically consistent derivation of the compound hypothesis is possible, it suffices to show that the sensitivity and specificity of the method based on a compound rejection criteria is high on simulated datasets.

3.1.5 Find and remove sequences that were given an incorrect PID

The previous section was concerned with evaluating an entire bin to see if the data in the bin resulted from multiple input templates with the same PID. In this section, each sequence in a bin is inspected to see if it is from the same input template as the rest of the sequences in the bin.

For each bin, a distance matrix is constructed that contains the distance of each sequence from every other sequence in the bin. We used the generalized Levenshtein edit distance that does not allow the transposition of adjacent characters (van der Loo, 2014). This is the most computationally intensive step in MotifBinner and scales quadratically with the size of the bins - i.e. it is an $O(n^2)$ operation. To reduce running time, a maximum bin size is specified using the `max_sequences` parameter. If a bin contains more sequences than specified with the `max_sequences` parameter, then a number of sequences equal to `max_sequences` (default is 400) is randomly sampled from the bin and all the other sequences in the bin are discarded before the distance matrix is computed. Large bins result from over amplification of sequences with certain PIDs during the PCR step and consists of many reads of the same input template. Discarding a portion of these sequences does not result in information loss provided that enough sequences remain in the bin.

Two different options can be used to compute the distances. The `stringDist` function of the `Biostrings` (Pages et al., 2017) library is the default. It is slower, but does not require the installation of an external dependency. Alternatively, the faster `vsearch` (<https://github.com/torognes/vsearch>) program can be used. `stringDist` returns a non-normalized edit distance. `vsearch` returns the edit distance normalized by the length of the pairwise alignment between the two sequences. The output from `vsearch` is altered to provide a raw edit distance

similar to that from stringDist. In a very small number of cases there are mismatches of size 1 between some distances in the results obtained from stringDist and vsearch due to rounding and loss of precision when converting the normalized distances from vsearch to non-normalized distances.

If the maximum distance between any two sequences in the dataset is greater than a threshold specified via the threshold parameter, the sequence(s) that are furthest away from all the other sequences is (are) removed from the dataset. In some cases, two or more sequences are equidistant from the other sequences, in which case both (or all) sequences are simultaneously removed. This process is repeated until the maximum distance between any two points drops below the threshold. The default threshold is $\frac{8}{600}$ meaning that if there are more than 8 errors in a read of length 600, then we expect that there is another process at play in addition to sequencing error and the read will be discarded as being from a different template than the rest of the sequences. It should be noted that the maximum distance between any two sequences is halved before comparing it to the threshold since a distance of 1 between two sequences mean that there was a sequencing error in only one of the sequences.

Sequences removed from the bins at this step are not considered in any of the subsequent steps of the binning process. These sequences are however reported in the output, where a fasta file is produced for each bin. The sequences removed in this step are indicated by a suffix '_out' appended to their sequence headers, while those that are retained are appended with the suffix "_src".

3.1.6 Align the sequences in each bin

After outlying sequences have been removed from the bins, alignment of the remaining sequences is trivial due to the high degree of similarity between the sequences. Thus, a muscle (Edgar, 2004) alignment with default settings was found to perform adequately.

3.1.7 Construct a consensus sequence for each bin

After alignment, a consensus sequence is constructed for each bin. For each position in the alignment, the letter that occurs most frequently is taken as the consensus letter. If more than one letter occurs with equal frequency, then an IUPAC ambiguity character is returned for that position.

3.1.8 Produce a report detailing the steps that were taken

A detailed report is produced recording various metrics about the binning process:

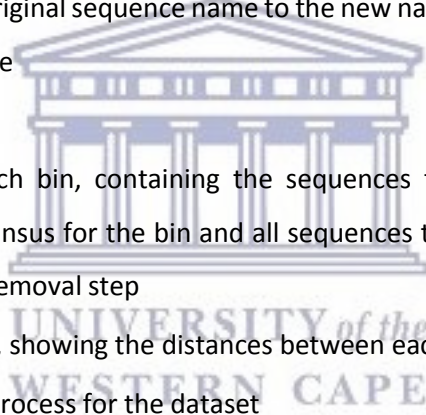
- Number of input sequences
- The number of sequence in which PIDs were found
- Sequence lengths

- The distribution of bin sizes
- The amount of sequences removed during the step that finds sequences with incorrect PIDs
- A list of the bins from which the most sequences were removed
- The number of degeneracies in the consensus sequences and the size of the bins from which the consensus sequences were produced
- An analysis of the distances between the sequences
- The running time of the different steps in MotifBinner
- The parameters that were used for the run

3.1.9 Save all the results and the report

A large amount of output is written to the disk, including:

- A file containing all the consensus sequences for the given data set
- A fasta file containing all the sequences in which PIDs could not be found
- A lookup table linking the original sequence name to the new name containing the primer ID that was found for that sequence
- A folder containing:
 - A fasta file for each bin, containing the sequences that were assigned to the bin, including the consensus for the bin and all sequences that were removed from the bin during the outlier removal step
 - A plot, for each bin, showing the distances between each of the sequences in that bin.
- The report on the binning process for the dataset



3.2 Materials and Methods **Datasets**

This work made use of two biologically derived NGS datasets produced using the PID approach described in (S. Zhou et al., 2015). These datasets have been published in (Bhiman et al., 2015) and a detailed description of the datasets can be found in that publication. Briefly, the datasets were derived from participant CAP256 of the CAPRISA 002 Acute Infection study, a cohort of 245 high-risk, HIV-negative women that was established in 2004 in Durban, South Africa, for follow-up and subsequent identification of HIV seroconversion. This individual is known to have been superinfected (re-infected with a distinct strain of HIV-1 at 15 weeks post infection.

The viral RNA was converted to cDNA using a primer designed to bind to HXB2 position 1094 to 1118 of the gp160 protein. It also included a randomly assigned 9-mer region (the PID) and a region to which the primers used for PCR will bind. Hence the cDNA synthesis process will produce cDNA molecules

that starts on the 3' end with the target region of the PCR primers, followed by the PID, and the sequence of the gp160 gene from position 1118 towards the start of the gp160 gene.

PCR was performed using a reverse primer binding to the region introduced during the cDNA synthesis step and a forward primer with a gene specific region targeted to HXB2 position 332 to 358 of the gp160 gene. PCR primers also included the required sequencing platform specific regions so that the final product after PCR consisted of positions 332 to 1118 of the gp160 gene and the relevant sequencing platform specific sequencing.

Paired-end sequencing was performed on the PCR product yielding two reads from each end of the molecule. The first read, called the forward read, starts from position 332 and extends 300 bases towards the 3' end (roughly HXB2 position 632 of the gp160 gene depending on indels in the specific variant). The second read, called the reverse read, starts on the 3' end, with the region introduced during cDNA synthesis – the binding site for the PCR primers and then the PID. The PID is followed on the 5' end by the sequence of gp160 from position 1118 to approximately position 842 (276 nucleotides).

Due to the highly variable nature of the 1st and 2nd variable loops (covered by the forward read), alignment of this region is challenging. Hence we only used the reverse reads from these datasets for the evaluation of MotifBinner. A small portion of the reverse reads covered positions 710 to 958 instead of 842 to 1118 of gp160, due to a miss-priming event. All reads resulting from the miss-priming event were removed from the dataset.

The first dataset was generated from a sample collected roughly 6 weeks post infection (visit code 2000), while the second dataset was generated from a sample collected at approximately 193 weeks post infection (visit code 4260). Hence, the two datasets are referred to here as '6wpi' and '193wpi' respectively. For the 6wpi dataset, all viruses were closely related to the primary infecting virus, while significant viral diversity had evolved by 193wpi (due in part to the superinfection event), see Figure 23 (Bhiman et al., 2015).

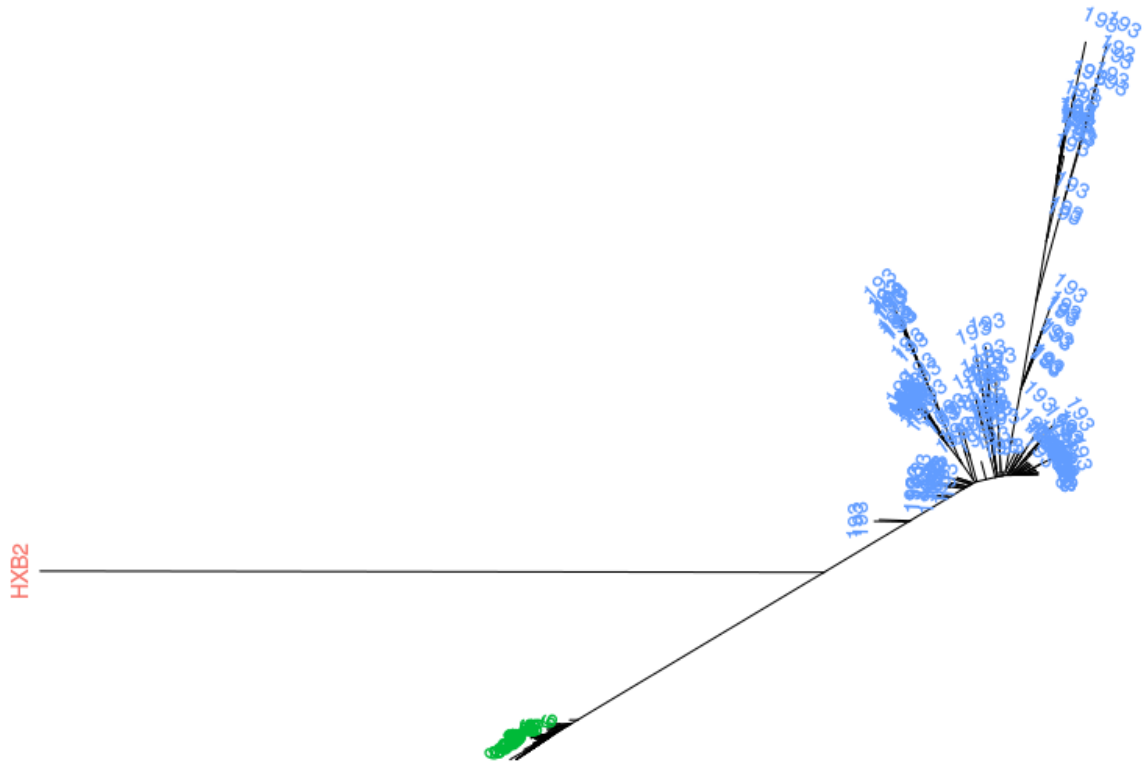


Figure 23: Differences in sequence diversity between the two datasets, illustrated by an unrooted phylogenetic tree. A maximum likelihood tree was constructed using *FastTree* (Price, Dehal, & Arkin, 2009) and plotted with *ggtree* (Yu G, Smith D, Zhu H, n.d.), using all the unique sequences from the 6wpi (green) and 193wpi (blue) datasets, as well as HXB2 (red) as a reference/outgroup.

Additionally, each dataset was processed using two different methods: The first method processed the data according to the PID approach, using the PID tags contained within the sequence, while the second method processed the data independently of these PID tags. The two versions of the processed datasets are referred to as the 'PID version' and the 'non-PID version'. The steps taken to produce these datasets are listed below:

- 1) The raw fastq files containing the reverse reads were selected.
- 2) The length and quality of the reads from each dataset were tabulated. Low quality reads from each dataset were filtered out using the *Shortread* package (Morgan et al., 2009), with a minimum average read quality score of 25 (Q25) and a minimum read length of 275.
- 3) Sequences resulting from miss-priming were removed by searching for a motif (ACCATGCAATAATGTCAGCACAGTACAA) that occurred only in the miss-primed sequences. The search was performed with the *vmatchPattern* function of the *Biostrings* (Pages et al., 2017) package, allowing for a total of 7 mismatches.

- 4) The sequences were generated with primers designed to stagger the sequencing start site. To ensure that all reads started at the same position in our datasets, bases to the 5' of the consensus start site (CAAACAATAATAGTACATCTCAATGAA) were removed from the reads in both datasets. This was achieved using the `vmatchPattern` and `padAndClip` functions of `Biostrings` (Pages et al., 2017), allowing up to 8 mismatches when searching for the start site in the sequence data.
- 5) Following these data clean-up steps, `MotifBinner` was used to further process the PID version of the two datasets into the resulting consensus sequences. Section 3.1 describes the process `MotifBinner` follows in detail. Briefly, the PID is located for each read and used to group reads with identical PIDs into bins. After a number of steps designed to assess and improve the quality of the bins, an alignment is constructed for each bin. The alignments are condensed into consensus sequences by representing each position in the alignment by the nucleotide that occurs most frequently at that position. The binning reports for these datasets are included in Section 8.1 and Section 8.2. The resulting consensus sequences were aligned in codon space using `MACSE` (Ranwez, Harispe, Delsuc, & Douzery, 2011) with default settings. The alignments were manually curated. In a `MACSE` alignment, incomplete codons are always padded on the left hand side with exclamation characters. In many cases, the alignment of the nucleotides can be improved by moving the gap padding to the right hand side of the incomplete codon. Additionally, the exclamation characters were converted to gap characters (dash) for consistency.

In order to assess the benefit of using the PID approach, the two datasets inputted into `MotifBinner` were processed according to the steps listed below, to create non-PID versions of the datasets:

- 1) Steps 1 to 4 above were followed, with the exception that in step 1, the PID motif (located at the 3' end of the reads) was also removed. This was achieved by stripping out the PID motif (CAGGAGGGGAYCTAGAARTTACAACNNNNNNNNNCTGAGCGTGTG), as well as any bases following the 3' end of this motif, using the `vmatchPattern` and `padAndClip` functions of `Biostrings`, allowing up to 5 mismatches.
- 2) In order to compare the PID versus non-PID methodologies, the sequences within each dataset had to be aligned consistently. As mentioned above, the PID datasets were aligned using `MACSE`. The non-PID datasets were profile aligned to the consensus sequences of the PID-dataset, using `MAFFT` (Katoh, Misawa, Kuma, & Miyata, 2002) with the `--add` option. This option adds unaligned sequences to an existing alignment and will not make any changes to the existing alignment other than adding gaps to it.

3) These alignments were also manually curated. MAFFT does not align sequences in codon space. Thus, the alignment was manually curated to obtain a more biologically relevant, 'in frame', alignment.

In this way, two versions of each dataset were generated (PID and non-PID) for use in the subsequent analysis, namely:

1. 6wpi_nonpid
2. 6wpi_pid
3. 193wpi_nonpid
4. 193wpi_pid



3.3 Results and Discussion

Various simulations and analyses were performed to inform and validate the design of MotifBinner. The datasets used were described in the previous chapter. The first part of this section carefully investigates the primary problems encountered while processing PID datasets and motivates the solutions implemented in MotifBinner. To quantify the benefits of the PID approach, two datasets are processed twice, once using the information contained in the PIDs and once ignoring that information. The results obtained from the different processes are compared and discussed.

3.3.1 ~~Justification for~~ Validation of MotifBinner's design. ~~the design of MotifBinner~~

As illustrated in section 3.1, MotifBinner executes a number of steps to process PID data. This section aims to highlight the importance of each of these steps, providing supporting data to justify their inclusion.

3.3.1.1 Locate and identify the PIDs in each sequence

The identification of the PID motif is essential for processing data according to this method. However, sequencing error in the specific primer region(s) – used as the search motif, reduce the number or 'hits', if only exact matches are considered. Allowing for errors in these regions, (the prefix and suffix motifs), allows the recovery of more PIDs. Figure 24 shows an example of the primer region in raw sequences.

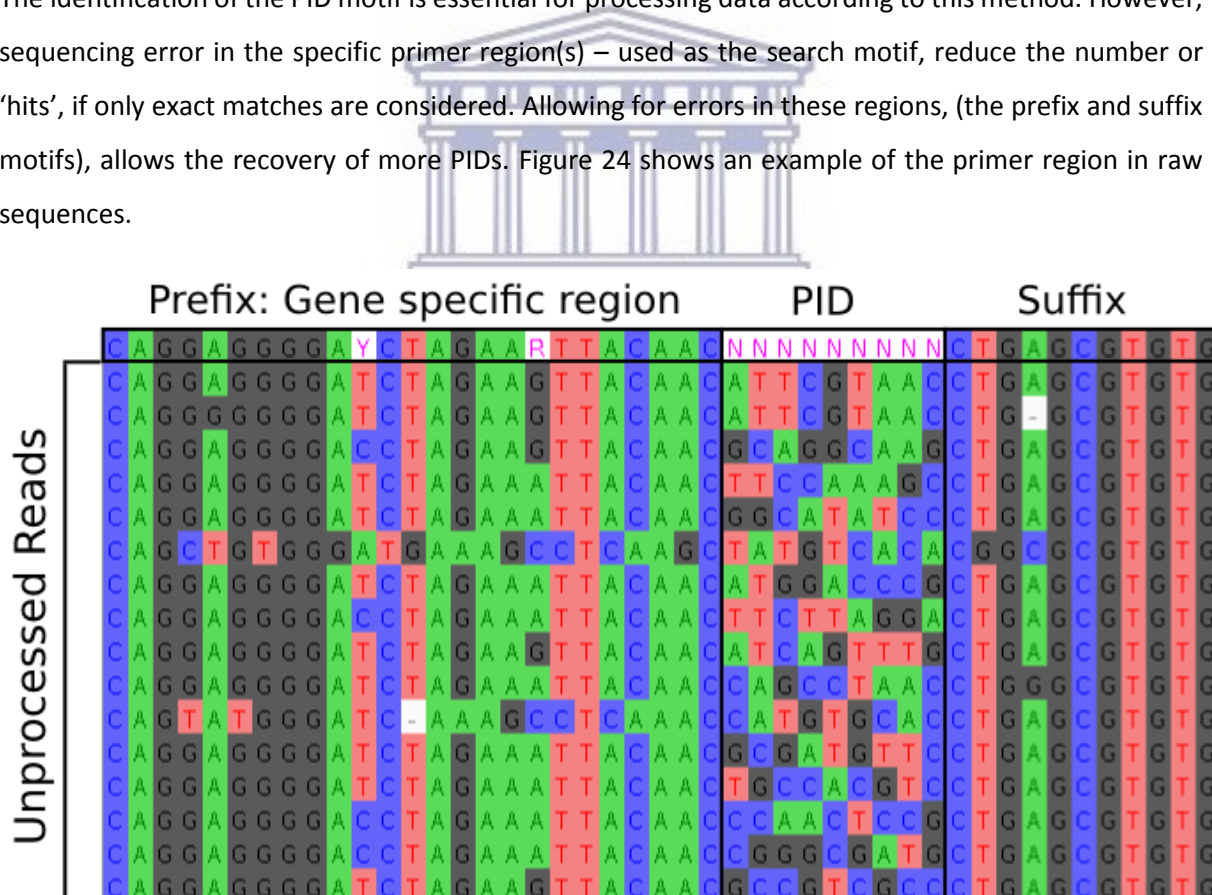


Figure 24: An alignment of selected sequences, over the PID containing region, from the 6wpi dataset. The first line shows the search motif.

The increase in the number of matches that are returned when allowing for mismatches in the search motif is shown in Table 8. The importance of this step is highlighted when one considers that PIDs could be found in 7510 more sequences when using fuzzy matching instead of exact matching. This additional 7510 sequences increases the total number of usable sequences by 25% from 29989 to 37499. These sequences were typically of lower quality than those found when no mismatches were allowed (average quality scores of 34.6 and 32.2² respectively). However, the average quality of these sequences was high enough (< 1 in 1000 expected error) to justify their inclusion.

Table 8: The number and quality of sequences in which PIDs were found in the 6wpi dataset for different levels of allowed mismatches in the search for the motif (CAGGAGGGGAYCTAGAARTTACAACNNNNNNNNNCTGAGCGTGGTG). Each row shows the number of additional hits resulting from an increase in the number of mismatches allowed. A total of 29989 PIDs were found when using an exact search. Of the 8979 sequences in which no hits were found when no mismatches were allowed, a PID could be found in 4917 when allowing one mismatch.

Sequences Searched	Hits	Mismatches Allowed	Mean Quality	Interquartile Range of Quality
38968	29989	0	34.6	33.7 - 36.3
8979	4917	1	32.6	30 - 35.6
4062	1013	2	29.7	27 - 31.9
3049	560	3	30.5	26.7 - 34.6
2489	264	4	30.4	26.6 - 34.1
2225	252	5	32.3	28.5 - 35.7
1973	177	6	32.7	29.5 - 35.7
1796	265	7	33.6	31.8 - 35.9
1531	62	8	30.7	27.8 - 34.8

The PID search proceeds by iteratively lowering the number of mismatches allowed, ensuring that matches are always achieved using the highest specificity. The maximum allowed number of mismatches is specified using the `max_mismatch` parameter. The number of mismatches allowed in the first search is set using the `max_mismatch_start` parameter. This design prevents the situation where the search string used to find the PID matches multiple locations in the sequence as illustrated in Figure 25.

² 32.2 is the weighted average of the qualities of the sequences in which PIDs were found when allowing between 1 and 8 mismatches.

Primer Design					A	G	T	S	A	A	T	T	N	N	N	N	C	A	G	C	
																					Mismatches
Sequence	T	G	T	G	A	A	T	C	A	A	T	T	C	A	C	C	C	A	G	C	
Match 1					A	G	T	S	A	A	T	T	N	N	N	N	C	A	G	C	1
Match 2	A	G	T	S	A	A	T	T	N	N	N	N	C	A	G	C					3

Figure 25: A contrived example in which a single search pattern can match the same sequence at multiple locations.

Finding rare variants is one of the main applications of the PID approach. In this case each additional sequence included in the dataset improves the odds of finding a potentially rare variant. Thus, justifying the use of fuzzy matching in the identification of the PID motifs.

3.3.1.2 Discard bins based on invalid PIDs

When a sequencing error occurs within the PID tag, this creates either a false unique PID tag, or causes the tag to erroneously be identical to another PID tag sequence. For the first example, consider the situation where a PID of length 9 is represented 500 times in the dataset (due to PCR amplification of the specific cDNA sequence).

In this example 4500 nucleotides will be sequenced just for the PIDs themselves. With an error rate of 1 in 100, it is expected that roughly 45 sequencing errors will be made, leading to approximately 45 reads in the dataset with incorrect PIDs. This creates a parent bin of 455 sequences, with up to 45 ‘offspring’ bins, where a PID that was sequenced correctly is referred to as the parent PID and those with sequencing errors are called offspring PIDs. (S. Zhou et al., 2015) first reported this phenomena and described a computational method for resolving it.

A number of variables had to be considered during the implementation of a modified version of the approach described in (S. Zhou et al., 2015), namely:

- The selection of a suitable modelling approach
- Determining the optimal sample size required to estimate the 99th percentile
- Identifying the most appropriate regression model to fit, and the fitting procedure to use
- Determining the linearity of the relationship over the size of the sliding window

3.3.1.2.1 Modelling approach

While powerful methods for centile estimation exist (Rigby & Stasinopoulos, 2005), we decided that a more basic approach was appropriate. For small ranges of parent bin sizes, we estimate the 99th percentile of the main offspring bin size and then fit a nonlinear regression model through those estimated percentiles. The motivation for this decision is threefold:

- Accessibility of the concepts to the intended audience: The concepts of a percentile and nonlinear regression are much more widely known than centile estimation;
- Simplicity of the model: The nonlinear regression model is a simple 4 parameter model, while centile estimation methods employ smoothers, making them harder to distribute, and communicate;
- Abundance of data: Since the data is generated using a basic simulation, we can generate very large amounts of data, adequately compensating for the lower power of the more basic approach.

3.3.1.2.2 Sample size required to estimate the 99th percentile

Modelling the relationship between the main offspring bin size and the parent bin size requires estimating the 99th percentile of the distribution of main offspring bin size over a sliding window of parent bin sizes. An important consideration is the sample size required to obtain a decent estimate of the 99th percentile.

The absolute bias when estimating percentiles and the mean was investigated by simulating 1000 samples from a standard normal distribution, and comparing the estimates of the mean, 95th percentile, and 99th percentile to the analytically computed values. The results are shown in Figure 26 and Table 9.

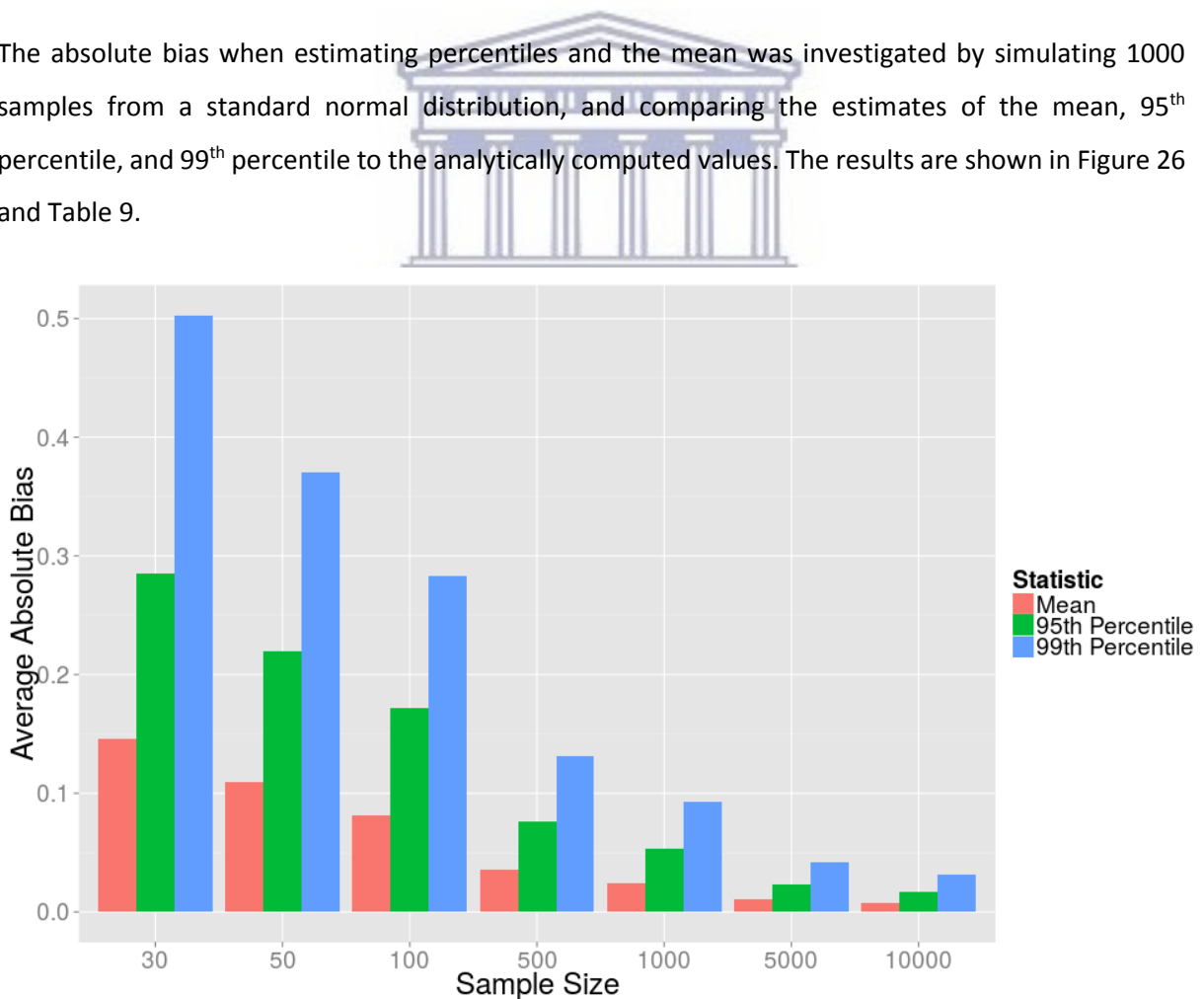


Figure 26: The absolute bias when estimating the mean, 95th percentile and 99th percentiles for various sample sizes.

The absolute bias for estimating the 99th percentile is significantly higher than for the mean, but decreases steadily as sample size increases, suggesting that large sample sizes should be used.

Table 9: The absolute bias when estimating the mean, 95th percentile and 99th percentiles for various sample sizes.

Sample Size	Mean	95 th percentile	99 th percentile
30	0.146	0.292	0.508
50	0.112	0.228	0.381
100	0.081	0.167	0.297
500	0.036	0.076	0.134
1000	0.026	0.054	0.095
5000	0.011	0.024	0.043
10000	0.008	0.017	0.030

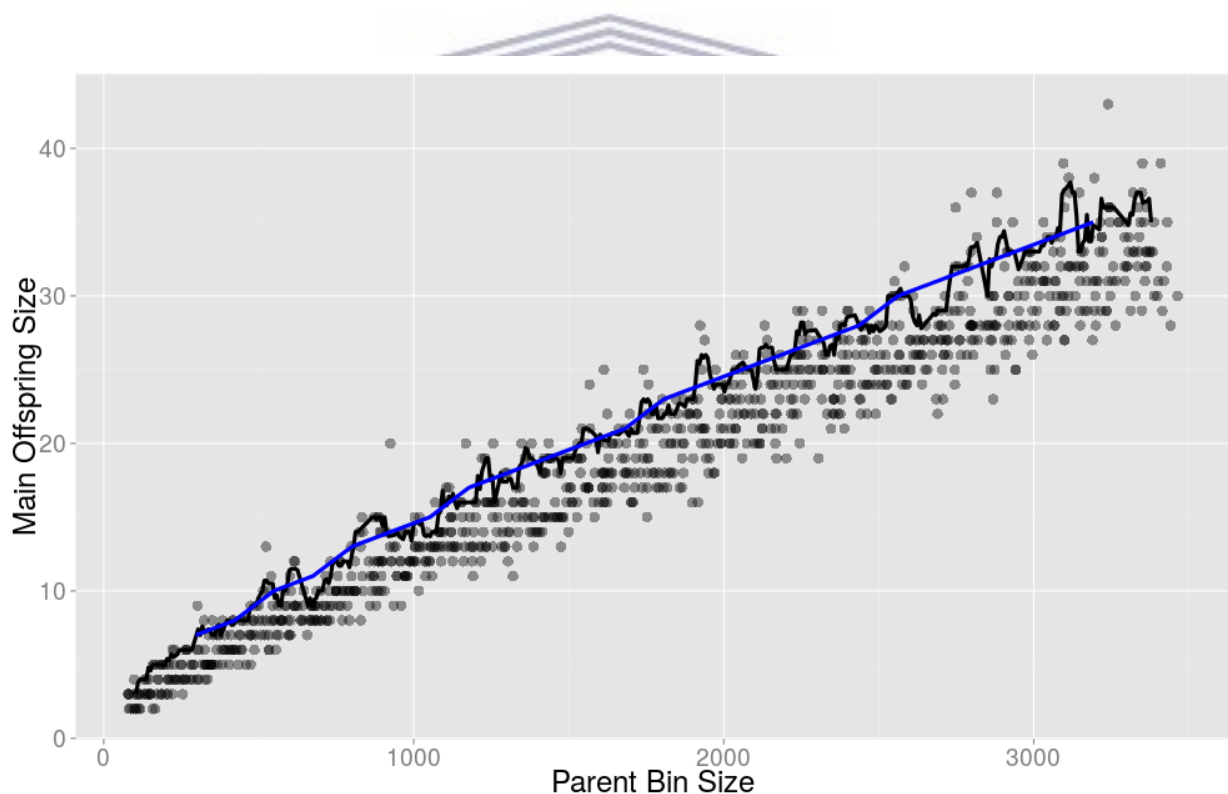


Figure 27: Relationship between parent and offspring bin sizes. A small simulated dataset (5 simulations for each step in parent bin size where the parent bin step size was 20) of the relationship between the parent bins size and the main offspring bin size. The 90th percentile was computed along a sliding window on this dataset (shown as the black line). Additionally the 90th percentile was computed along the same sliding window on a very large dataset (2000 simulations for each step in parent bin size where the parent bin step size was 20) simulated with identical parameters (shown as the blue line).

The effect of this result is illustrated in Figure 27, where the 90th percentiles were estimated on small and large datasets. Note the much smoother (blue) line resulting from the larger dataset. The 90th percentile was used instead of the 99th for this illustration since the 99th percentile is exceedingly variable on the small sample dataset.

3.3.1.2.3 Model selection and fitting approach

The model formula $y = a + bx + cx^d + \varepsilon$ was inferred from visual inspection of simulated datasets. As described in section 2.3, quality control procedures were built into the fitting process to ensure that the models were reliably fitted to the datasets. Additionally, diagnostic plots were inspected to ensure that the formula was appropriate for the datasets. An example of such a diagnostic plot is shown in Figure 28. Note that there is no obvious relationship between the standardized residuals and the fitted values other than a very slight increase in the variance of the residuals. The arcs in the plot result when the discrete observed bin sizes is compared to the continuous fitted values.

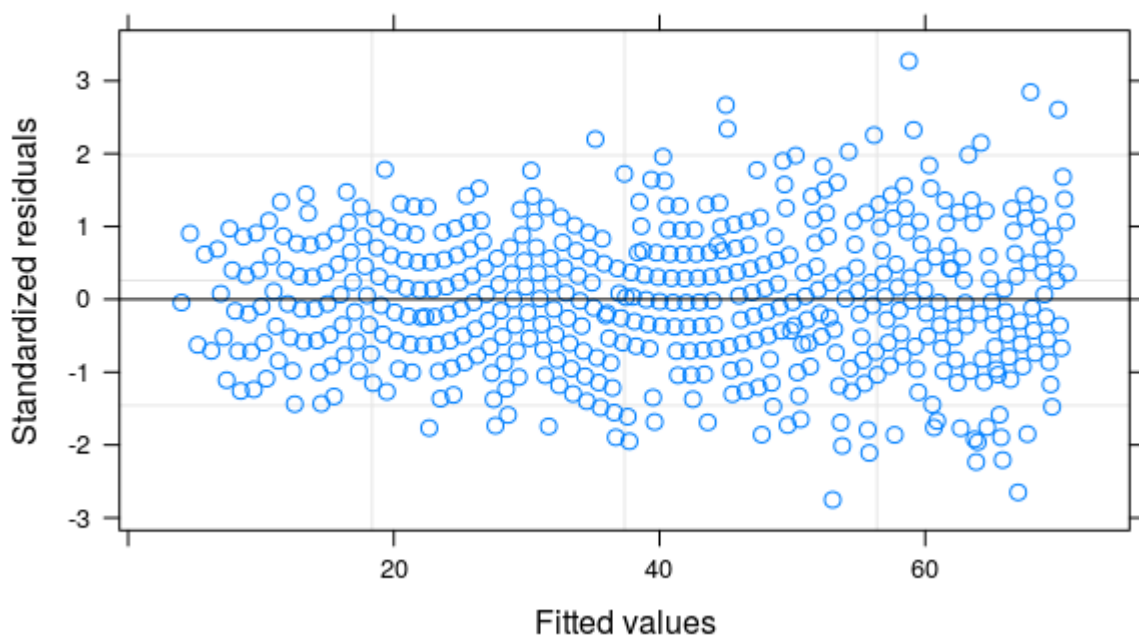


Figure 28: A diagnostic plot to illustrating the goodness of fit of the model of parent and main offspring bin sizes. The standardized residuals plotted against the fitted values of the nonlinear model fitted to the dataset simulated for the case when the PID is of length 8 and the assumed sequencing error rate is 1 in 100.

3.3.1.3 Determine which bins are chimeric

The PID approach assumes that all sequences with an identical PID tag originated from the same cDNA molecule. However, in some cases, different sequences variants are labelled with the same PID tag (Sheward et al., 2012). In cases where many PID bins in the dataset contain multiple sequences from

more than one input template, it indicates a potential flaw in one or more of the sample preparation steps. The occurrence of such cases has the potential to influence the resulting consensus sequence for that PID bin, making the detection, and exclusion, of these sequences an important process. We termed these bins 'chimeric', identified them by comparing the number of non-consensus nucleotides in each position to a binomial distribution. To improve the power of the method, the comparison is extended to two positions so that the bin will be called chimeric if either:

1. The number of deviations from the consensus letter at any single position is 'very large', or
2. If the number of deviations from the consensus letter at any two positions are 'large'.

Using simulated data, we determined that that this approach performs well for detecting chimeric bins. Thus, these data were subsequently used to calibrate the rejection criteria for the null hypotheses as stated in section 3.1.4. Here we outline the assumptions and approach to the simulations, before presenting a demonstration of the simple rejection criterion based on only one position. We then extend the method to the compound rejection criterion, by testing a number of cutoffs for the two parts of the compound criterion.

3.3.1.3.1 Calibration of chimera detection with simulations

Let n denote the number of sequences of length m after alignment in a bin and ϕ denotes the true sequencing error rate. Under the assumption that all the sequences are from the same input template, the number of deviations at any position follows a binomial distribution with the size parameter given by n and the chance of success given by ϕ . The assumption of a single input template can then be validated by computing how likely it is to observe the number of deviations from the consensus letters at a single given position, using the previously mentioned binomial distribution.

This approach must be applied to all positions in the alignment, hence a correction must be made for the multiple testing of hypotheses. For this purpose, the Holm-Bonferroni correction was applied in MotifBinner (Holm, 1979).

In order to simulate the effectiveness of detecting chimeric bins using this approach, a number of bins, with varying levels of chimerism, were simulated using varying conditions outlined below.

In short, a single input sequence of length 300 was used as the original input template (template A). A second input template (template B) was constructed by introducing a varying number of nucleotide substitution into the original sequence. A varying number of reads were then generated from each of template A and B. In order to generate these reads, sequencing error had to be simulated. This was achieved by drawing a random number from a binomial distribution for each read. The size of the binomial distribution was equal to the length of the template and the success probability was equal

to the sequencing error rate. For each read, nucleotide substitutions were introduced at a number of positions equal to the number that was drawn from the binomial distribution for the read. The positions were randomly sampled without replacement. The resulting simulated sequences from templates A and B were then pooled together to create the bin. The chimeric detection test, described in section 2.4, was then applied to the simulated bin, and the result recorded. This process was repeated 1000 times for each case as outlined below.

In cases where the simulated bins had zero reads from template B, i.e. all reads were from template A, a false positive will occur if the bin is flagged as chimeric. For bins in which the number of reads from template A was similar to the number of reads from template B, a true positive will result when then such a bin is labelled as chimeric.

To explore the simple approach based on chimeric bin detection using a single position, simulations using at five separate variables had to be carried out. To simultaneously explore the five variables, using multiple options for each variable, would require exploring a five dimensional space, leading to an intractable number of simulations. Instead, each factor was evaluated, using sensible defaults for the other factors.

- The effect of different bin sizes must be explored. Two values were used for defaults namely 10 and 20. In addition, values of 5, 6, 8, 15, 30, 50 and 100 were also considered.
- The effect of using different p-values for the rejection criterion were explored. The default was set to 0.001, with additional values to of 0.0001, 0.01 and 0.05 also evaluated.
- The impact of different sequencing error rates also had to be explored. A default setting of $\frac{1}{100}$ was used, with the following values also investigated: $\frac{1}{50}$, $\frac{1}{75}$, and $\frac{1}{125}$.
- The influence of different sequence lengths also needed to be explored. A default length of 300bp was used, with additional sequence lengths of 150, 500 and 750 also evaluated.
- Finally, the effect of different levels of sequence divergence between templates A and B had to be investigated. The level of divergence was set to 1 nucleotide substitution between templates A and B as a default value, with values of 2, 3, 5, 10, 30 and 60 also evaluated.

As each factor is presented, a brief discussion will motivate the choice of default and alternative values.

The first set of simulations explore the effectiveness of the approach by bin size, see Table 10. Note that no more than 50% of the sequences are allowed to be from the second template. This is because the effect is symmetric around 50% and bins with more than 50% chimerism will not add any information. For small bin sizes, the meaning of a chimeric bin in the presence of potential sequencing

error is not clearly defined. For example a bin of size 5 has 3 sequences from the original template and 2 from the second template in the most extreme case. This is not enough to distinguish a legitimate chimeric bin from sequencing error. As the bin size increases, the ability to detect chimeric bins requires a smaller and smaller proportion of sequences from second template for the bin to be labeled as chimeric.

Table 10: The effect of bin size on the detection of chimeric bins. The proportion of samples labelled as chimeric for bins of size 5, 6, 8, 10, 15, 20, 30, 50 and 100 with the number of reads from template B ranging from 0 to 10, and default sequencing error rate ($\frac{1}{100}$), adjusted p-value used for cutoff (0.001), and sequence length (300). The first row represents false positives and are highlighted in red. All cells where more than 90% of bins were flagged as chimeric are highlighted in green. 1000 samples were simulated for each case.

		Bin Size								
		5	6	8	10	15	20	30	50	100
Number of sequences from second template	0	1.1%	0.2%	1.3%	2.0%	0.6%	0.7%	0.5%	0.0%	0.2%
	1	1.0%	1.0%	1.3%	2.1%	0.3%	1.0%	0.5%	0.3%	0.2%
	2	5.4%	6.0%	6.9%	6.6%	1.3%	2.0%	0.6%	0.4%	0.1%
	3		98.5%	95.1%	66.8%	12.3%	14.5%	2.6%	1.5%	0.3%
	4			100.0%	99.5%	98.7%	87.7%	23.4%	8.1%	2.5%
	5				100.0%	100.0%	99.9%	98.0%	35.9%	6.7%
	6					100.0%	100.0%	100.0%	98.6%	21.7%
	7					100.0%	100.0%	100.0%	100.0%	61.4%
	8						100.0%	100.0%	100.0%	98.6%
	9							100.0%	100.0%	100.0%
	10								100.0%	100.0%

Based on these bin size simulations, the next simulations were carried out using bin sizes of 10 and 20, since a bin of size 20 should reliably be called chimeric, while a bin of size 10 provides a reasonable idea of how the sensitivity of the method is affected for bins of smaller size.

The next set of simulations explored different thresholds for the rejection criterion (shown in Table 11). It is clear that the value obtained after the Holm-Bonferroni correction is no longer an accurate p-value. This is evident from the fact that in cases where the null hypothesis is true (first row), the proportion of bins incorrectly labelled as chimeric is much larger than the cutoff, for cases where the cutoff equals 0.01 or 0.05. This breakdown in the statistical meaning of the p-values does not concern us since we are only interested in constructing a reliable mechanism for detecting chimerism. We note

that the proportion of chimeric bins incorrectly called chimeric, when there is a very small number of reads from the second template, is very low when a p-value of either 0.001 or 0.0001 was used as cutoff. Also, as the number of reads from the second templates increases, the proportion of bins identified as chimeric increases rapidly from near zero percent to near 100%, implying that both 0.001 and 0.0001 are sensible cutoffs. We chose to set the p-value to 0.001.

Table 11: The effect of the adjusted p-value used as cutoff on the detection of chimeric bins. The proportion of samples labelled as chimeric for cutoffs of 0.0001, 0.001, 0.01 and 0.05 with the number of reads from template B ranging from 0 to 10, and defaults for bin size (10 & 20), sequencing error rate ($\frac{1}{100}$), and sequence length (300). The first row represents false positives and are highlighted in red. All cells where more than 90% of bins were flagged as chimeric are highlighted in green. 1000 samples were simulated for each case.

p-values		Bin size = 10				Bin size = 20			
		0.0001	0.001	0.01	0.05	0.0001	0.001	0.01	0.05
Number of sequences from second template	0	0.0%	2.1%	4.4%	54.3%	0.0%	1.3%	6.6%	28.2%
	1	0.0%	2.3%	4.5%	52.7%	0.2%	1.2%	5.5%	26.4%
	2	0.2%	5.0%	11.4%	72.4%	0.1%	2.5%	7.5%	38.6%
	3	7.8%	63.5%	99.3%	99.9%	1.6%	13.4%	34.0%	99.2%
	4	98.6%	99.8%	100.0%	100.0%	14.7%	86.7%	98.9%	100.0%
	5	100.0%	100.0%	100.0%	100.0%	98.4%	99.8%	100.0%	100.0%

The third factor to consider is the effect of the assumed sequencing error rate on the effectiveness of the technique. Table 12 presents the results from a simulation exploring sequencing error rate. A higher sequencing error rate will imply that a higher number of deviations from the consensus letter is expected at each position so that it will be harder to detect chimeric bins. From the table we see that the method is still effective over the range of error rates exhibited by current sequencing technologies.

Table 12: The effect of the sequencing error rate on the detection of chimeric bins. The proportion of samples labelled as chimeric for sequencing error rates of and $\frac{1}{125}$, $\frac{1}{100}$, $\frac{1}{75}$ and $\frac{1}{50}$ with the number of reads from template B ranging from 0 to 10, and defaults for bin size (10 & 20), adjusted p-value used for cutoff (0.001), and sequence length (300). The first row represents false positives and are highlighted in red. All cells where more than 90% of bins were flagged as chimeric are highlighted in green. 1000 samples were simulated for each case.

Sequencing error rate	Bin size = 10				Bin size = 20			
	1/125	1/100	1/75	1/50	1/125	1/100	1/75	1/50
0	1.7%	2.1%	1.2%	0.3%	1.2%	1.2%	1.2%	0.9%
1	1.9%	1.3%	1.3%	1.8%	0.8%	1.7%	0.5%	0.9%
2	7.5%	7.1%	2.3%	1.9%	1.9%	2.2%	1.5%	1.4%
3	92.5%	64.6%	15.6%	14.0%	12.6%	14.9%	3.4%	3.7%
4	100.0%	99.8%	98.5%	96.0%	98.5%	87.0%	25.2%	19.8%
5	100.0%	99.9%	100.0%	99.9%	100.0%	99.7%	98.4%	78.2%
6					100.0%	100.0%	100.0%	99.3%
7					100.0%	100.0%	100.0%	100.0%
8					100.0%	100.0%	100.0%	100.0%
9					100.0%	100.0%	100.0%	100.0%
10					100.0%	100.0%	100.0%	100.0%

The effect of sequence length on chimera detection was evaluated next (shown in Table 13). As the sequences became longer, more comparisons had to be made and hence a more severe correction must be applied by the Holm-Bonferroni technique. It is clear that the effect of sequence length is very minor.

Table 13: The effect of the sequence length on the detection of chimeric bins. The proportion of samples labelled as chimeric for sequence lengths of 150, 300, 500 and 750, with the number of reads from template B ranging from 0 to 10, and defaults for bin size (10 & 20), sequencing error rate ($\frac{1}{100}$), adjusted p-value used for cutoff (0.001). The first row represents false positives and are highlighted in red. All cells where more than 90% of bins were flagged as chimeric are highlighted in green. 1000 samples were simulated for each case.

Sequence Length	Bin Size = 10				Bin Size = 20			
	150	300	500	750	150	300	500	750
0	1.8%	2.0%	1.7%	1.2%	0.6%	1.5%	1.8%	1.9%
1	1.0%	2.5%	1.5%	0.8%	0.8%	0.7%	1.0%	1.5%
2	8.0%	7.1%	4.7%	1.8%	2.2%	2.0%	2.4%	1.9%
3	77.0%	65.4%	36.4%	16.5%	14.7%	13.9%	11.8%	7.0%
4	99.8%	99.6%	99.0%	99.2%	92.0%	88.3%	70.1%	45.1%
5	100.0%	100.0%	100.0%	100.0%	99.8%	99.9%	99.8%	99.0%
6					100.0%	100.0%	100.0%	100.0%
7					100.0%	100.0%	100.0%	100.0%
8					100.0%	100.0%	100.0%	100.0%
9					100.0%	100.0%	100.0%	100.0%
10					100.0%	100.0%	100.0%	100.0%

Lastly the effect of the magnitude of the genetic distance between the two templates was explored. It is expected that as the distance between the two templates becomes larger, it becomes easier to detect chimerism. This effect was not observed for smaller differences between the two templates (up to 10 substitutions), with the level of detection remaining fairly constant (Table 14).

For larger differences, the opposite was observed – it became harder to detect chimerism. This is likely due to the fact that the sequencing error rate used in the test for chimeras was estimated from the sequences in the bin. For very divergent sequences, the estimated error rate becomes very large. This failure to estimate the sequencing error correctly might also explain the behavior observed for bins of size ten when the difference between the templates was ≥ 30 substitutions (highlighted in yellow).

Table 14: The effect of the divergence between templates A and B on the detection of chimeric bins. The proportion of samples labelled as chimeric when the difference between the two templates are 1, 2, 3, 5, 10, 30 or 60 substitutions, with the number of reads from template B ranging from 0 to 10, and defaults for bin size (10 & 20), sequencing error rate ($\frac{1}{100}$), adjusted p-value used for cutoff (0.001), and sequence length (300). The first row represents false positives and are highlighted in red. All cells where more than 90% of bins were flagged as chimeric are highlighted in green. 1000 samples were simulated for each case.

Number of mutations		Bin Size = 10						
		1	2	3	5	10	30	60
Number of sequences from second template	0	2.8%	2.4%	2.2%	2.1%	1.7%	1.9%	1.9%
	1	3.4%	1.6%	1.9%	1.6%	0.5%	0.3%	0.0%
	2	7.2%	6.9%	9.8%	4.9%	2.8%	0.6%	0.6%
	3	63.2%	49.1%	33.9%	30.1%	49.7%	6.6%	0.0%
	4	99.1%	100.0%	100.0%	100.0%	99.1%	70.6%	0.2%
	5	100.0%	100.0%	100.0%	100.0%	100.0%	2.5%	0.0%
Number of mutations		Bin Size = 20						
		1	2	3	5	10	30	60
Number of sequences from second template	0	0.8%	1.1%	1.0%	1.0%	0.8%	0.8%	0.7%
	1	0.7%	1.6%	1.0%	1.6%	1.0%	0.0%	0.2%
	2	2.8%	3.5%	3.9%	4.0%	1.9%	2.0%	0.4%
	3	16.1%	22.5%	27.7%	25.5%	11.8%	1.9%	0.0%
	4	89.8%	81.9%	71.4%	61.0%	80.5%	27.6%	2.2%
	5	99.7%	100.0%	100.0%	100.0%	100.0%	73.3%	3.3%
	6	100.0%	100.0%	100.0%	100.0%	100.0%	97.8%	29.4%
	7	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	34.8%
	8	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	99.4%
	9	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	99.3%
	10	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%

One solution is to let the user input the error rate, but this was rejected because the number of parameters the user must specify is already large. Another solution is to base the rejection criterion on more than one position. The question then becomes what threshold to use, and how many positions to evaluate? For simplicity we only considered a single extra position and a simulation was performed to explore different cutoffs (see Table 15).

Table 15: The effect of using different cutoff values when using two positions to detect chimerism. The proportion of samples labelled as chimeric for cutoffs based on the adjusted p-values of two positions of 0, 0.002, 0.005 and 0.01 with the number of reads from template B ranging from 0 to 10, and defaults for bin size (10 & 20), cutoff based on a single position of 0.001, sequencing error rate ($\frac{1}{100}$), and sequence length (300). Templates A and B differ by 3 substitutions. The first row represents false positives and are highlighted in red. All cells where more than 90% of bins were flagged as chimeric are highlighted in green. 1000 samples were simulated for each case.

p-values	Bin Size = 10				Bin Size = 20			
	0	0.002	0.005	0.01	0	0.002	0.005	0.01
0	2.8%	3.2%	1.3%	2.5%	0.6%	1.0%	1.4%	1.7%
1	2.0%	1.6%	2.4%	2.3%	1.5%	1.5%	1.2%	1.4%
2	9.1%	9.9%	10.6%	11.0%	3.8%	4.7%	4.0%	4.6%
3	36.8%	70.5%	99.2%	100.0%	29.2%	32.8%	27.5%	34.1%
4	100.0%	100.0%	100.0%	100.0%	69.6%	95.6%	100.0%	100.0%
5	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%
6					100.0%	100.0%	100.0%	100.0%
7					100.0%	100.0%	100.0%	100.0%
8					100.0%	100.0%	100.0%	100.0%
9					100.0%	100.0%	100.0%	100.0%
10					100.0%	100.0%	100.0%	100.0%

By considering an additional position when determining chimerism, the method became more sensitive in detecting the presence of chimeric bins, while maintaining a low probability of a type I error. When comparing Table 15 with Table 14, one notes that the likelihood of calling a bin chimeric is higher when at least three reads are from template B, when using the compound criterion as opposed to only using a single position. Setting the p-value cutoff for the compound criterion to 0.005 or 0.01 increased the power to detect chimerism, while also retaining a low false positive rate. A final simulation was performed to determine if the compound rejection criterion method addressed the issues related to large deviations between the two templates (see Table 16).

Table 16: The effect of the divergence between templates A and B on the detection of chimeric bins when a cutoff based on two positions is used. The proportion of samples labelled as chimeric when the difference between the two templates are 1, 2, 3, 5, 10, 30 or 60 substitutions, with the number of reads from template B ranging from 0 to 10, and defaults for bin size (10 & 20), sequencing error rate ($\frac{1}{100}$), adjusted p-value used for cutoff based on one position (0.001), adjusted p-value used for cutoff based on two positions (0.005), and sequence length (300). The first row represents false positives which are highlighted in red. All cells where more than 90% of bins were flagged as chimeric are highlighted in green. 1000 samples were simulated for each case.

Number of mutations	of	Bin Size = 10						
		1	2	3	5	10	30	60
Number of sequences from second template	0	1.8%	1.8%	2.2%	2.0%	1.6%	2.1%	1.7%
	1	2.6%	2.5%	1.8%	1.9%	0.5%	0.4%	0.0%
	2	7.7%	9.3%	8.0%	10.1%	10.6%	0.5%	0.4%
	3	66.3%	98.5%	98.7%	92.8%	59.3%	12.1%	0.0%
	4	99.8%	100.0%	100.0%	100.0%	100.0%	79.0%	0.0%
	5	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	0.5%
Number of mutations	of	Bin Size = 20						
		1	2	3	5	10	30	60
Number of sequences from second template	0	1.2%	1.3%	1.2%	0.4%	0.9%	1.1%	1.3%
	1	0.9%	1.2%	0.9%	1.2%	1.1%	0.1%	0.0%
	2	2.4%	2.8%	3.9%	3.5%	4.4%	2.1%	0.1%
	3	15.3%	25.9%	31.2%	35.3%	48.7%	6.3%	0.1%
	4	89.3%	99.4%	100.0%	100.0%	93.4%	50.4%	2.9%
	5	99.9%	100.0%	100.0%	100.0%	100.0%	97.3%	15.0%
	6	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	33.2%
	7	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	99.5%
	8	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	99.9%
	9	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%
	10	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%

A comparison between Table 16 and Table 14 reinforces the previous finding that the compound rejection has slightly higher power to detect chimerism. It shows that the compound rejection criterion also has only a minor influence in the cases where the two templates differ significantly. Based on these findings, we conclude that the performance of using only a single position is adequate for our purposes. Hence the default behavior of MotifBinner is to use only a single position for

detecting chimerism. In order to change this behavior, the user will need to edit the `compute_bin_metrics` function.

3.3.1.3.2 Application of chimera detection to the datasets

Upon first analysis of the datasets, the 6wpi data was not trimmed so that all sequences started at the same position on the 5' end. When evaluating the number of chimeric bins in the datasets, we found that out of the 972 bins in the 6wpi dataset, 461 were flagged as chimeric. For the 193wpi dataset, 723 bins were checked for chimerism and 600 were flagged as chimeric. These are high rates of chimerism and a cause for concern. A number of bins were manually inspected to confirm the high rates of chimerism. From Figure 29 one can see that the 193wpi bins are chimeric since there are clusters of sequences with the same polymorphisms (the vertical lines). The chimerism of the 6wpi data is artificially inflated by the irregular starting positions of the sequences.

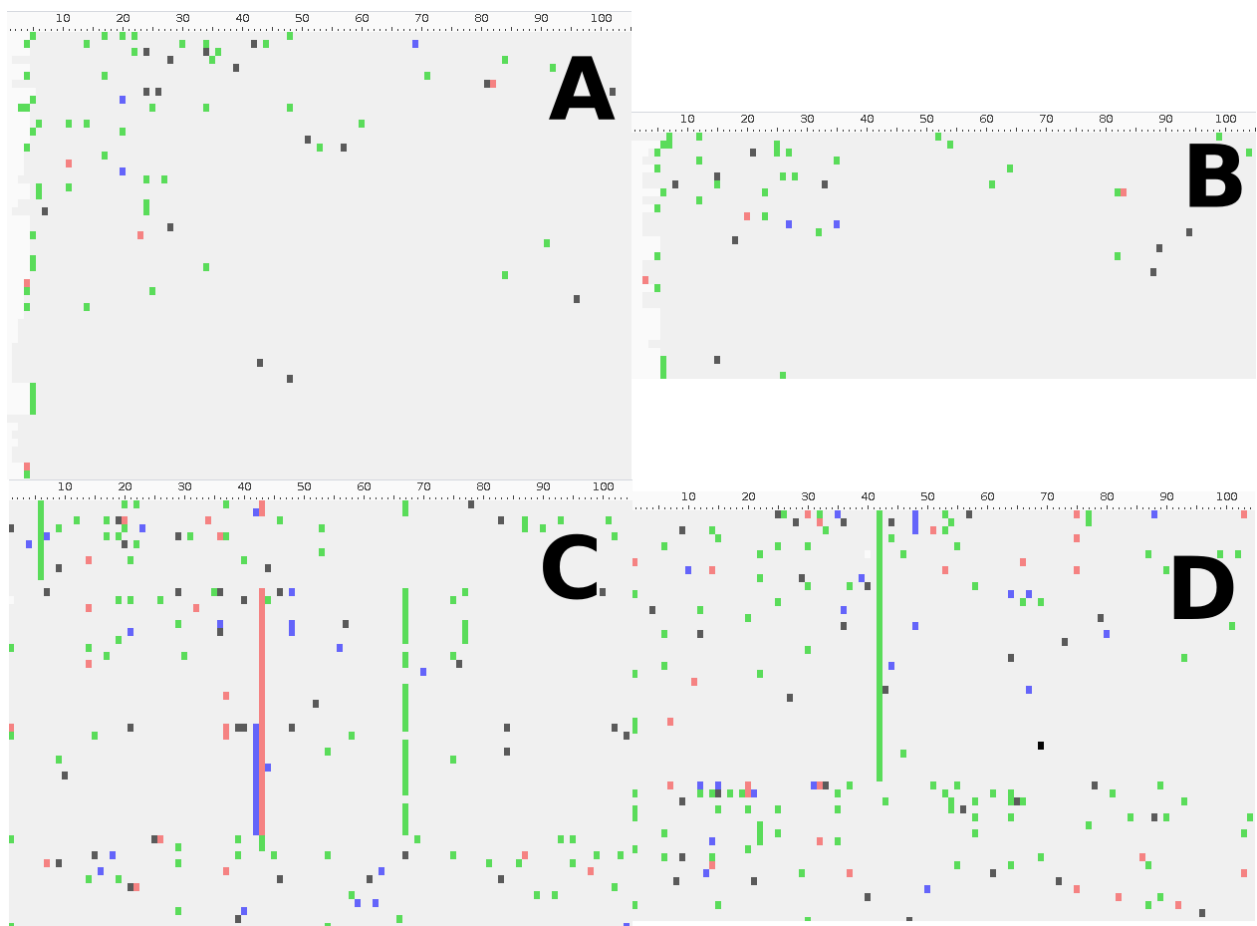


Figure 29: A sample of pieces of the alignments of bins from the datasets in which deviations from the consensus is highlighted. Results for the 6wpi dataset is based on an early version of the dataset in which the 5' end was not trimmed. From the top left proceeding clockwise: Bin AAAAAATT_46 of dataset 6wpi (A); Bin GCAGGCAAG_21 of dataset 6wpi (B); Bin GCATGCCTA_96 of dataset 193wpi (D) and Bin AAAAAATA_59 of dataset 193wpi (C).

After this discovery, the pipeline was modified to include trimming on the 5' end. This modification resulted in the number of chimeric bins dropping from 461 out of 972 (47%) to 291 out of 928 (31%). Figure 30 shows the same bins from the 6wpi dataset as Figure 29, except that the trimming step was corrected. One can see that the most variable region of the bin (the 5' end) has been removed. However, the rate of chimerism was still high.

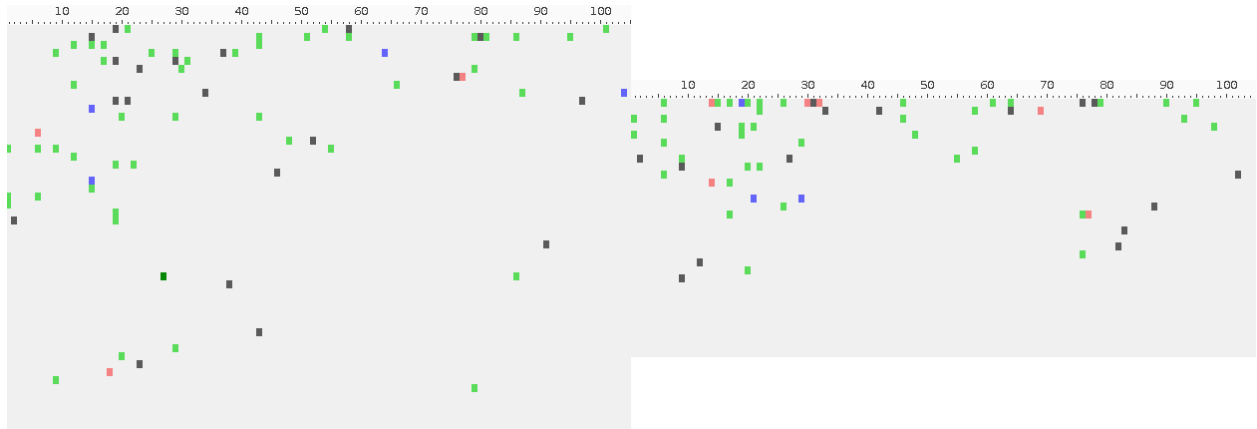


Figure 30: A sample of the alignments from bins in the correctly trimmed 6wpi dataset. Deviations from the consensus are highlighted. Bin AAAAAATTT_51 of dataset 6wpi (left); Bin GCAGGCAAG_21 of dataset 6wpi (right).

3.3.1.3.3 Investigation of the high rates of chimerism

A more comprehensive investigation of the high rates of chimerism was performed by comparing sequences in a bin to the consensus sequence of the bin, as well as to consensus sequences of other bins, see Figure 31. If a sequence is more similar to the consensus sequence of another bin than to the consensus sequence of its own bin, then it is likely to be true chimerism. Chimerism detection is further complicated by PCR recombination. A read of a molecule resulting from a PCR recombination event will be a hybrid in which one part of the sequence is from one input template and the rest of the sequence is from another input template. Depending on the differences in the templates involved in the recombination event such a read may be similar to one, both or neither of the input templates. This analysis will not adequately classify recombined reads and they may or may not be counted towards a bin's chimerism proportion based on differences in the input templates for the recombined read. Future work should explore the prevalence of and possible metrics for measuring PCR recombination events.

Figure 31 shows that on average, over 25% (26.5% and 28.9% for datasets 6wpi and 193wpi respectively) of the sequences in a bin are more similar to the consensus sequence of another bin. Noticeably, the distributions of sequences more similar to the consensus sequences of another bin is

very similar for the 6wpi and 193wpi datasets, in contrast to the results obtained using the test for chimerism build into MotifBinner, which reported rates of chimerism of 31% and 83% for the two datasets. For bins whose consensus sequences contained degenerate sequences, a very high percentage of the reads were closer to the consensus sequence of another bin.

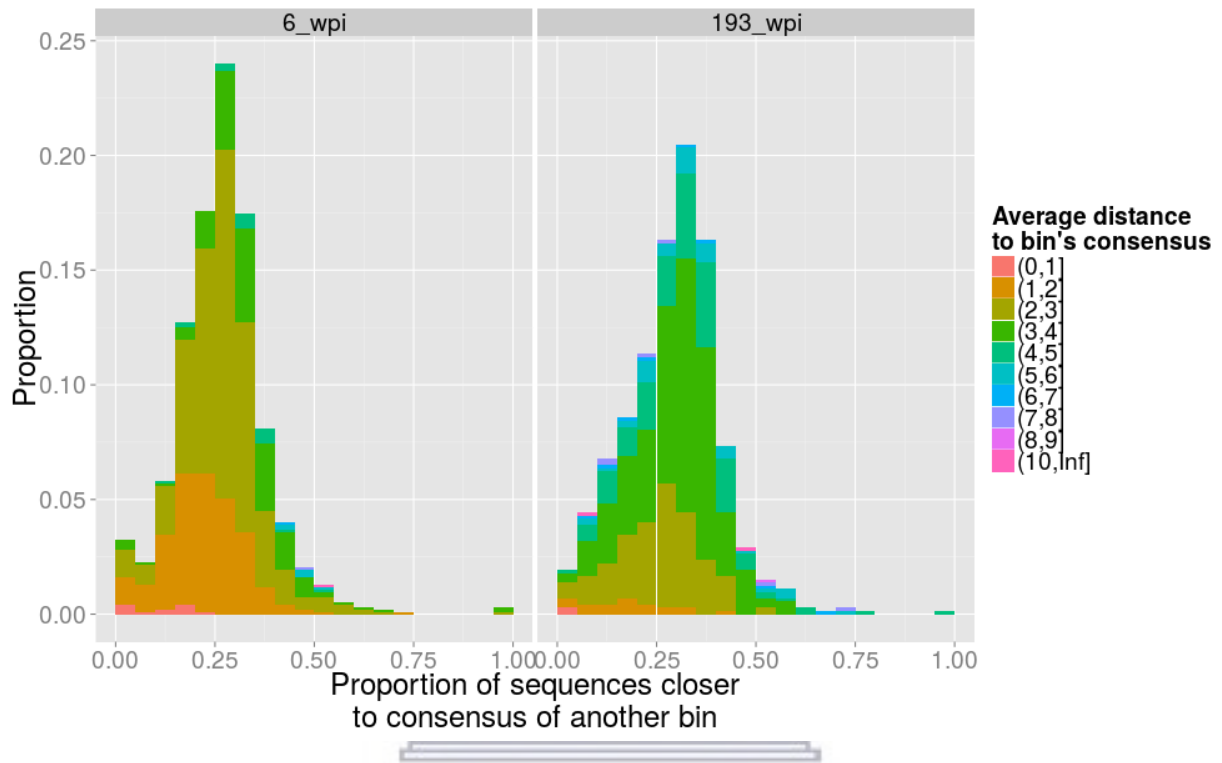


Figure 31: An investigation of the rates of chimerism. In both datasets, each sequence was assigned to the consensus sequence which is most similar to it. In each bin the proportion of sequences assigned to consensus sequences of other bins was computed. Histograms of these proportions are shown in the figure. The histograms are colored by the average distance between the consensus of the bin and the sequences in the bin.

Sequencing errors in the sequences of a particular bin might spuriously cause them to become more similar to the consensus sequence from another bin. In order to discover how frequently this occurs, a simulation was performed. The simulation recreated the 6wpi and 193wpi datasets, however, the bins were simulated in such a way that all of the reads were generated from the same template. In other words, none of the bins in the simulated dataset were chimeric. The datasets were simulated by using the consensus sequences of the real data, simulating the sequencing of each consensus sequence a number of times equal to the size of the bin. The sequencing error rate was computed separately for each bin by counting the number of mismatches between the sequences and the consensus of the bin (the restricted Levenshtein / edit distance) and dividing it by the total number of nucleotides in the bin. This error rate was used to simulate sequencing error as described in section 3.3.1.3.

Using these simulated datasets, the distance of the reads from their (and other) consensus sequences was calculated (Figure 32). In contrast to Figure 31, the distributions are different for the two simulated datasets. In the simulated dataset based on the 193wpi dataset, the average proportion of reads closer to the consensus of another bin is lower than previously observed (9.2% vs. 28.9%). This suggests that the probability of a random sequencing errors modifying sequences assigned to a bin in such a way that they become more similar to the consensus sequence of another bin is low. Thus, the high rate of chimerism (detected by comparing the sequences in the bins to the consensus sequences of other bins) in the 193wpi dataset is unlikely to be the result of sequencing error.

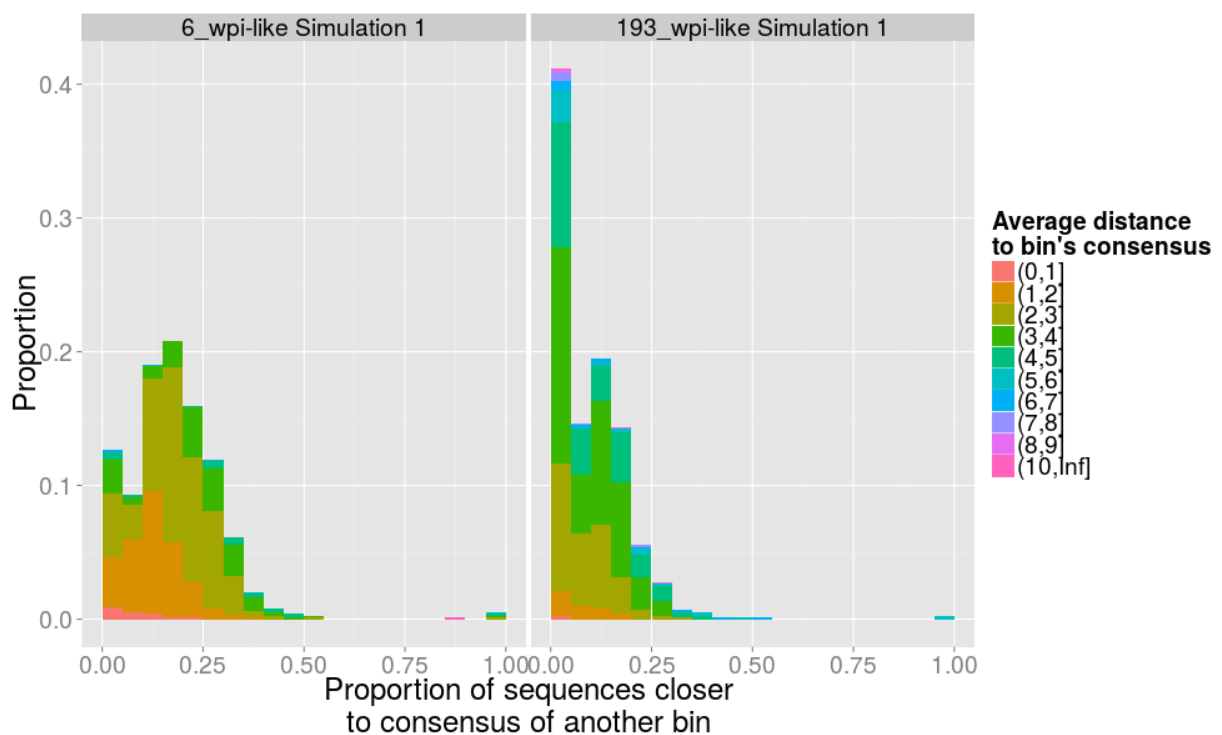


Figure 32: Simulation showing proportion of reads more similar to the consensus sequence of another bin when there is no chimeric bins. The construction of the plot was previously described Figure 31.

When an equivalent analysis was performed on the 6wpi dataset, the average proportion of reads closer to the consensus of another bin was also lower in the simulated version (17.5% vs 26.5%). However, the effect is less pronounced for the 6wpi dataset than for the 193wpi dataset. Additionally, in the simulated dataset based on the 193wpi time point, a large proportion of bins (41.4%) contained no sequences that were more similar to the consensus of another bin. In the other three datasets, this proportion was below 13%. These differences can be explained by the different levels of diversity in the 6wpi and 193wpi dataset. In the 6wpi dataset, the unique consensus sequences are very similar to each other with most differing by three or less nucleotide substitution from the other consensus

sequences. In the 193wpi dataset, most consensus sequences differ from each other by five to seven nucleotide substitutions.

3.3.1.3.4 Summary

Initial simulations of the method to detect chimeric bins were promising, showing low false positive rates and high sensitivity to detect chimerism in bins of size 10 and larger. When chimera detection was applied to real data, high, but variable rates of chimerism were observed, with the dataset containing greater sequence diversity showing a higher rate of chimerism. It is unclear whether this is due to a truly higher rate of chimerism in this dataset, or simply due to a better rate of detection.

A more detailed investigation compared the distance between sequences within a bin, to the consensus sequences of other bins. Sequences which were more similar to a consensus sequences from another bin provide evidence that the bin may have been chimeric. By comparing the results from the biological data to those from simulated data with no chimerism, the conclusion was reached that there are significant rates of chimerism. The effect of chimerism on the accurate construction of a true input template is assessed in section 3.3.2.4 PCR recombination presents additional unresolved complications to the measurement of chimerism.

3.3.1.4 Find and remove outlier sequences from chimeric bins

It is frequently the case that bins include sequences that are significantly different from the other sequences in the bin. Possible causes for this include two different input templates getting tagged with the same PID, a sequencing error in a previously unique PID causing it to become identical to another PID, and PCR recombination. The differences between sequences originating from different templates can be large (e.g.: the variable loops of *env* in HIV or when the PID approach is applied to metagenomics sequences) leading to complications in the alignment step.

In order to remove outlying sequences, the user specifies the highest sequencing error rate they are willing to allow in any given bin (specified via the `threshold` parameter). MotifBinner then iteratively removes the most outlying sequence from the bin until the maximum distance between any two sequences in the bin falls below the specified error rate. The result of this is that sequences derived from templates which are sufficiently divergent from the most abundant template in the bin, are removed before the alignment and consensus sequence generation steps. Additionally, any sequences derived from the most abundant template, but with an abnormally large number of sequencing errors, will also be removed. The result is that the sequence alignment for the bin is more accurate and thus, the possibility of deriving an erroneous or chimeric consensus sequence is reduced.

A plot showing the overall reduction in bin sizes resulting from outlier removal is included in the binning report. Figure 33 shows these plots for the two datasets. Note the larger average reduction in bin size for the 193wpi dataset (32.5% reduction), relative to the 6wpi dataset (19.1% reduction). This is in line with the higher level of chimerism identified in the 193wpi dataset. MotifBinner also produces a plot for each bin on which a principle component analysis (PCA) on the distance matrix of the sequences in the bin yielded two or more eigenvectors. The plot shows the distances between the sequences in the bin as a scatter plot based on the first two principal components obtained by the PCA. Figure 34 shows two examples of the scatter plots based on the PCA of the bins.

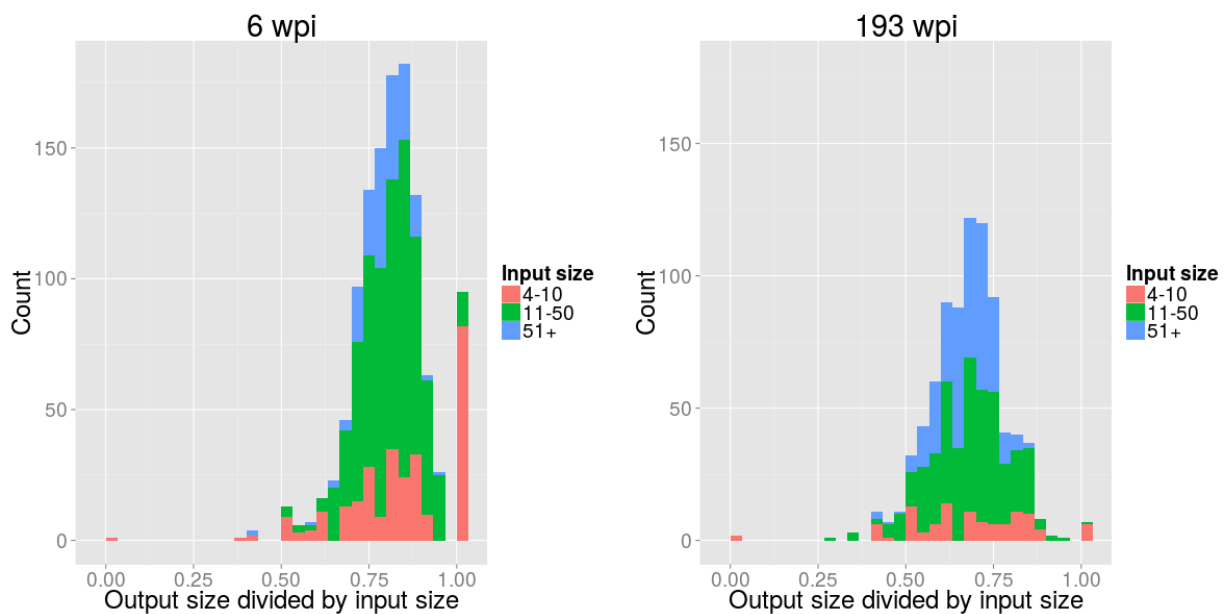


Figure 33: Histograms showing the reduction in size of the bins of the 6wpi dataset (left) and the 193wpi dataset (right) resulting from the removal of outliers. The x-axis shows the size of the bin after outliers were removed as a fraction of the bin size before outliers was removed. The y-axis counts how many bins fall in each category. The fill color indicates the size of the bin before outliers were removed. Note that bins of size 2 can only have the values 0 and 1. Likewise, bins of size 3 can only have the values 0, 0.67 and 1.

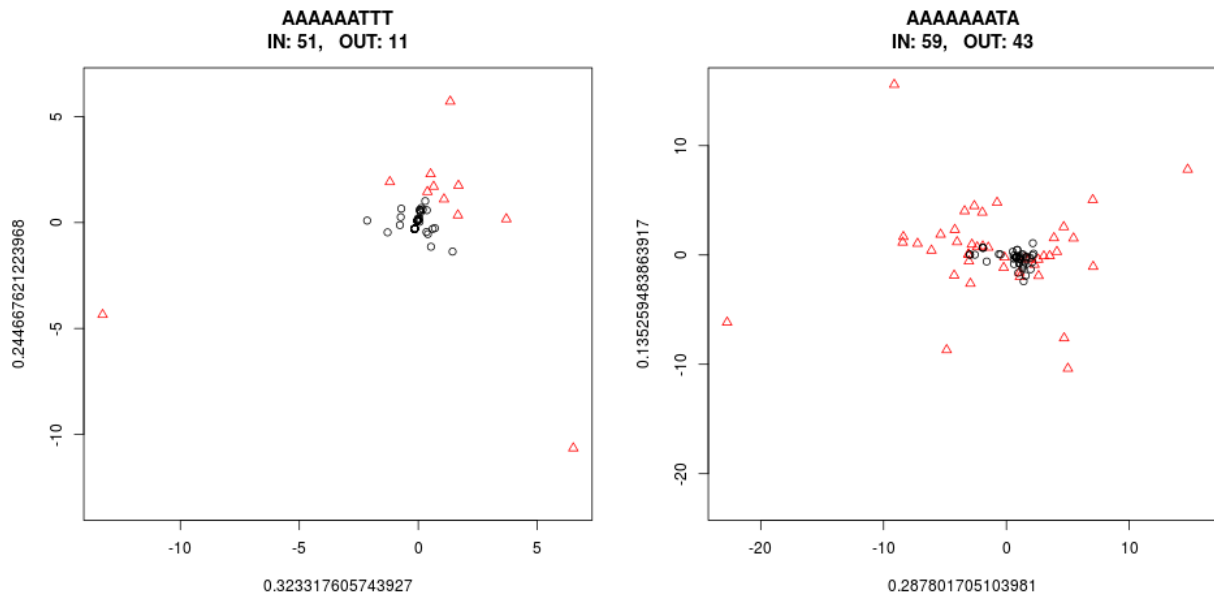


Figure 34: The first two principal components of a PCA analysis of the distance matrices performed on the AAAAAATTT and AAAAAATA bins of datasets 6wpi and 193wpi respectively. Each black dot shows a single sequence that was kept in the bin and each red triangle shows a sequences that was removed. The labels on the x and y-axes show how much of the variation in the distance matrix is explained by the first and second principal coordinates.

3.3.2 Evaluating the benefit of using the PID approach

The two main benefits of the PID approach are that it allows one to:

1. Correct haplotype frequencies by normalizing for the skewed/biased amplification of certain templates, introduced by PCR.
2. Correct sequencing error by exploiting PCR resampling. Creating a consensus sequence from the reads in a given bin corrects for non-systematic sequencing errors, with the assumption that the differences between the reads are the result of sequencing and PCR error.

An additional benefit of the primer ID approach is that it reduces the number of sequences in the analysis dataset to a manageable number. This is particularly important when considering the sequence alignment steps, which are typically very slow for large numbers of sequences. While read mappers such as Bowtie 2 (Langmead, Trapnell, Pop, & Salzberg, 2009) and BWA (Li & Durbin, 2009) are able to perform reference alignments at considerable speed, certain highly variable sequences, such as those from viral templates, do not map well to a single reference sequence, making a multiple sequence alignment preferable.

To evaluate the PID approach, the differences between the PID and non-PID versions of the 6wpi and 193wpi datasets were compared. Figure 35 shows 75 randomly selected sequences from the PID and non-PID versions of the 193wpi dataset. A noticeably greater level of variation is visible in the non-PID

sequences. There was some variation in the sequences from the PID version, but the variation is localized in specific positions (for example position 220 in the figure) while the variation in the sequences of the non-PID version occurs across all positions, as one would expect for stochastic mutations such as sequencing error.

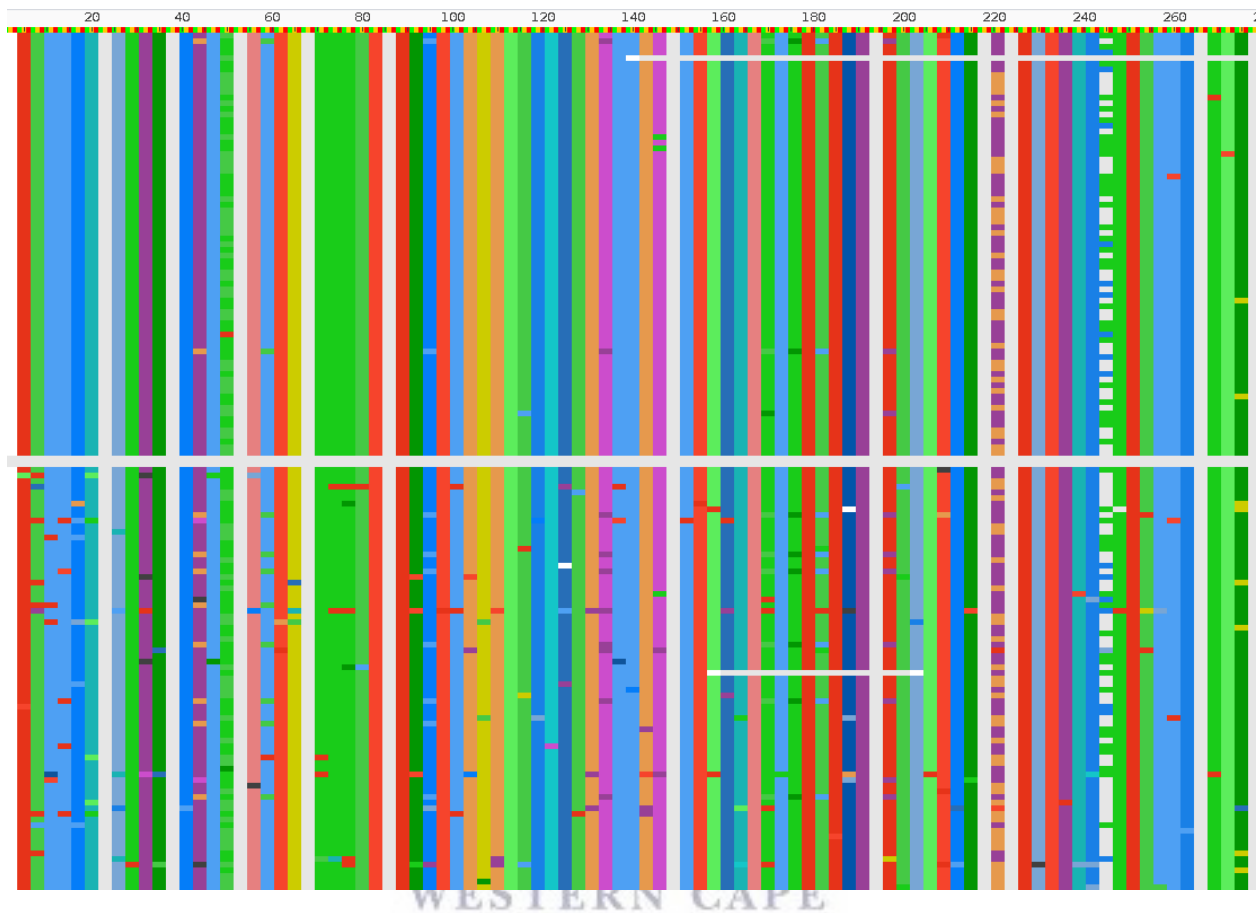


Figure 35: Comparison of 75 randomly selected consensus sequences from the PID version of the 193wpi dataset (above horizontal white bar) and 75 randomly selected sequences from the non-PID version of the 193wpi dataset (below the horizontal white bar). Positions at the top are for nucleotides while the sequences are colored by amino acid. Gaps are white.

The differences visible in Figure 35 between the two versions of the datasets need to be quantified and interpreted with reference to bias introduced from PCR amplification and sequencing error. A number of different comparisons and analyses were performed and used to illustrate the effect of the PID approach. The following were considered:

- The number of sequences per PID bin (direct observation of PCR induced amplification bias)
- The distribution of nucleotides at each position in the alignment of all the sequences (the amount of variation per position)
- The frequencies of the sequences and consensus sequences (the amount of variation per read)

While the results for the 193wpi dataset are shown here, similar results were found for the 6wpi dataset. None of the comparisons and analyses of the datasets can directly measure the sequencing error, since it is confounded by the PCR bias and chimerism. To directly investigate sequencing error we carried out a simulation, which also explored the effects of varying levels of chimerism on the ability to accurately construct a consensus sequence.

3.3.2.1 Adjusting for the number of sequences per PID bin

The effect of PCR bias can most clearly be demonstrated by looking at the number of sequences produced for each PID. Since each bin theoretically represents a single input cDNA molecule, differences in the number of sequences in each bin reflect the level of biased amplification of templates during PCR. This is a direct quantification of the PCR bias.

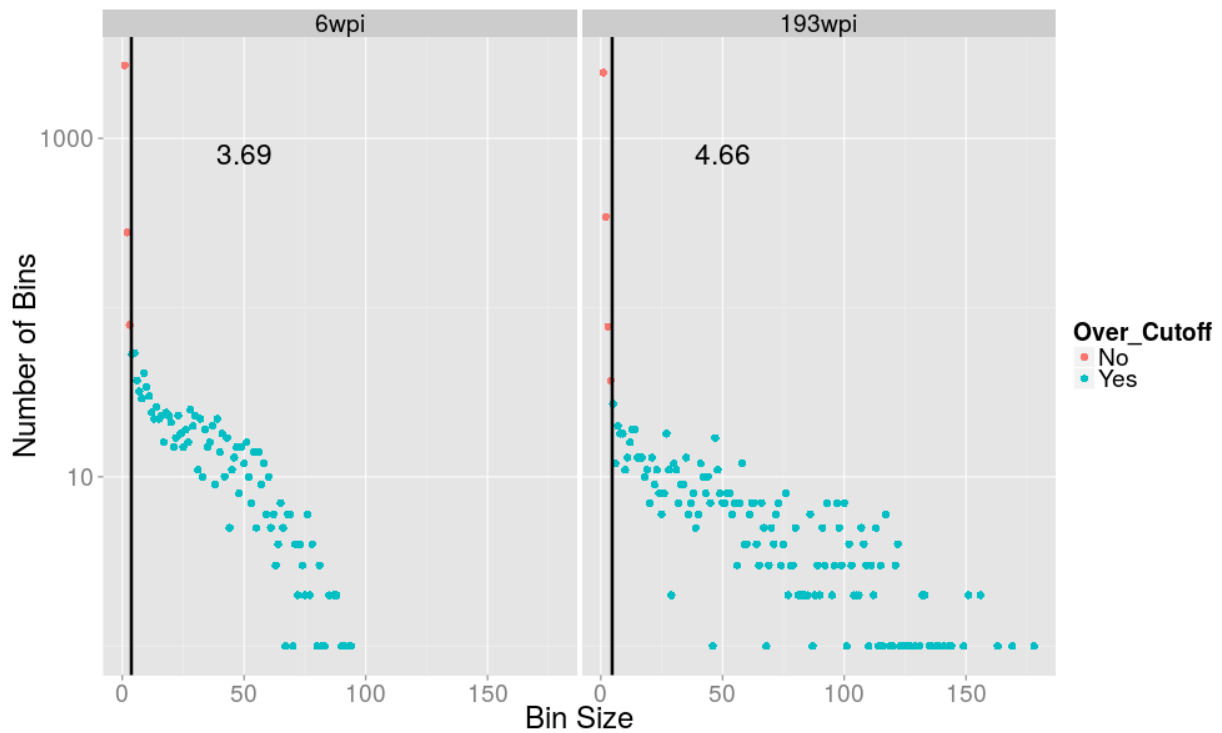


Figure 36: Quantification of the bias caused by PCR. The figure shows the number of bins (y-axis) of a given size (x-axis) there were in the two datasets.

Figure 36 shows the distribution of different bin sizes for the two datasets. Assuming no PCR bias, each plot would only have a single data point, as each input template would have been amplified equally during PCR. This figure shows how prominent biased amplification during PCR is. For example, in the 193wpi dataset, if we were not using the PID method, we would count each sequence in a bin of size 178 as a separate sequence, instead of collapsing them all into a single consensus sequence. This has the potential to introduce artifacts when carrying out analysis based on variant frequencies.

3.3.2.2 Reduction in variability in the distribution of nucleotides at each position in the alignment

Both the higher sequencing error rate and the skewing of the sequence frequencies can be observed when looking at the mismatches between the nucleotide composition at each position of the multiple sequence alignment of all the sequences in the PID version of the dataset and the non-PID version of the datasets. A mismatch is computed for a single position by constructing a vector in which each element represents the percentage of sequences that have the nucleotide associated with that element at the current position. Two such vectors are computed, one for the PID version of the dataset and one for the non-PID version of the dataset. The two vectors are then subtracted, element-wise, from each other and the absolute value of each element is taken. All the elements are then added together to derive the proportion of sequences that differ between the two datasets. For example, consider the case where, in the PID version of the dataset, 95% of nucleotides, at a given position, are A, with 0% C, 0% G, 4% T and 1% being a gap. Assume that in the non-PID version, these percentages are 90%, 0%, 0%, 10% and 0% respectively. Following the above procedure, the difference for this position will be:

$$|(0.95 \ 0 \ 0 \ 0.04 \ 0.01) - (0.9 \ 0 \ 0 \ 0.1 \ 0)| = (0.05 \ 0 \ 0 \ 0.06 \ 0.01)$$

Yielding a total of 0.12 when adding the elements of the vector together. Thus, in this example, 12 % of the sequences differed between the two versions of the dataset at this position. This value is computed for each position, providing an indication of how the two datasets differ across the length of the sequence (Figure 37). From inspection of this figure, one notes that there are a significant number of mismatches over large portions of the dataset.

Multiple sequence alignments can contain low quality regions in and adjacent to areas with a high number of gap insertions. Hence it is important to verify that the regions with a large frequency of mismatches were not driven by gap-rich regions of the alignment. Hence, Figure 37 also shows the proportion of gaps in the alignment. There were a number of frameshift mutations in the non-PID dataset, mostly due to homopolymer errors. Thus, to maintain the correct reading frame, the alignment contains a very high proportion of three nucleotide wide gaps (>99%) over these regions. As a result, the percentage mismatch between the datasets in these, predominantly gap, positions is much lower than in other regions of the alignment. For example, the percentage of sequences that differ between the PID and non-PID datasets at positions where gaps occur at 95% or more is only 0.0105%, while at positions which contain gaps at a frequency of 5% or less the percentage is 2.878%. This is a result of the low rate of indel errors during the sequencing approach yielding a small number of sequences with these frameshift errors. During the alignment step, gaps were inserted in all the

sequences which do not have frameshift errors yielding these regions in which the two datasets are highly similar.

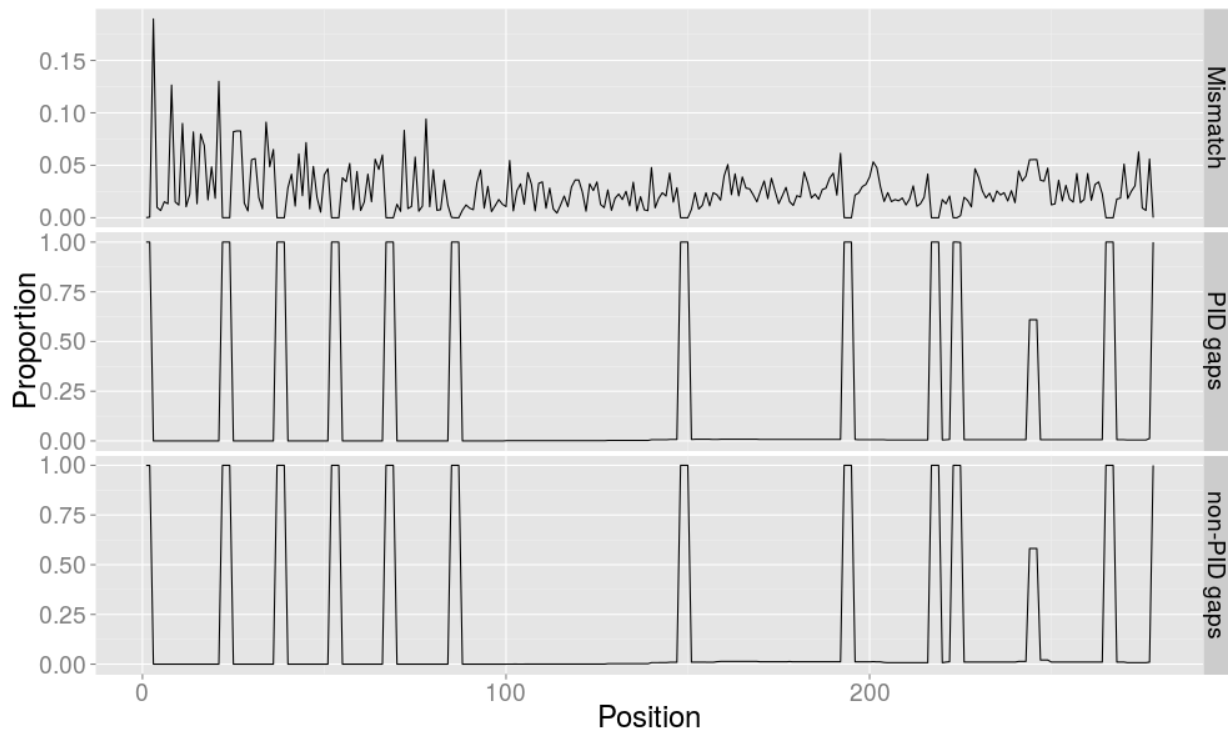


Figure 37: The proportion of sequences with mismatching nucleotides at each position between the non-PID and PID versions of the 193wpi dataset (top pane). The middle and bottom panes shows the percentage of sequences with gaps by position for the non-PID and PID versions of the dataset respectively. Note that the y-axis for the top pane is different from the y-axis of the middle and bottom panes.

The degree of mismatch between the positional nucleotide frequencies for the two datasets demonstrated in Figure 37 is an artifact of both a higher rate sequencing in the non-PID dataset and the uncorrected PCR induced bias. To distinguish between the two sources of error, Figure 38 shows the proportion of sequences with mismatching nucleotides at each position between the non-PID and PID datasets in the context of the variability observed in the two datasets. The middle pane shows the percentage of sequences in PID version of the dataset that does not match the consensus sequence of all the sequences in the PID version of the dataset by position. The bottom pane shows the same measure on the non-PID dataset. The peaks represent a mutation away from the consensus of the dataset that occurred consistently. The higher peaks in the middle pane are also present in the bottom pane and are of similar heights implying that the more frequently observed mutations can be seen with both datasets. However, the portions between the peaks are much smoother in the PID dataset allowing one to observe low peaks which are indistinguishable from the noise generated by the sequencing error in the non-PID version of the dataset.

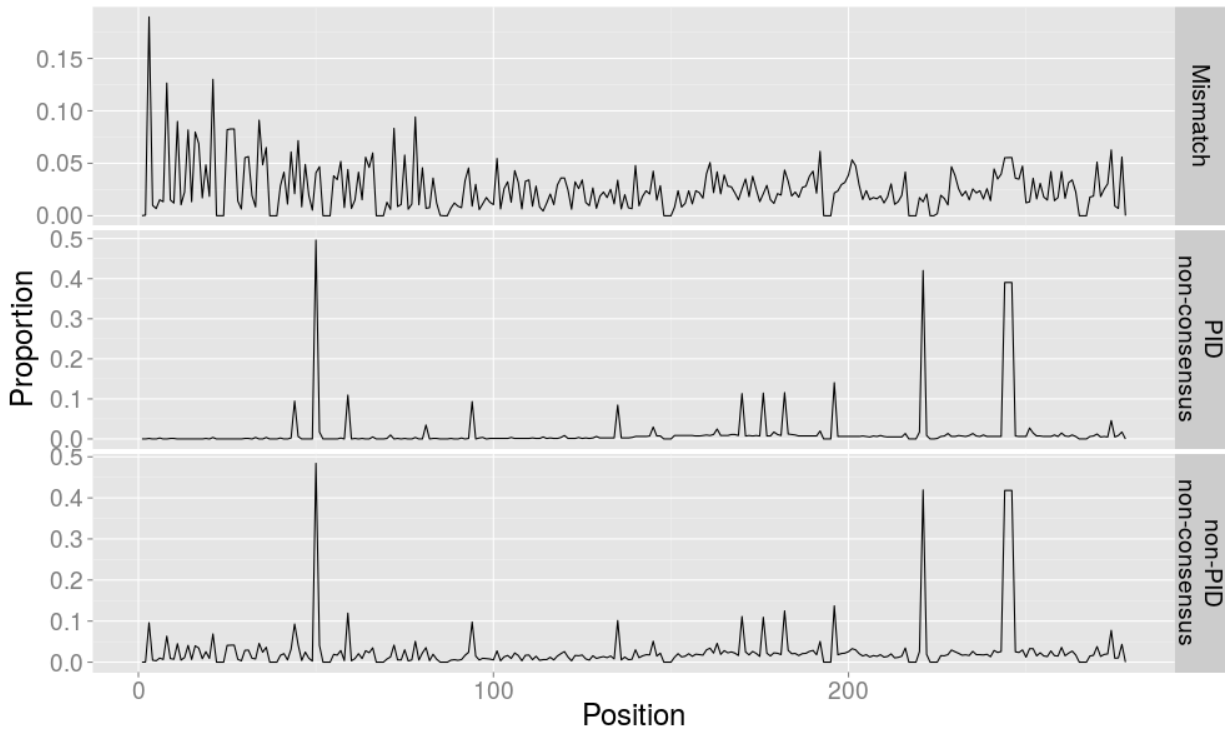


Figure 38: The proportion of sequences with mismatching nucleotides at each position between the non-PID and PID versions of the 193wpi dataset (top pane). The middle and bottom panes shows the percentage of sequences that differ with the consensus sequence of the dataset by position for the non-PID and PID versions of the dataset respectively. Note that the y-axis for the top pane is different from the y-axis of the middle and bottom panes.

The higher rates of mismatches visible on the left hand side of the top pane of Figure 38 is due to the higher variability in the non-PID dataset which can be observed in the bottom pane of Figure 38. The higher variability is caused by the higher rates of sequencing error at the end of reads. Figure 39 shows the qualities of the reads in the dataset by position, clearly showing that the regions with higher mismatch between the non-PID and PID versions of the dataset also have lower quality scores. The lower quality at the ends of the reads did not lead to increased variability at the corresponding positions of the PID dataset.

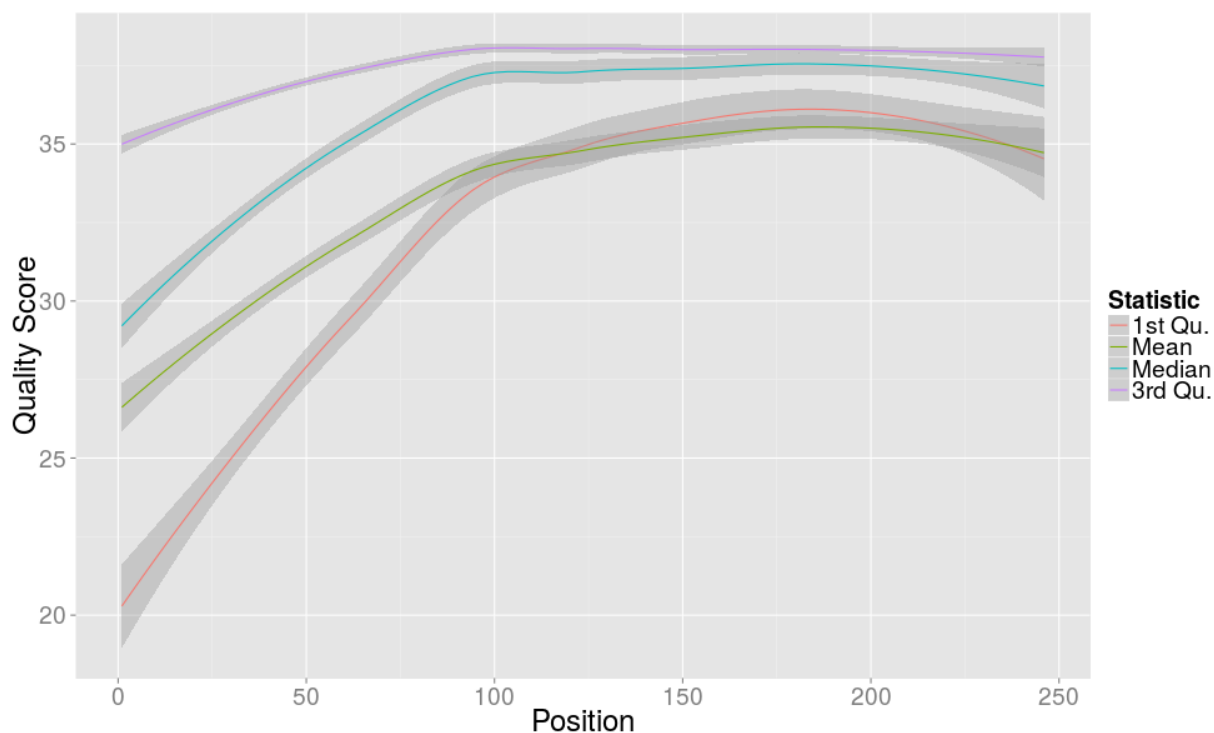


Figure 39: The sequencing quality of the reads in the 193wpi dataset by position. The reads shown in this figure are the reverse reads meaning that their ends are at the left hand side of the figure.

3.3.2.3 The frequencies of the sequences and consensus sequences

Instead of analyzing the variability in the data by the positions in the alignment, the variability can also be explored by counting the number of unique sequences and the frequencies and prevalence of the most abundant sequences. When looking at the top 10 most frequent sequences in the PID and non-PID versions of the 6wpi datasets, as shown in Table 17, one notes that the percentage of the dataset that consists of the most frequent sequences varies significantly between the two dataset versions. The most frequent sequence in the PID version accounts for 34.58% of the dataset, while the most frequent sequence in the non-PID version only accounts for 7.08% of the dataset. In contrast, when looking at the rank of the frequencies (giving the most frequently occurring sequence rank 1) for the top 10 most frequent sequences, the most frequent sequences in the PID version of the dataset tended to also be the most frequent sequences in the non-PID version of the dataset. Importantly, the most frequent sequence was the same in the two versions of the dataset. Likewise, the second most frequent sequence was also the same for the two dataset versions. There were differences for lower ranked variants, with the 3rd most frequent sequence in the PID version of the dataset being ranked 5th in the non-PID version of the dataset.

Table 17: The frequency, percentage of the sample and rank of the top 10 most frequent sequences in the 193wpi PID dataset and those metrics for the same sequences on the non-PID dataset.

non-PID dataset			PID dataset		
Freq.	Sample %	Rank	Freq.	Sample %	Rank
2972	7.08%	1	278	34.58%	1
1702	4.06%	2	143	17.79%	2
364	0.87%	5	41	5.10%	3
273	0.65%	7	24	2.99%	4
551	1.31%	3	23	2.86%	5
312	0.74%	6	20	2.49%	6
177	0.42%	8	16	1.99%	7
92	0.22%	12	9	1.12%	8
64	0.15%	18	6	0.75%	10
110	0.26%	11	6	0.75%	10
66	0.16%	15.5	6	0.75%	10

Instead of looking at the top 10 most frequent sequences in the PID version of the dataset, as in Table 17, one can also inspect the top 10 most frequent sequences in the non-PID version of the dataset listed in Table 18. The 9th most frequent sequence in the non-PID version of the dataset is not represented at all in the PID version of the dataset. The 4th most frequent sequence in the non-PID version of the dataset is one of the least frequent sequences in the PID version of the dataset (rank 25.5 out of 112, it occurred only three times).

Table 18: The frequency, percentage of the sample and rank of the top 10 most frequent sequence in the 193wpi non-PID dataset and those metrics for the same sequences on the PID dataset.

non-PID dataset			PID dataset		
Freq.	Sample %	Rank	Freq.	Sample %	Rank
2972	7.08%	1	278	34.58%	1
1702	4.06%	2	143	17.79%	2
551	1.31%	3	23	2.86%	5
377	0.90%	4	3	0.37%	25.5
364	0.87%	5	41	5.10%	3
312	0.74%	6	20	2.49%	6
273	0.65%	7	24	2.99%	4
177	0.42%	8	16	1.99%	7
176	0.42%	9	-	-	-
140	0.33%	10	5	0.62%	13

While Table 17 and Table 18 only consider the most frequently occurring sequences, Figure 40 considers all sequences that occur in both versions of the dataset. Figure 40 A shows the distribution of the ranks assigned to the PID sequences. The large number of ties (sequences with the same rank) result from a large number of sequences occurring at the same frequencies, specifically, the large number of sequences occurring only 1, 2 or 3 times, as shown by the three high bars. Figure 40 B reinforces the observation that for the most frequent sequences, the ranks correlate well (for those sequences occurring in both datasets). However, at lower ranks the correlation is weaker as can be seen the large number of different ranks assigned to the sequences in the non-PID dataset which all occurred only once in the PID dataset (the height and spread of the points in the right most 'column' in Figure 40 B).

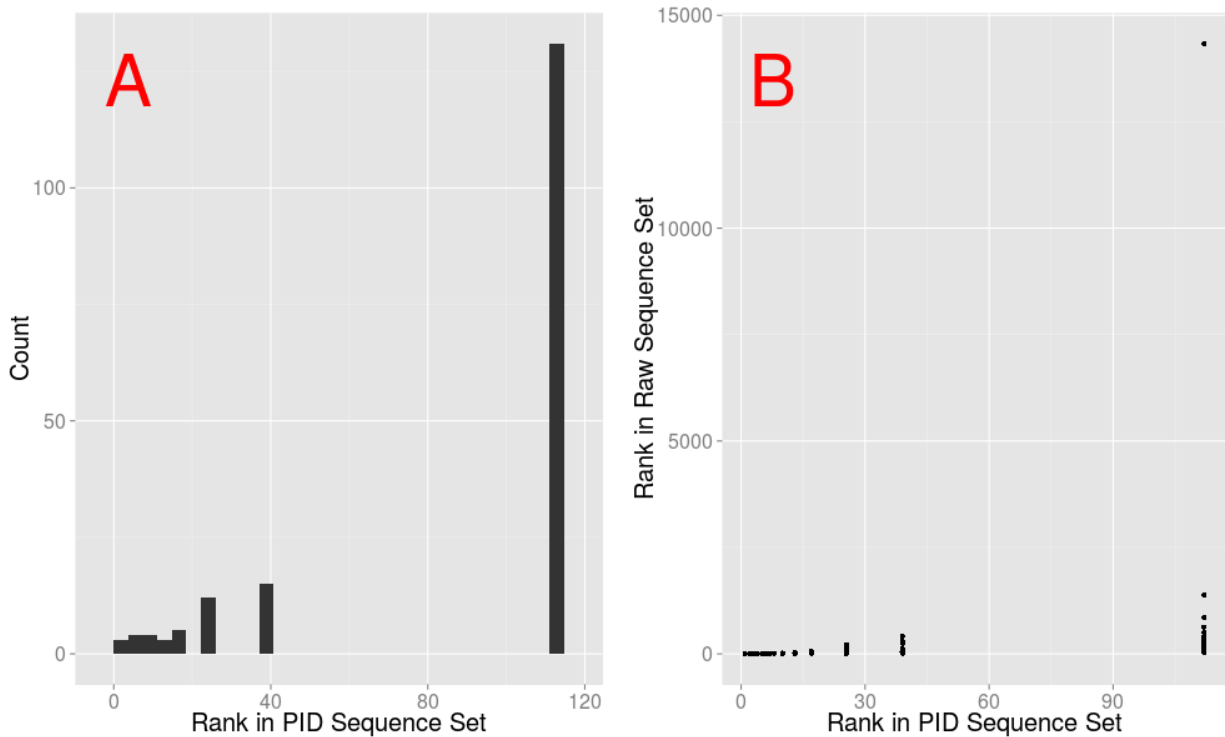


Figure 40: A) A histogram showing the ranks in the PID version of the 193wpi dataset and B) a scatter plot of the ranks assigned to sequences in the PID version of the dataset (x-axis) to the ranks assigned to those same sequences in the non-PID version of the dataset (y-axis).

While the most frequently occurring sequences can be identified in the non-PID version of the dataset, the prevalence at which the most frequently occurring variants occur differ significantly from that reported by the PID version of the dataset. This discrepancy can be explained by the higher rate of sequencing errors in the non-PID version of the dataset. For a sequence of length 245 (the length of the sequences in the 193wpi dataset with gaps removed) and a sequencing error rate of $\frac{1}{100}$ a sequence has only an 8.5% of being sequenced without a sequencing error. Hence the majority of the sequences in the non-PID dataset are sequences with sequencing errors diluting the prevalence of the correctly sequenced sequences. Figure 41 shows the distributions of the distances between the sequences and the consensus sequence for each of the two versions of the 193wpi datasets. The larger average distance observed in the non-PID version (7.56 vs. 4.20) is another side-effect of the larger sequencing error rate.

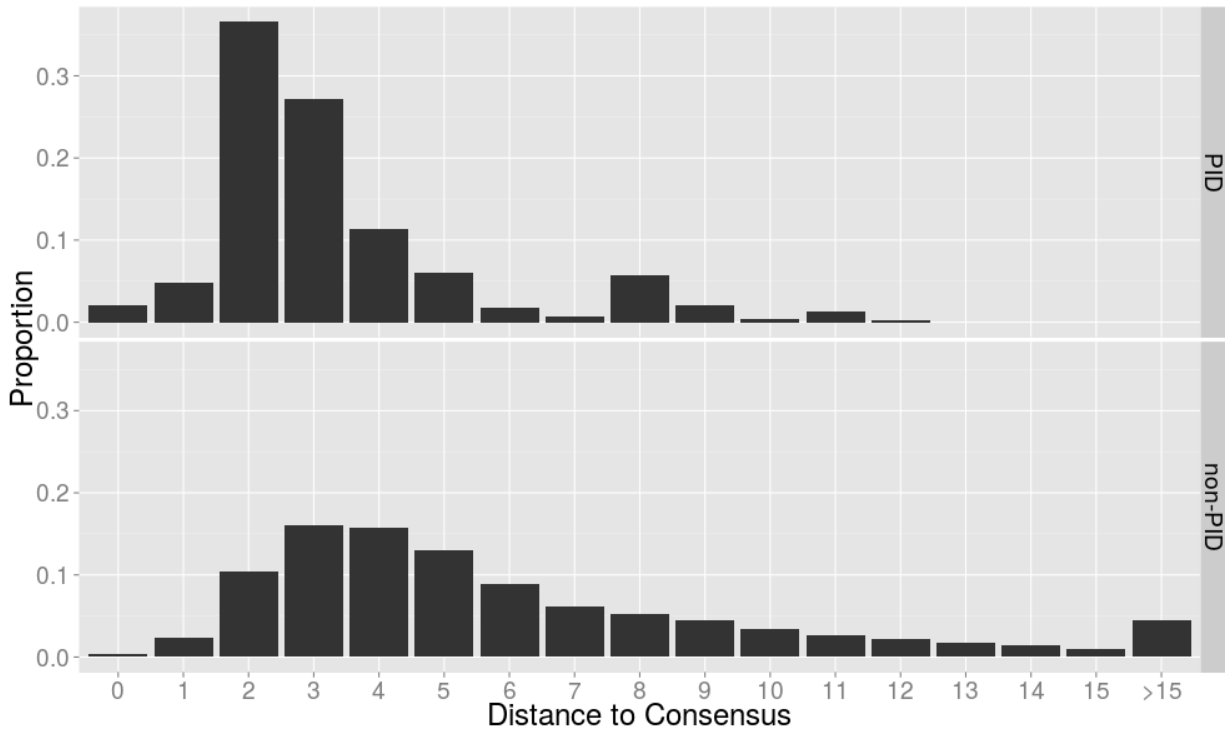


Figure 41: Histograms of the distance between sequences to the consensus sequence for the PID and non-PID versions of the 193wpi dataset. The consensus sequence was constructed for each version of the dataset. The restricted Levenshtein or edit distance between the consensus and each sequences in the two versions of the datasets. Each bar shows the proportion of sequences that was the distance represented by the bar from the consensus sequence. The last bar on the right hand side aggregates all distances larger than 15.

The distances between the sequences and the consensus for the two versions of the 193wpi dataset were caused by both the sequencing error and the variability in the viral population. In a dataset with a homogenous population, the distances in the PID version of the dataset were close to zero while the distances in the non-PID version remained large due to the higher sequencing error. The 6wpi dataset has much lower viral diversity, Figure 23, hence it exhibits these features as can be seen in Figure 42. The relative difference in average distance between the non-PID and PID were larger in the 6wpi dataset (2.929 vs. 0.129).

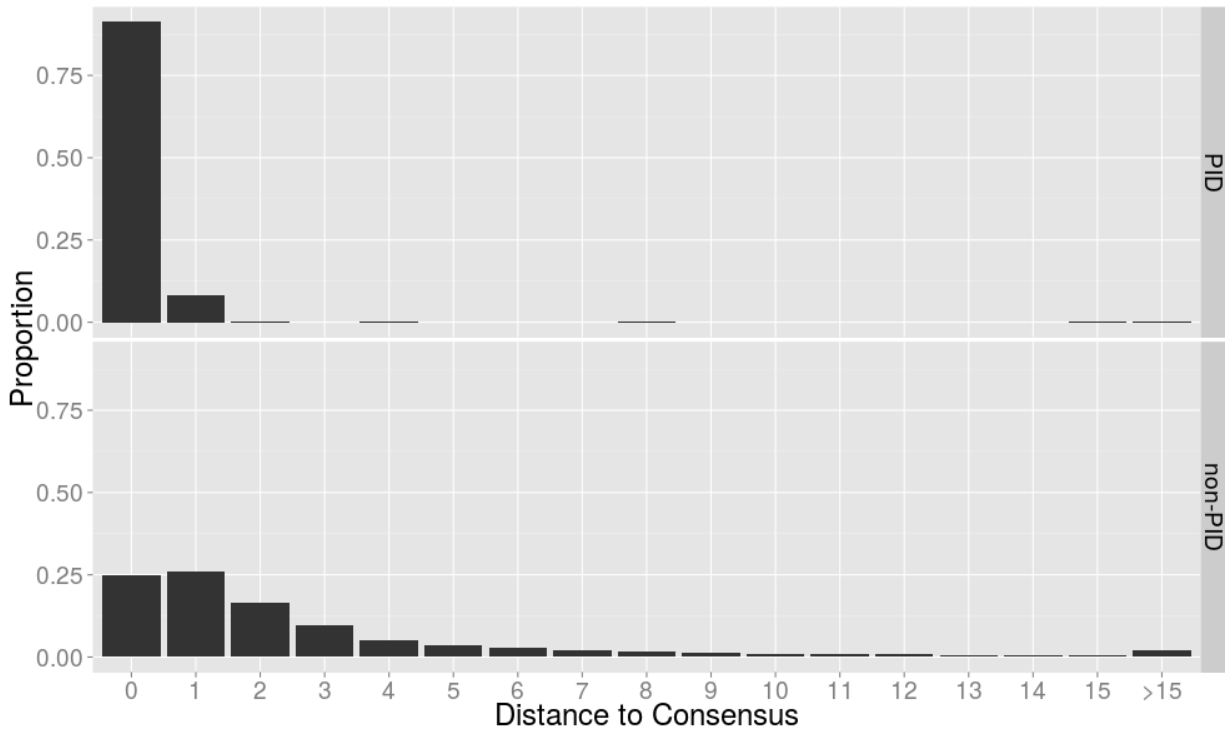


Figure 42: Histograms of the distance between sequences to the consensus sequence for the PID and non-PID versions of the 6wpi dataset. The consensus sequence was constructed for each version of the dataset. The restricted Levenshtein or edit distance between the consensus and each sequences in the two versions of the datasets. Each bar shows the proportion of sequences that was the distance represented by the bar from the consensus sequence. The last bar on the right hand side aggregates all distances larger than 15.

3.3.2.4 Simulations of the effect of the PID approach on sequencing errors

Following a similar approach as described in section 3.3.1.3.3 to investigate the rates of chimerism in the bins, a simulation was performed to assess the effect of the PID approach on sequencing errors. The simulations used the bin sizes and consensus sequences of the 193wpi data and simulated each bin in the dataset 30 times using rates of chimerism of 0%, 20%, 40% and 50%. The sequencing error rate was estimated by comparing the original sequences in the bin to the consensus sequence for the bin. In cases where the consensus sequence of a bin in the dataset had degeneracies, the degenerate nucleotides were replaced with a random nucleotide before simulation sequences from it. For chimeric bins, another bin was selected at random and some of the reads were generated from the other bin's consensus sequence, irrespective of the differences between the consensus sequences of the two bins.

Using this simulated dataset, the consensus sequences constructed from the simulated bins were compared to the target consensus sequence(s) (in cases where the chimerism rate was larger than 0%, two input consensus sequences were used) from which the bin was simulated. The distance between the constructed consensus and the target consensus as well as the number of degeneracies

in the consensus sequences are shown in Figure 43. In the case of no chimerism, the target consensus is reconstructed reliably. When 20% or 40% of the sequences in the bins come from another template, the number of degeneracies in the constructed consensus sequences remain low, but a number of the constructed consensus sequences are different from their target consensus sequences. The edit distance between the constructed and target consensus sequences tended to be between 3 and 5 over the length of the sequences, similar to the distances between the consensus sequences in the 193wpi dataset. When the bins were 50% chimeric, large numbers of degeneracies occurred in the constructed consensus sequences, leading to larger edit distances between the constructed and target consensus sequences.

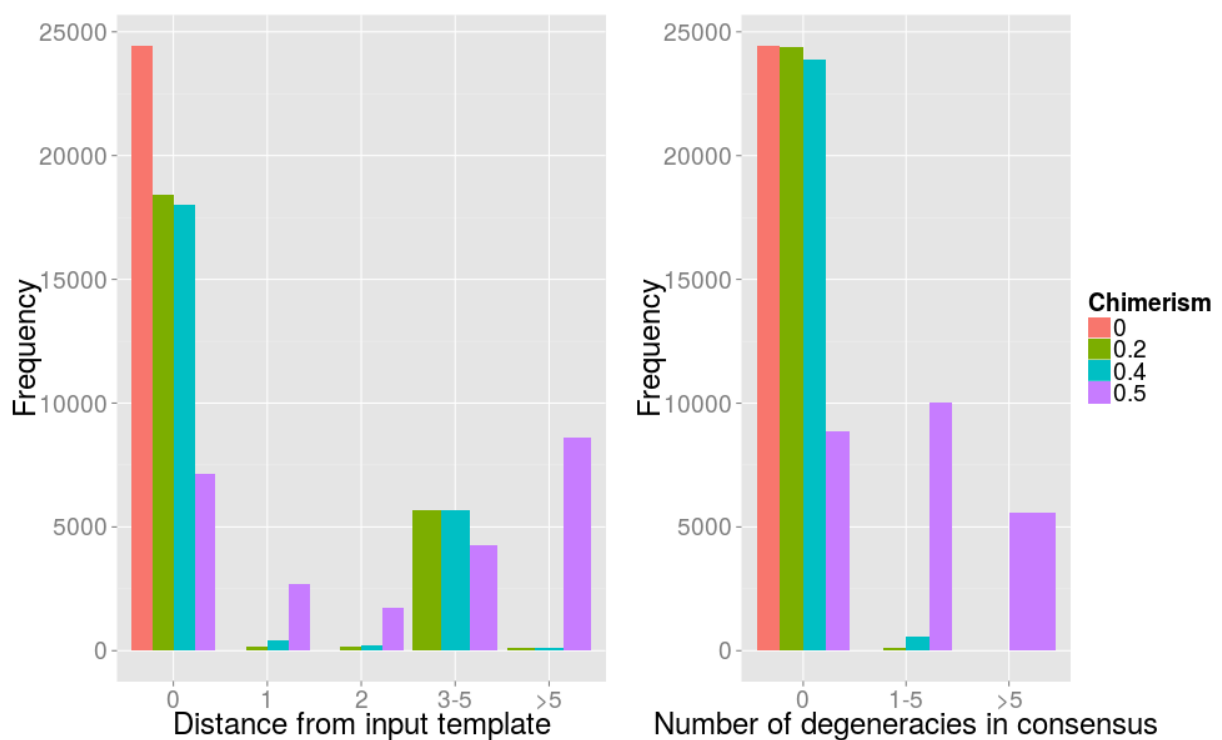


Figure 43: The effect of the primer ID approach on sequencing error in the presence of chimerism. A simulation was used to generate reads from an input sequence. A consensus sequence was constructed from the simulated reads. The distance between the consensus sequence constructed from the simulated reads and the input template as well as the number of degeneracies in the consensus sequence is shown in the histograms. Chimerism was simulated by generating a portion of the reads for a bin from a second input template. Bin sizes, sequencing error rates and the pool of input templates were chosen so that the simulation is similar to the 193wpi dataset.

3.3.2.5 Summary

To summarize, the PID approach highlights several issues related to the frequency distribution of variants and the sequencing error rates in a dataset:

1. The number of sequences produced for a single input template follows a distribution with significant variance. In the examples shown here, there was a single input template that was sequenced 178 times, while many other templates were sequenced only a very small number of times (<10).
2. When looking at the variability of the sequences by position, both datasets highlight the same positions that are highly variable. However, the higher levels of noise in the non-PID version of the dataset resulting from high levels of sequencing error mean that positions with lower levels of variability cannot be identified as clearly as in the PID version of the dataset.
3. While those sequences that are most frequent in the PID version of the dataset tend to be most frequent in the non-PID version of the dataset also, the converse is not true. There are sequences that are frequent in the non-PID version of the dataset that either do not occur in the PID version of the dataset, or occur at a very low frequency.
4. The proportion of specific variants in the PID and non-PID datasets also differs. The most frequent sequence in the PID version of the dataset comprises 34.58% of the entire dataset, while the most frequent sequence in the non-PID version of the dataset accounts for only 7.08% of the dataset. This difference is driven by the higher level of sequencing error in the non-PID dataset, leading to a large number of sequences falsely being identified as unique variants. Without using the PID approach, it becomes much harder to differentiate between legitimate variants and spurious variants caused by sequencing errors.

3.4 Conclusions and future work

Deep sequencing of diverse populations is a useful technique, allowing one to investigate complex phenomena like the presence and frequency of drug resistant variants (Jabara et al., 2011), as well as measuring the diversity of microbial species in various biological samples (Lundberg et al., 2013). However, when sequencing a population with a high level of diversity, differentiating between sequencing error and real variation becomes problematic. The result of this is that the cutoff frequency for detecting low abundance polymorphisms is limited by the level of sequencing error rate. Additionally, skewed amplification of certain templates during PCR confounds the measurement of the frequency distribution of different variants.

The PID approach (Jabara et al., 2011) has been proposed to address these problems. By tagging each input template with a unique nucleotide sequence (PID), all amplicons derived from a specific template should share the same PID tag. The resampling of the initial template can be used to correct for non-systematic sequencing errors, as taking the consensus base at each position should correct for these errors. At the same time, the PCR skewing problem is also addressed, since collapsing all

sequences with the same PID into one sequence, corrects for the skewed amplification of certain templates.

However, this technique introduces a number of difficulties for the data processing step, requiring researchers who wish to use this method to have access to specialized bioinformatics tools. In this chapter, various components required to process sequence data using the PID approach were carefully investigated. A robust and easy to use solution to these problems was implemented in an R package called *MotifBinner*.

Additionally, the effects of the PID approach on the dataset produced was also investigated. When searching for the PID in the input sequences, we showed that using a fuzzy search result in the utilization of a greater number of these input sequences to construct the final consensus sequences. Future work should extend the fuzzy matching to also allow indels in the search motifs.

As shown in (S. Zhou et al., 2015), sequencing errors in the PID itself results in large number of spurious PIDs which occur at a low frequency. The maximum expected size of such spurious bins is computed using a simulation approach. This maximum expected size is used as a cutoff and all bins smaller than this cutoff are discarded. An advance on this would be to compare the PIDs of the small potentially spurious bins to the PIDs of larger bins. Smaller bins with PIDs that are significantly different from the PIDs of all the large bins are unlikely to be the result of sequencing errors in a PID and can theoretically be included in the final dataset. Additionally, the sequences that form supposed offspring bins, can be compared to the consensus sequence of the parent bin with the closest matching PID tag. If the distance between the PID tags, as well as the sequence – consensus sequence distance, are low, it is highly likely that the offspring bin was produced from this parent bin.

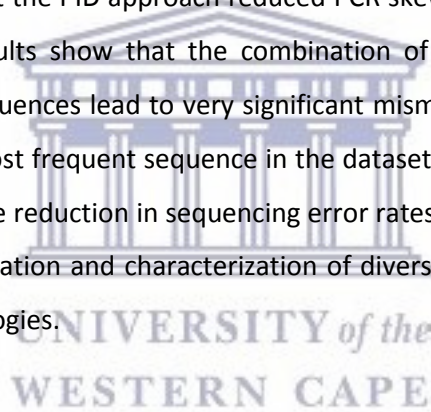
For various reasons different input templates can be assigned the same PID. This results in PID collisions creating chimeric bins. By assuming that the variation in a bin is the result of a single input template and sequencing error, a distribution was derived for the number of non-consensus letters at any position in a bin. If the bin deviates significantly from this distribution, then the bin is probably chimeric. This technique revealed a flaw in the trimming of one of the example datasets used in this thesis. The proportion of bins labelled chimeric in real datasets are still abnormally large. This is potentially due to sequencing error hotspots and future work should address this. Potential future solutions are to include an additional parameter that specifies how non-uniformly sequencing errors may be distributed along the positions in the bin, or to extend the test to look at more positions in the bin simultaneously. After completion of this work, we learned that recombination during the PCR step can chimeric sequences in which a portion of the sequence does not match the template that was

originally tagged with the PID of the chimeric sequence. This phenomenon is explored in the next chapter.

Having sequences with incorrect PIDs in a bin may lead to complications in the alignment and consensus generation steps. This is combatted by iteratively removing the most outlying sequence in each bin until the maximum distance between the furthest two sequences in the bin falls below a threshold. After removing such outlying sequences, alignment is simple and a `muscle` alignment using default parameters performs adequately. The multiple sequence alignment of each bin is then collapsed into a consensus sequence and these consensus sequences are reported as the final dataset.

`MotifBinner` produces a comprehensive report for each run, as well as detailed information about each bin. This allows for easy exploration of the binning process and informs the user of potential pitfalls in the dataset. The report also tracks the exact input parameters used leading to improved reproducibility.

Lastly, it was directly shown that the PID approach reduced PCR skewing by quantifying the number of sequences per PID. The results show that the combination of PCR skewing and uncorrected sequencing error in the raw sequences lead to very significant mismatches in the proportion of the dataset that results from the most frequent sequence in the dataset. It has been demonstrated that the PID approach leads to a large reduction in sequencing error rates (S. Zhou et al., 2015), making it an essential tool for the quantitation and characterization of diverse populations, when using next generations sequencing technologies.



4 Protocol Optimization for PID sample preparation

We sought to characterize the PID approach and test various strategies that address the issue of recombination introduced during PCR. Our goals were to measure the ability of the PID approach to reduce the effect of sequencing error, to detect minority variants in viral quasispecies, to measure the rates of PCR induced recombination and explore factors that affect PCR induced recombination.

Several techniques exist for reducing PCR induced recombination. (Thompson, Marcelino, & Polz, 2002) recommends reconditioning the reaction by diluting the amplified reaction into a fresh reaction mixture and running another three cycles of PCR. Stochastic events during PCR leads to biased amplification and can be offset by partitioning the reaction, running the partitions in parallel and then pooling the resulting fluid (Wagner et al., 1994). PCR recombination occurs primarily when partially amplified product from earlier cycles act as primers for different input templates in subsequent cycles leading to the creating of chimeric sequences. Increasing the duration of the elongation cycle reduces the amount of partial product, leading to lower recombination rates. The majority of PCR recombination occurs when the DNA concentration in the PCR reaction is high. Thus limiting the number of cycles is another strategy for reducing PCR recombination (Kanagawa, 2003; J. Liu et al., 2014).

We produced reference samples with known variants at known frequencies to study the effectiveness of increasing PCR elongation time, decreasing the number of PCR cycles, and sample partitioning, by means of dPCR (droplet PCR), on PCR induced recombination. Furthermore, we sought to characterize the error rates of the PID approach, its ability to detect minor variants and to estimate the prevalence of the variants.

4.1 Methods

This work made use of 40 datasets produced by sequencing 5 different artificially constructed samples using 8 different library preparation protocols. The samples were constructed by mixing molecules from 3 unique HIV-1 viruses together in different ratios. These three sequences, referred to as CAP63, CAP210 and CAP239, were obtained from the CAPRISA 002 Acute Infection study and infectious molecular clones were constructed using the technique described in (Ochsenbauer et al., 2012). Infectious virus stocks were generated by transfecting plasmids and harvesting the virus containing supernatant. The viral loads were determined twice for each virus stock and based on those viral loads, the viral stocks were pooled at the target ratios of 33/33/33; 5/85/10; 76.7/20/3.3; 1/3.3/95.7; and 98.7/0.33/1. Due to the inaccuracy of viral loads, the pooling was imperfect meaning that the true ratios of the samples are unknown.

The viral RNA was converted to cDNA using the primer GCCTTGCCACACGCTCAGNNNNNNNNTGTGTTGTAAYTTCTAGRTC. This primer is designed to bind to HXB2 position 1102 to 1121 of the gp160 protein, have a randomly assigned 9-mer region that forms the PID and a universal primer binding site for use in subsequent PCR steps.

First round PCR was performed using a reverse primer binding to the universal binding site introduced during the reverse transcription step and a forward primer with a gene specific region targeted to HXB2 position 661 to 683 of the gp160 gene (TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGNNNNNNNGCTGGTTATGCGATTCTAAAGTG). The primers for the second round of PCR includes the required sequencing platform specific regions as described in (S. Zhou et al., 2015) so that the final product after PCR consisted of positions 684 to 1101 of the gp160 gene and the relevant sequencing platform specific primers. Droplet PCR was performed in an emulsion made with Biorad's droplet generator. The emulsion was broken down using chloroform and then purified using SPRIselect beads.

Paired-end sequencing was performed on the PCR product yielding two reads from each end of the molecule. The first read, called the forward read, starts from position 684 and extends towards the 3' end up to approximately position 959 of gp160 in HXB2. The second read, called the reverse read, starts on the 3' end, with the region introduced during cDNA synthesis – the binding site for the PCR primers and then the PID. The PID is followed on the 5' end by the gene specific region of the primer sequence that matches HXB2 gp160 from positions 1121 to position 1102. This is followed by the sequence of the target molecule up to approximately position 831. This results in an overlap between the forward and reverse reads between positions 831 and 959. The eight different library preparation protocols are described in Table 29 (duplicate of Table 21).

Table 19: The eight different protocols used to prepare the samples for sequencing.

	Round 1			Round 2	
	Elongation Time	PCR Type	Number of cycles	PCR Type	Number of cycles
nrPCR (normal PCR)	2.5min	normal	25	normal	25
rcPCR (reduced cycle PCR in second round)	2.5min	normal	25	normal	15
2dPCR (both rounds amplified with dPCR)	2.5min	droplet	25	droplet	25
mxPCR (first round dPCR, second round normal PCR)	2.5min	droplet	25	normal	25
nrPCR (normal PCR)	10min	normal	25	normal	25
rcPCR (reduced cycle PCR in second round)	10min	normal	25	normal	15
2dPCR (both rounds amplified with dPCR)	10min	droplet	25	droplet	25
mxPCR (first round dPCR, second round normal PCR)	10min	droplet	25	normal	25

4.1.1.1 Processing Raw Sequences

The sequencing process is error-prone, introducing significant noise into the data. Many errors are large and obvious, implying that some basic processing can significantly improve the quality of the data. When the sequencer cannot call a base with sufficient certainty, it inserts an N at that position. All sequences in which more than 2% of the sequence were Ns were removed.

The Illumina sequencing process sequences by synthesizing the complementary strand to a sequence that is hybridized onto a slide. Special tags are used to ensure that all sequences are synthesized at the same rate. If a sequence is not the standard length, then either a short molecule was hybridized to the slide or the system that ensures synchronized synthesis failed. Either of these cases indicates a significant failure in the system, justifying the elimination of such reads. However, we keep sequences as short as 295 base pairs, since if the process was compromised only at the very end, then the merging of the paired-end reads might correct the error.

The base call for a single position for a single sequence is performed by comparing the amount of light emitted at the spot on the slide to which that sequence was hybridized during each of the four synthesis rounds (one round for each base: A, C, G and T). If one base clearly emitted more light than the others, then the call is made with high certainty and a high quality score is assigned. As the amount of light emitted during the 2 or more of the rounds approaches the same value, a lower quality score is assigned (and in the case of such ambiguity that it is unclear which round was the brightest, a score of zero is assigned and an N is assigned to the position). The number of incorrect base calls in sequences in which the average score across all positions is low (below 20) or in which a number of positions have very low scores (15% of the positions have scores below 10) is likely to be high, thus we discard them. The criteria of 15% of positions having scores below 10 is very lax. Using values of 5% and 15 would suffice in many cases. However, in our data, we have a large overlap between the forward and reverse reads together with a tendency for extremely poor quality at the very end of the reverse read. In some cases the merging process can correct the errors, so we chose to use these lenient parameters. The final step in the process is to merge the sequences using PEAR (J. Zhang, Kobert, Flouri, & Stamatakis, 2014) with default settings.

4.1.2 Generating Consensus Sequences

The PID approach offers another opportunity to reduce the level of noise in the dataset, thus we follow the steps discussed in section 3.1 to further improve the data quality. Briefly, the merged reads were grouped into bins based on the PID detected in their reverse primers. By comparing the PIDs of small bins to the PIDs of large bins, small bins with a high probability of resulting from a sequencing error in a large bin were discarded. Each bin was aligned and a consensus sequence was constructed using the quality scores of the reads in the bin. At each position, the quality scores for each letter that occurred in the reads were tallied and the consensus was called at the letter with the highest sum of quality scores. If the second highest sum was within 35 for the highest sum, an N was inserted in the consensus sequence. Consensus sequences containing Ns were discarded unless otherwise indicated.

4.1.3 Constructing an array of accuracy and quality score tallies

~~Using a four dimensional data structure of nested lists in R, a library of all matches and mismatches between the input templates and merged and consensus sequences was constructed, and quality scores by position was constructed. These matches and mismatches was tracked by position and by the quality score base call and stored in a four dimensional data structure of nested lists in R. Specifically, the dimensions track the position, the base in the sequence, the base in the template sequence and the quality score, with the value in the data structure tallying the number of times the combination of these four features occurred. Two such structures were constructed for each sample, one for the merged sequences and one for the consensus sequences. To populate this data structure,~~

each merged or consensus sequence was pairwise aligned to a consensus of the three input templates in which ambiguity characters were inserted in positions where the templates mismatch. Using a for loop to step through the length of the alignment, at each position at which the consensus of the templates was not an ambiguity character a tally was added to the entry in the data structure that corresponds with what was observed.

4.1.4 Detecting Recombination

Recombination during the PCR process, exacerbated by the long primers needed by the PID approach, introduces many errors (in the form of chimeric sequences) into the dataset. Using the unique IDs assigned to the molecules together with the fact that we know the true sequences we can flag each sequence as either legitimately originating from one of the input sequence or as being the result of sequencing a recombinant sequence. Using the three input clones as templates, a library of all possible detectable recombinants that could result from a single recombination event was constructed. For example, the first three positions where CAP63 and CAP210 differ are positions 22, 24 and 35. Hence the library will contain the constructs:

- 1) CAP63 from position 1 to 22, recombination event at position 23 or 24, then CAP 210 till the end of the sequence.
- 2) CAP63 from position 1 to 24, recombination event between positions 25 and 35, CAP210 till the end of the sequence.
- 3) CAP210 from position 1 to 22, recombination event at position 23 or 24, then CAP63 till the end of the sequence.
- 4) CAP210 from position 1 to 24, recombination event between positions 25 and 35, then CAP63 till the end of the sequence, and so forth.

Each merged sequence and consensus sequence was compared to this library of potential recombinants as well as the three original input sequences. Each read was assigned to the member of the library which was closest to it in terms of edit distance subject to the constraint that at least two positions must be specific to a variant for it to be included in a chimeric sequence. This constraint was designed to limit the number of spurious recombinants introduced due to sequencing error. Using the merged sequences, the consensus sequences and the assignment of each of these to either one of the original templates or a chimera, various properties of the datasets could be easily explored as detailed in Table 46 [in Appendix 8.3](#)

4.2 Results and Discussion

To study different PCR protocols, we generated three infectious molecular clones (IMC) on 293T cells. CAP063, CAP210 and CAP239 vRNA isolated from 293T supernatant was mixed together in 5 different ratios (Table 20) and processed using the 8 PCR variations, listed in Table 21, of the custom amplicon design described in section 4.1. The mixtures were constructed based on initial viral loads of the isolated vRNA. Due to the variability of viral loads the true prevalences of the variants are unknown, thus we cannot compute the bias of the prevalence estimates. In the region spanned by this amplicon, the three variants differ from each other by 57 mutations on average (Table 22) and it includes regions of high homology and high diversity (Figure 44).

Both raw (a raw sequence is a sequence that was minimally processed and merged as described in section 4.1, but not yet processed into consensus sequences with MotifBinner) and consensus sequences were classified as either resulting from one of the three clones or as resulting from a recombination between two clones. Each raw and consensus sequence was compared to each possible unique sequence that can result from a single recombination event between two of the input templates as well as the un-recombined input templates and assigned to the one that it is most similar to. Using this dataset we compared the amplification efficiency, sequencing error rates, prevalence estimates and PCR recombination rates across the different library preparation approaches. We also examined the relationship between sequence homology and the PCR recombination rate.

Table 20: Target prevalence of the three variants in the 5 different sample composition pools. Due to the variability of the viral loads used to determine the volumes that needed to be pooled to achieve these ratios, the true prevalences in the samples are unknown.

	Target		
	CAP063	CAP210	CAP239
MAJ_33	33.3%	33.3%	33.3%
MAJ_85	5%	85%	10%
MAJ_77	76.7%	20%	3.3%
MAJ_96	1%	3.3%	95.7%
MAJ_99	98.7%	0.33%	1%

Table 21: The eight different protocols used to prepare the samples for sequencing.

	Round 1			Round 2	
	Elongation Time	PCR Type	Number of cycles	PCR Type	Number of cycles
nrPCR (normal PCR)	2.5min	normal	25	normal	25
rcPCR (reduced cycle PCR in second round)	2.5min	normal	25	normal	15
2dPCR (both rounds amplified with dPCR)	2.5min	droplet	25	droplet	25
mxPCR (first round dPCR, second round normal PCR)	2.5min	droplet	25	normal	25
nrPCR (normal PCR)	10min	normal	25	normal	25
rcPCR (reduced cycle PCR in second round)	10min	normal	25	normal	15
2dPCR (both rounds amplified with dPCR)	10min	droplet	25	droplet	25
mxPCR (first round dPCR, second round normal PCR)	10min	droplet	25	normal	25

Table 22: Mutations between variants.

	CAP063	CAP239
CAP239	58	-
CAP210	64	48

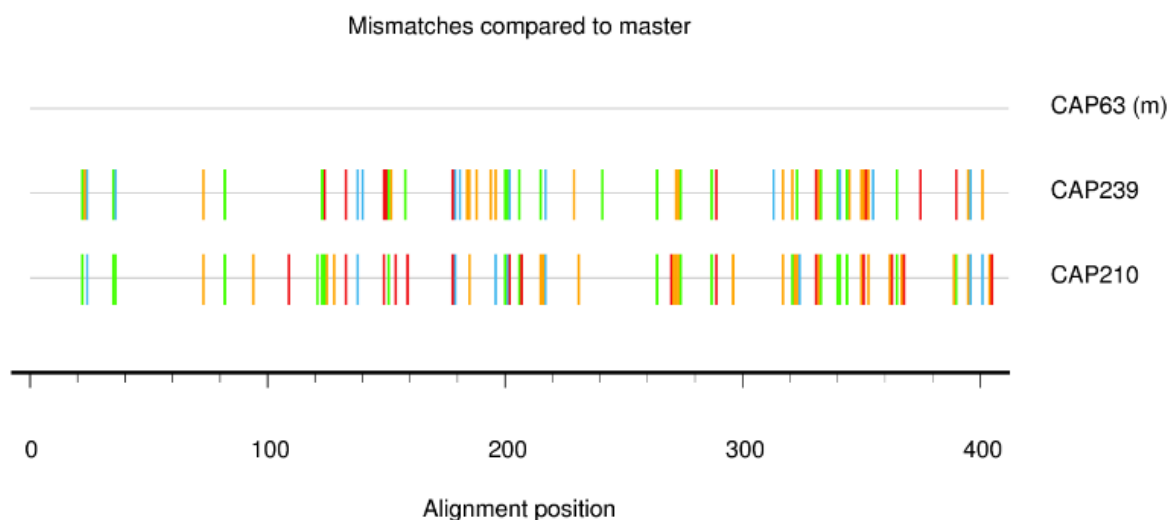


Figure 44: Highlighter plot showing differences between variants using the CAP63 variant as the reference.

4.2.1 Protocols that contained dPCR amplified less input templates.

PCR is used to increase the number of molecules that are available for sequencing. When making changes to PCR protocols, this amplification can be impaired in many ways. To check that the overall amplification efficiency was not impaired, we looked at the concentrations of PCR product after the second round of amplification. If the protocol changes prevent a fraction of molecules from getting amplified, then the number of unique PIDs that show up in the dataset will be reduced. Figure 45 shows concentrations of PCR product as well as the number of raw and consensus sequences obtained for each of the different protocols and elongation times.

Highest PCR product concentrations were achieved with the nrPCR protocol (Figure 45). The rcPCR protocol yielded the second highest concentrations, just slightly less than what was reached with the nrPCR protocol. Since the last 10 cycles added a negligible amount of extra PCR product, we deduce that the reaction was already nearing saturation by cycle 15 of the second round. Both protocols involving dPCR (2dPCR and mxPCR) yielded lower concentrations than the protocols that did not involve any dPCR. This is expected due to the inefficient DNA extraction step introduced by dPCR. The mxPCR protocol resulted in higher concentrations than the 2dPCR protocol because the second round of normal PCR was able to offset some of the diminished amplification from the first cycle. However, it did not reach the same concentrations as the protocols based purely on normal PCR.

The number of consensus sequences obtained from the protocols whose first round amplification was performed by normal PCR is 1.9 times higher than those involving dPCR implying that the inefficient extraction step leads to the loss of unique templates. The number of raw sequences obtained largely depends on the relative concentrations at which different samples were multiplexed during sequencing and has no relationship to the number of consensus sequences obtained.

The similar concentrations achieved by the nrPCR and rcPCR protocols suggest that the reaction reached saturation during the final cycles of the nrPCR protocol since those last cycles added a negligible amount of product. Later sections explore the effect of running the reaction in such sub-optimal conditions on PCR substitution error rates and PCR recombination rates. The lower concentrations achieved by the mxPCR and 2dPCR protocols are expected because of the inefficient extraction that dPCR introduces. A substantial reduction in the number of unique templates recovered was observed when using any protocol that involved dPCR. Together, the reduced concentration and lower numbers of unique templates recovered, suggest that and protocol involving dPCR will be a poor choice when the viral load of the sample is low or when the goal is to recover as many unique sequences as possible. Future work on using dPCR to reduce PCR induced recombination might benefit from exploring less stable emulsions to make the extraction step more efficient.

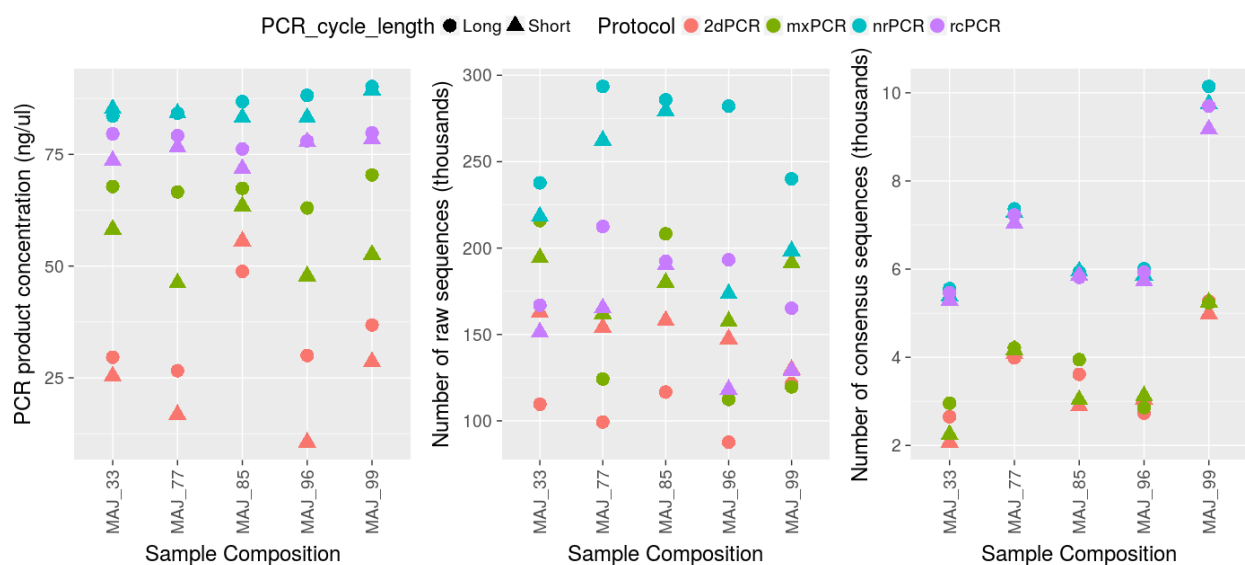


Figure 45: The concentrations of PCR product after amplification (left) and the number of consensus sequences obtained from each sample (right).

4.2.2 Biased amplification occurred in all protocols.

The stochastic nature of PCR amplification implies that certain templates may spuriously be amplified much better or worse than others. This bias causes wasteful redundancy in very large bins. On the other hand, small bins are at higher risk of producing incorrect consensus sequences. The size distributions of valid bins are shown in Figure 46. We observed that all four of the protocols sometimes produce long tails of oversized bins.

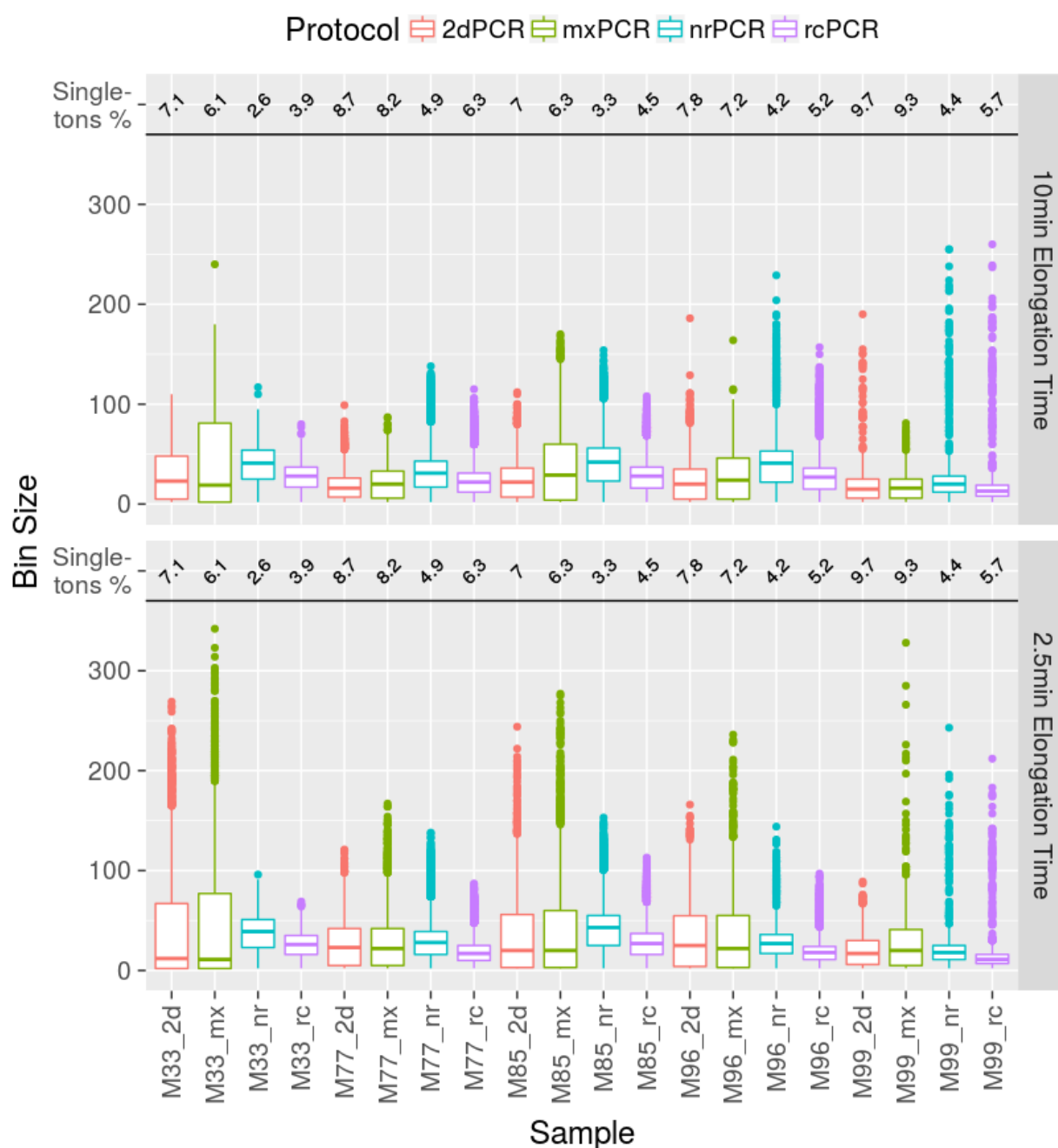


Figure 46: Distributions of bin sizes for each sample after removing bins spuriously introduced by sequencing errors in the PID.

The 25th percentile of bin size distributions are consistently larger for protocols that only utilize normal PCR (nrPCR and rcPCR) (16 vs. 4.6) while the interquartile range also tends to be more compact for the non-dPCR protocols (20.9 vs. 42.4). The samples produced by protocols that utilized dPCR had higher rates of singletons, drawing the 25th percentile down. Sequencing error is thought to be the primary mechanism that produce singletons and the sequencing error rates were higher in the protocols that involve dPCR (*Table 24*). A larger portion of reads from samples that were prepared with protocols that involved any dPCR (2dPCR and mxPCR) were assigned to offspring bins that have to be removed (5.8% vs. 10.6%). None of the protocols reduced the occurrence of over-sized bins. The prevalence of reads assigned to offspring bins was higher in protocols involving a dPCR step.

4.2.3 The Primer ID approach reduces the effect of sequencing errors.

The three clonal viruses were identical at 329 of the 412 nucleotides sequenced. Using these positions, we measured the aggregated sequencing and PCR substitution error rates. *Table 23* shows these error rates together with the quality scores provided by the MiSeq platform (for raw sequences) or the MotifBinner software package (for consensus sequences) for each sample. *Table 24* provides a summary for each protocol of the data presented in *Table 23*. Neither the elongation time nor the sample composition had an effect on the error rates in the raw sequences while significant variation was observed across different protocols (*Figure 47* and *Table 20*). The error rates for what this thesis refers to as raw sequences (sequences that have been minimally processed and merged), shown in *Table 23*, vary from 0.001251 (nrPCR; Long; MAJ_85) to 0.006104 (mxPCR; Long; MAJ_99) which is well below the error rates reported for unprocessed raw reads in literature. This illustrates the benefit of even the extremely minimal cleanup that was performed to produce these raw reads.

The samples were sequenced in two different sequencing runs, with the first run sequencing those prepared with the nrPCR and 2dPCR protocols and the second run sequencing the samples prepared with the rcPCR and mxPCR protocols. There was a difference between the quality scores for the raw sequences between the two runs (37.38 vs 36.69), translating to observable differences in the error rates of the raw sequences shown in *Figure 47* as the height of the dots belonging to the mxPCR and rcPCR protocols (The poor results of the 2dPCR protocol is discussed later). These differences were especially pronounced in the lowest quality regions (*Figure 48*), as seen by the large number of very dark blue dots above 0.01 in the y-axis.

Within the sequencing runs, the quality scores are comparable, but the protocols that involved at least one round of dPCR (2dPCR and mxPCR) had higher error rates. The error rate in high quality sequences

(scores over 30), was an order of magnitude higher in protocols involving at least 1 round of dPCR (2dPCR and mxPCR) shown in Table 25 (0.33 and 0.25 vs 0.03 and 0.05).

We speculate that this is mainly driven by higher rates of substitution errors during the dPCR steps as compared to normal PCR. Since the errors occur during the PCR process, the sequencer calls the bases with confidence (high quality scores). This would also explain the marginally higher error rates seen in the consensus sequences of the protocols that involve at least one dPCR step since a single PCR error is more likely to affect multiple sequence (than sequencing errors) and are thus more likely to affect the consensus sequences. We hypothesize that the increased rate of sequences assigned to offspring bins in the protocols that has a dPCR step is driven by this higher substitution error rate during the dPCR steps.

Notwithstanding the high quality scores achieved for the 2dPCR protocol, it still had high error rates. The second round of PCR during this protocol was also dPCR and it was seeded with billions of unique templates implying that each droplet was occupied by a large number of non-unique molecules. This co-occupancy of different molecules potentially worsens PCR recombination (discussed later) and might rapidly deplete the reagents available inside the droplets leading the reaction to run in poor conditions resulting in substitution errors during PCR. Again, since these errors occur during PCR, the sequencer has no access to information about them and calls the bases with high quality scores.

The error rates were elevated in the last position of the consensus sequences and it was driven by G to A substitution errors as shown by the elevated dots in the consensus sequence column of Figure 48. We were unable to find an explanation for this effect - it is prevalent across all datasets and does not coincide with increased error rates in the raw sequences. We note that a similar effect was observed in (S. Zhou et al., 2015).

The relationship between the error rates and the quality scores are shown in Figure 49. The relationship was consistent across all protocols, but varied when comparing raw and consensus sequences (as explained in the next paragraph). The error rates in the raw sequences behaved differently depending on whether the quality score was more or less than 20. From quality score 20 to quality score 38, the error rates decreased approximately linearly from 3% to 0.1%. When the quality score dropped below 20, accuracy deteriorated rapidly yielding error rates in excess of 30% at scores below 10. Datasets should be cleaned aggressively to avoid having sequences with bases with quality scores below 20.

Due to the consensus generation parameters, if the quality score `MotifBinner` were to assign to a base falls below 35, the base will be represented as an N and excluded from this analysis, unless

otherwise stated. Nonetheless, in the limited range between 35 and 38, a strong correlation can be observed (as shown in Figure 49) between the error rate and the MotifBinner assigned error rate (79.5%). It should be noted that an overwhelming majority of bases (68,330,339 out of 68,330,505) were assigned a quality score of exactly 38 by MotifBinner implying significant room for improvement in the algorithms that assign quality scores in MotifBinner. Overall, generating consensus sequences reduced the error rates by approximately an order of magnitude, calculated by comparing the “Error Rate” column in the “Raw Sequences” part of Table 24 to the “Error Rate” column in the “Consensus Sequences” part of the table. The error rate of 0.01% observed in the consensus sequences is comparable to previously published PID error rates and is likely primarily due to errors during reverse transcription (S. Zhou et al., 2015).

Generating consensus sequences from larger bins means that more information is incorporated and higher accuracy is expected. To investigate what bin sizes are required to achieve high quality consensus sequences, bin size was graphed against the chance that the consensus sequence is error-free in Figure 50. For protocols that do not involve a dPCR step, the highest accuracy consensus sequences are already achieved by bin size 10, with 1 in 20 consensus sequences matching one of the input variants exactly. For protocols involving a dPCR step, the highest quality bins are seen at bin sizes in excess of 20. Additionally, the accuracy of these consensus sequences is lower than those that were produced by non-dPCR protocols (nrPCR and rcPCR). The increasing spread seen for larger bin sizes, is a result of the lower number of bins that has larger sizes, hence the estimation of the accuracy of such consensus sequences is more variable. This spread only reflects our inability to confidently assess the accuracy of these consensus sequences and not the actual accuracy of the sequences themselves.

Table 23: Error rates and average quality score for all positions where all three clones were identical for each dataset.

Protocol	Elon- gation Time	Compo- sition	Raw Sequences			Consensus Sequences		
			Total Number of Bases	Error of Rate	Average Quality Score	Total Number of Bases	Error Rate	Average Quality Score
2dPCR	Short	MAJ_33	48,243,902	0.403%	37.44	679,714	0.011%	38
2dPCR	Long	MAJ_33	31,892,931	0.4831%	37.42	868,560	0.0181%	38
2dPCR	Short	MAJ_77	44,920,015	0.4345%	37.36	1,337,385	0.0117%	38
2dPCR	Long	MAJ_77	28,388,752	0.5058%	37.37	1,304,485	0.0181%	38
2dPCR	Short	MAJ_85	46,697,931	0.4264%	37.39	945,546	0.0098%	38
2dPCR	Long	MAJ_85	34,171,914	0.4827%	37.42	1,184,729	0.0154%	38
2dPCR	Short	MAJ_96	43,486,562	0.419%	37.44	995,225	0.0113%	38

2dPCR	Long	MAJ_96	25,201,071	0.4959%	37.4	892,577	0.0173%	38
2dPCR	Short	MAJ_99	37,307,942	0.4242%	37.4	1,632,498	0.0071%	38
2dPCR	Long	MAJ_99	34,144,936	0.5294%	37.34	1,727,250	0.0156%	38
mxPCR	Short	MAJ_33	58,401,119	0.4546%	36.83	734,657	0.0185%	38
mxPCR	Long	MAJ_33	63,788,823	0.5309%	36.77	967,260	0.0206%	38
mxPCR	Short	MAJ_77	47,429,298	0.5296%	36.62	1,363,376	0.0181%	38
mxPCR	Long	MAJ_77	35,816,914	0.5576%	36.65	1,377,523	0.0244%	38
mxPCR	Short	MAJ_85	53,588,507	0.4881%	36.74	990,619	0.0172%	38
mxPCR	Long	MAJ_85	61,888,519	0.5344%	36.77	1,292,641	0.0194%	38
mxPCR	Short	MAJ_96	46,676,875	0.4889%	36.81	1,017,268	0.0168%	38
mxPCR	Long	MAJ_96	32,775,638	0.5332%	36.79	931,399	0.0242%	38
mxPCR	Short	MAJ_99	55,372,674	0.5159%	36.69	1,715,077	0.0086%	38
mxPCR	Long	MAJ_99	33,527,732	0.6104%	36.58	1,711,458	0.0202%	38
nrPCR	Short	MAJ_33	69,282,465	0.1331%	37.33	1,766,730	0.01%	38
nrPCR	Long	MAJ_33	75,214,993	0.1373%	37.3	1,818,383	0.009%	38
nrPCR	Short	MAJ_77	80,969,203	0.1339%	37.36	2,382,947	0.0076%	38
nrPCR	Long	MAJ_77	89,771,269	0.1275%	37.43	2,411,241	0.0095%	38
nrPCR	Short	MAJ_85	88,517,779	0.1307%	37.35	1,949,983	0.0088%	38
nrPCR	Long	MAJ_85	89,154,065	0.1251%	37.41	1,944,390	0.0091%	38
nrPCR	Short	MAJ_96	55,089,734	0.1389%	37.27	1,917,083	0.0091%	38
nrPCR	Long	MAJ_96	85,827,546	0.1278%	37.45	1,958,537	0.0111%	38
nrPCR	Short	MAJ_99	62,277,068	0.1364%	37.31	3,199,854	0.006%	38
nrPCR	Long	MAJ_99	72,779,735	0.1263%	37.43	3,319,281	0.0079%	38
rcPCR	Short	MAJ_33	47,340,468	0.3174%	36.59	1,735,146	0.01%	38
rcPCR	Long	MAJ_33	51,961,602	0.3373%	36.5	1,788,115	0.0088%	38
rcPCR	Short	MAJ_77	49,966,875	0.3186%	36.63	2,299,052	0.0092%	38
rcPCR	Long	MAJ_77	64,070,776	0.2935%	36.76	2,366,168	0.0096%	38
rcPCR	Short	MAJ_85	59,350,284	0.3071%	36.63	1,919,057	0.009%	38
rcPCR	Long	MAJ_85	59,197,299	0.294%	36.73	1,904,581	0.0083%	38
rcPCR	Short	MAJ_96	37,107,910	0.3429%	36.48	1,870,365	0.0103%	38
rcPCR	Long	MAJ_96	58,105,019	0.2914%	36.82	1,934,849	0.0108%	38
rcPCR	Short	MAJ_99	39,571,133	0.3276%	36.52	3,006,402	0.0072%	38
rcPCR	Long	MAJ_99	49,329,602	0.3047%	36.71	3,173,534	0.0073%	38

Table 24: Error rates and quality scores for raw and consensus sequences.

Protocol	Raw Sequences		Consensus Sequences	
	Error Rate	Average Quality Score	Error Rate	Average Quality Score
nrPCR	0.13%	37.37	0.01%	38
rcPCR	0.31%	36.65	0.01%	38
mxPCR	0.52%	36.73	0.02%	38
2dPCR	0.46%	37.39	0.01%	38

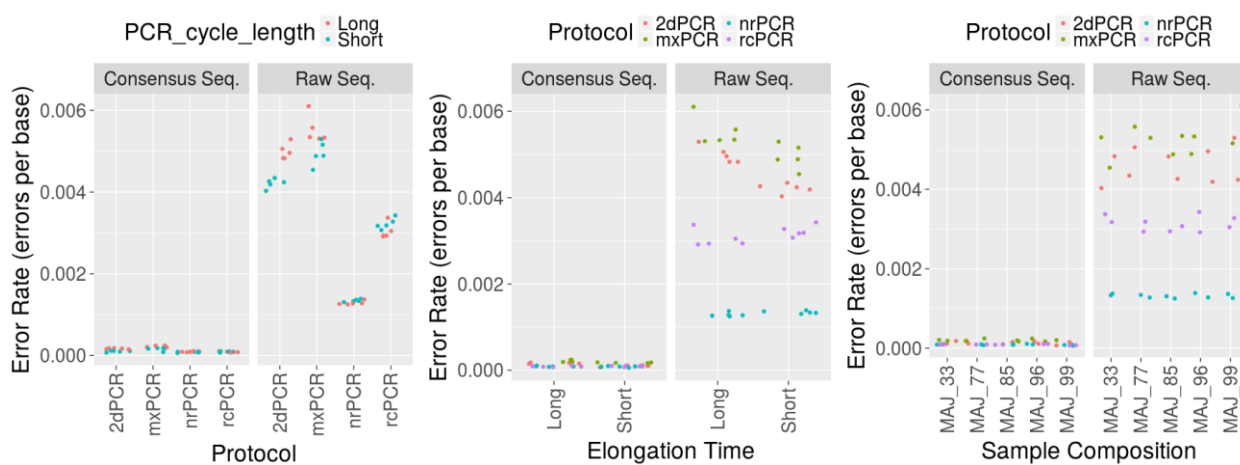


Figure 47: Error rate by protocol (Left), elongation time (middle) and sample composition (right)

UNIVERSITY OF
WESTERN CAPE

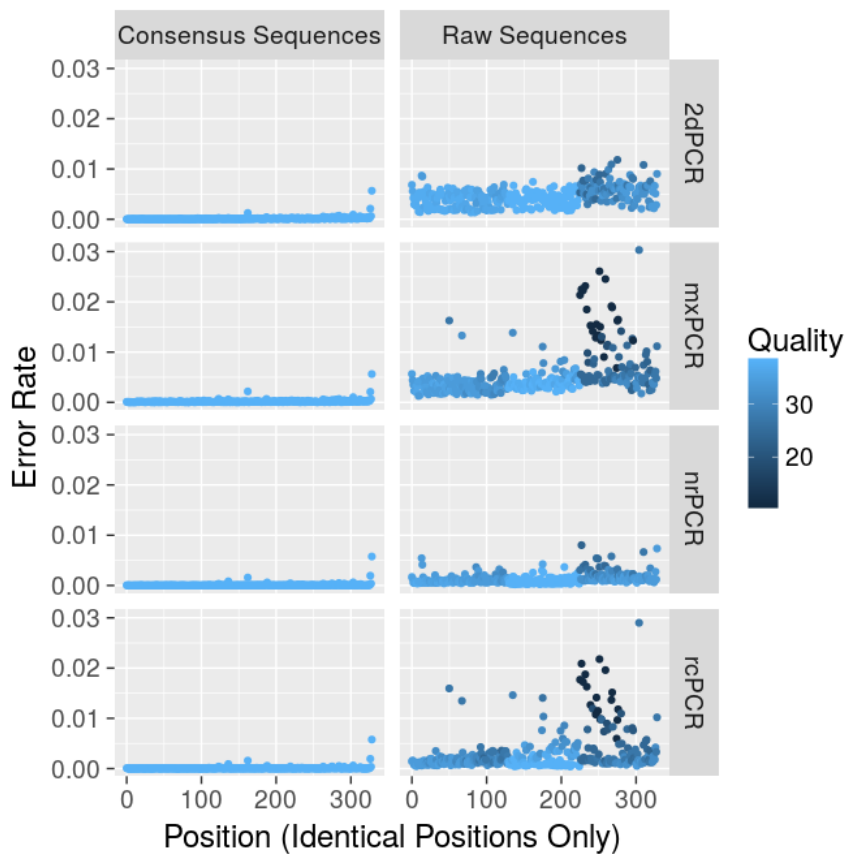


Figure 48: Error rates by position for raw and consensus sequences for each protocol. Only the 329 positions where the three variants matched are shown on the x-axis. Color indicates the 5th percentile of the quality scores for the relevant position, implying that 1 in 20 sequences had a quality score lower than the color indicates.

Table 25: Error rates in the raw sequences by categorized quality score

Protocol	Quality Score			
	≤ 10	(10,20]	(20,30]	> 30
2dPCR	16.91	11.89	1.59	0.33
mxPCR	26.11	12.56	1.61	0.25
nrPCR	25.10	11.46	0.97	0.03
rcPCR	32.17	12.22	1.23	0.05

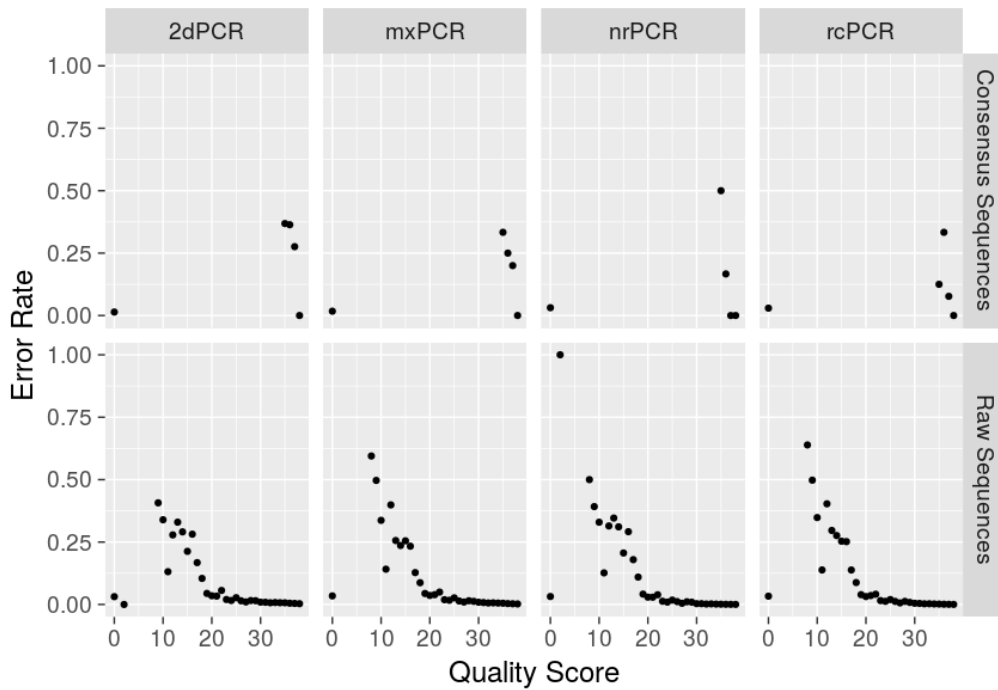


Figure 49: Error rate by quality score for each protocol.

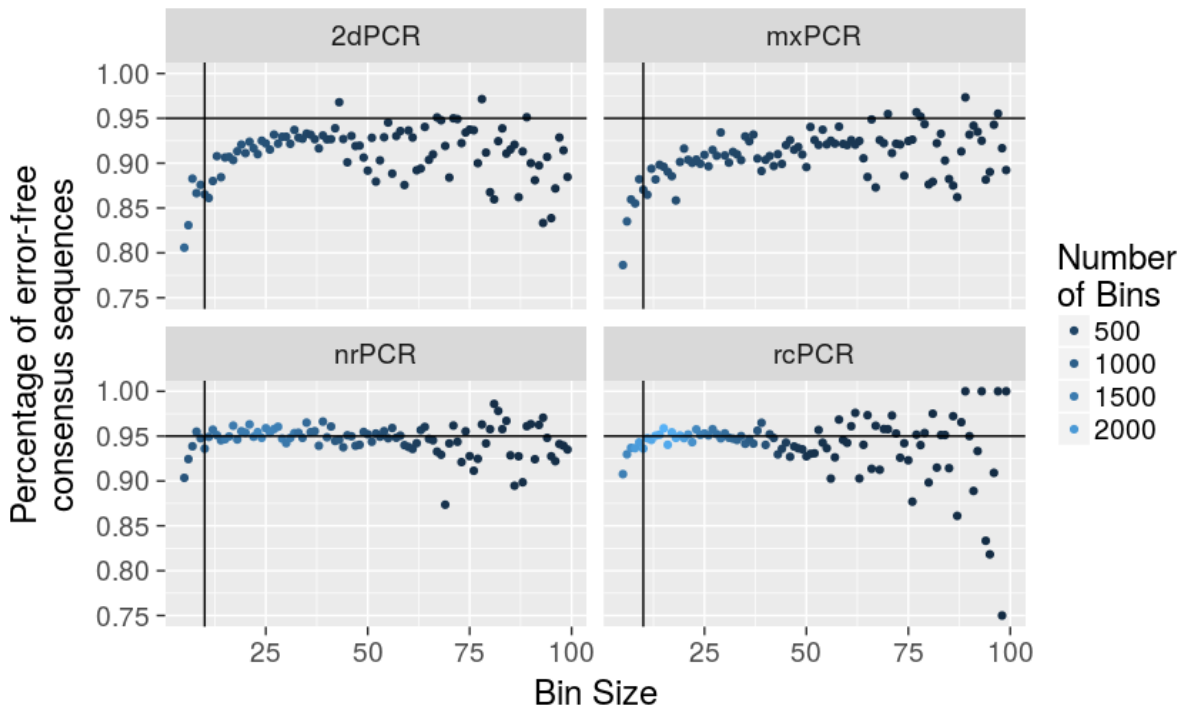


Figure 50: The relationship between bin size and the chance of obtaining an error-free consensus sequence.

4.2.4 Prevalence estimates based on PID data are less variable.

By comparing the raw and consensus sequences to the three different input templates, each sequence was identified as either one of the three input templates or a chimeric sequence. The chimeric

sequences were omitted from all analyses that involved prevalences. Using the sequences classified as originating from one of the templates, the prevalence of each variant was computed for each dataset. Due to the variability of viral loads, actual prevalence of each variant could not be determined accurately making it impossible to measure estimation bias. Hence our investigation focused on exploring relationships between estimates based on the library preparation protocols, comparing the variability of estimates based on raw sequences to the estimates based on consensus sequences and ensuring that all approaches can detect the lowest prevalence variants.

To investigate the effect of elongation time, number of cycles and the use of dPCR, the individual prevalence estimates were plotted in Figure 51, Figure 52, Figure 53, and Figure 54. The first figure, Figure 51, compares the differences between prevalence estimates by elongation time for estimates based on raw sequences. Each of the five samples contain three variants implying that 15 different prevalences must be estimated. Eight protocols were used so that there are eight replicates for each prevalence estimate. These eight estimates are split into two groups, one with long elongation times (the red points in Figure 51) and one with short elongation times (the blue points in Figure 51). There was no systematic pattern by which the estimates differ from each other based on the elongation time. This was also true for the consensus sequences as shown in Figure 52.

Similar to the investigation for the elongation time, the effects of number of cycles and dPCR on prevalence estimates are presented in Figure 53 (for estimates based on raw sequences) and Figure 54 (for estimates based on consensus sequences). In these two figures, the eight replicates are divided into four groups (2dPCR, mxPCR, nrPCR and rcPCR), so that there are only two estimates for each prevalence. Again, there is no pattern in which some of the points are systematically higher or lower than the others for either the estimates based on raw or consensus sequences. Based on these four figures we conclude that neither elongation time, number of cycles nor the use of dPCR has a substantial effect on the prevalence estimates. All investigations of the prevalences are thus based on pooled estimates. We recognize that due to the low number of replicates the ability to detect such effects are limited.

Box and whisker plots of the pooled prevalence estimates are show in Figure 55 and the individual estimates are listed in Table 26. These plots summarize the distribution of the estimates with 5 summary statistics and individually presents observations deemed to be outliers. The median is shown as the central thick horizontal bar and the 25th and 75th percentiles form the upper and lower ends of the box. The upper and lower edges of the box are called the upper and lower hinges. From the upper and lower hinges, lines extend upwards and downward up to the largest (smallest) value that greater (smaller) than the 75th (25th) percentile but within 1.5 times the length of the interquartile range (IQR,

the distance from the 25th to the 75th percentiles) of the upper (lower) hinges. Any values that are more than 1.5 times the length of the IQR away from the closest hinge are considered outliers and are individually represented with points. Each box and whisker displays the distribution of the eight replicates of the estimates that are available for the prevalence of each of the three variants in each of the five samples. The top part of Figure 55 displays estimates based on raw sequences while the bottom part of Figure 55 shows results based on consensus sequences.

Due to our uncertainty about the true ratios of the samples, we were not able to measure the bias of the estimates forcing us to restrict the analysis to the variability of the estimates. Estimates based on PID data were less variable than estimates based on the raw sequences (Figure 56). On average, the standard deviation of the estimates based on raw sequences were 3.3 times higher when compared to estimates based on the consensus sequences. The implications of this larger variability is illustrated in Table 26 and Table 27. The largest absolute difference between prevalence estimates was 20.68 and was observed on the MAJ_77 sample comparing the 2dPCR protocol with short elongation times to the rcPCR sample with short elongation times when estimating the prevalence of CAP63. This means that in one of the cases where the same thing was estimated twice (based on raw sequences) the two estimates differed by 20.68. The 2dPCR protocol with long elongation times estimated (using raw sequences) the prevalence at 82.78 while the rcPCR protocol with short elongation times estimated (with raw sequences) this same prevalence at 62.15. In contrast, the largest absolute difference observed when looking only at estimates based on consensus sequences was 4.91. The 2dPCR protocol with long elongation times estimated (with consensus sequences) the prevalence at 79.05 while the 2dPCR protocol with short elongation times estimated this same prevalence at 83.96.

Table 27 shows how much the individual estimates differed from the average of all estimates. In other words, each cell shows how much the prevalence estimate represented by that cell differed from the average of all the prevalence estimates represented by that same row. For example, the first value in Table 27 is -0.85, meaning that the estimate obtained using consensus sequences from the 2dPCR protocol with long elongation times for CAP63 in the MAJ_33 sample was 0.85 lower than the average of all the estimates of all the protocols for CAP63 on the MAJ_33 sample using consensus sequences. Analyzing Table 27, one can observe that there are 22 estimates based on raw sequences that differ by more than 3 from the average while there is only one estimate based on consensus sequences that differs by more than 3 from the average.

Lastly, each variant was represented by at least one consensus sequence across all samples and protocols. When the 2dPCR protocol is excluded, each variant was represented by at least 18 sequences, demonstrating the ability of the PID approach to detect low prevalence variants. Based on

these observations, estimates based on consensus sequences are superior because they are less variable and are still able to detect minority variants.

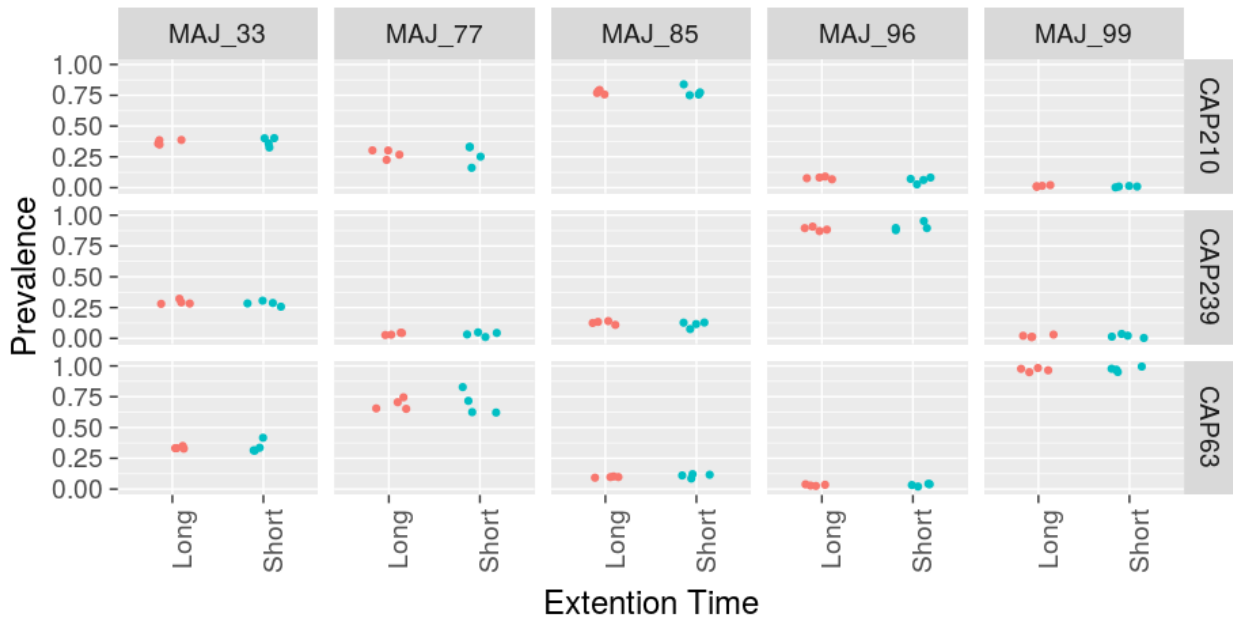


Figure 51: Individual prevalence estimates for each variant in each sample based on raw sequences by elongation time. There are only four estimates for each quantity of interest, one for each protocol: nrPCR, rcPCR, mxPCR and 2dPCR.

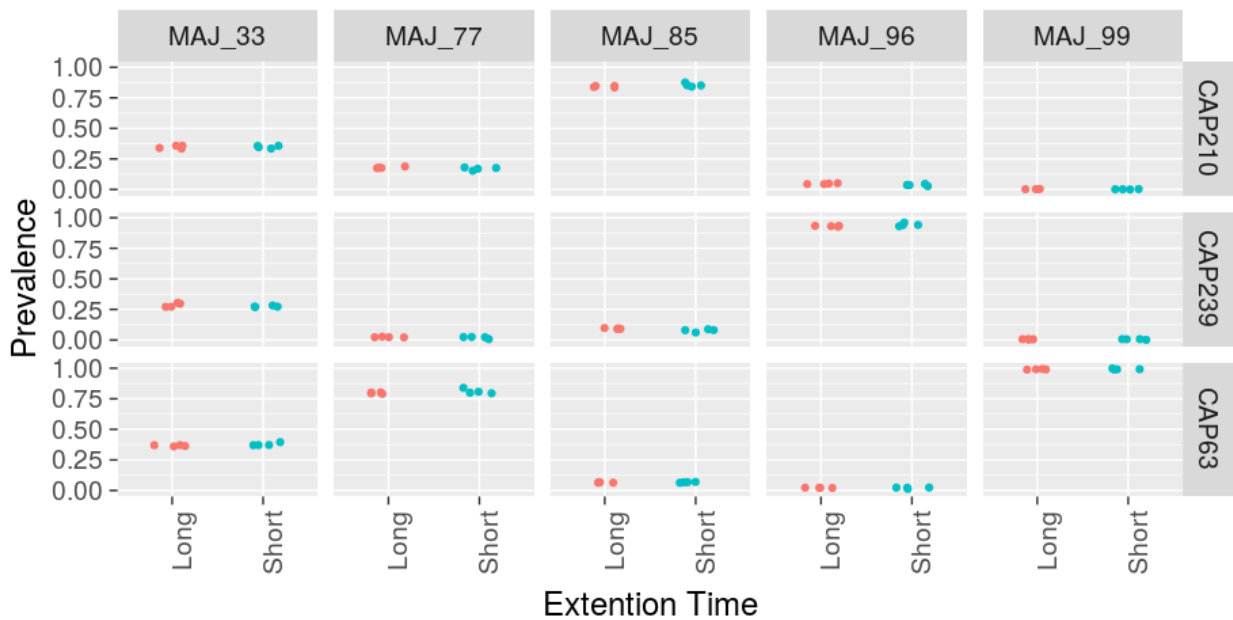


Figure 52: Individual prevalence estimates for each variant in each sample based on consensus sequences by elongation time. There are only four estimates for each quantity of interest, one for each protocol: nrPCR, rcPCR, mxPCR and 2dPCR.

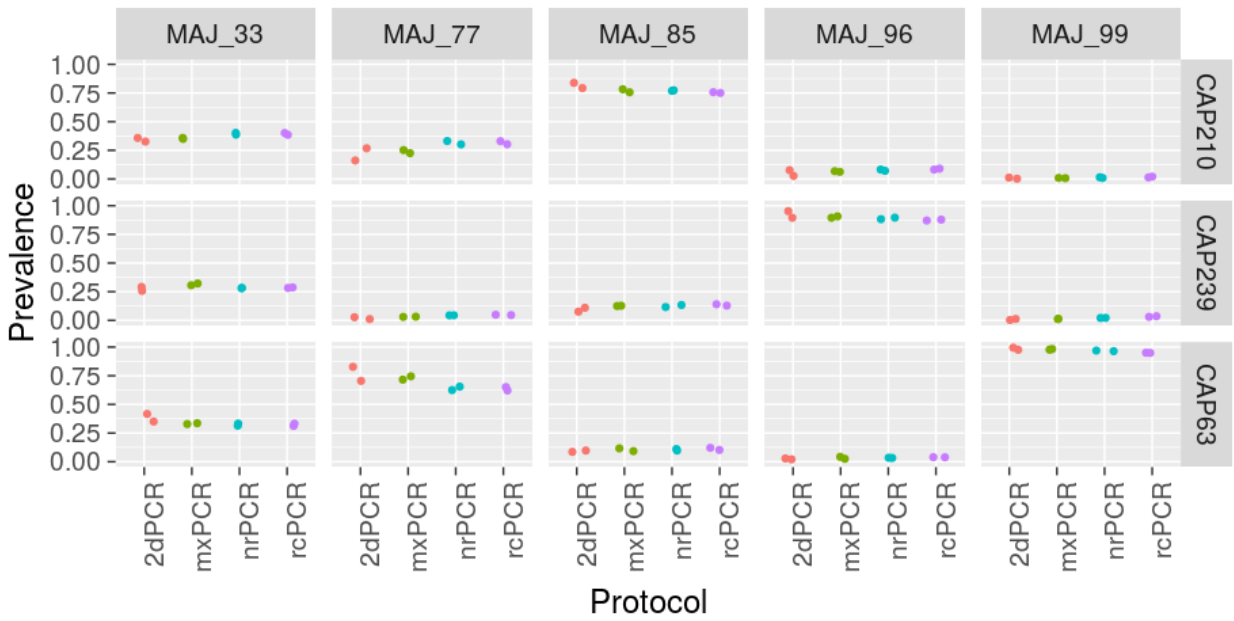


Figure 53: Individual prevalence estimates for each variant in each sample based on raw sequences by protocol. There are only two estimates for each quantity of interest, one for each elongation time.

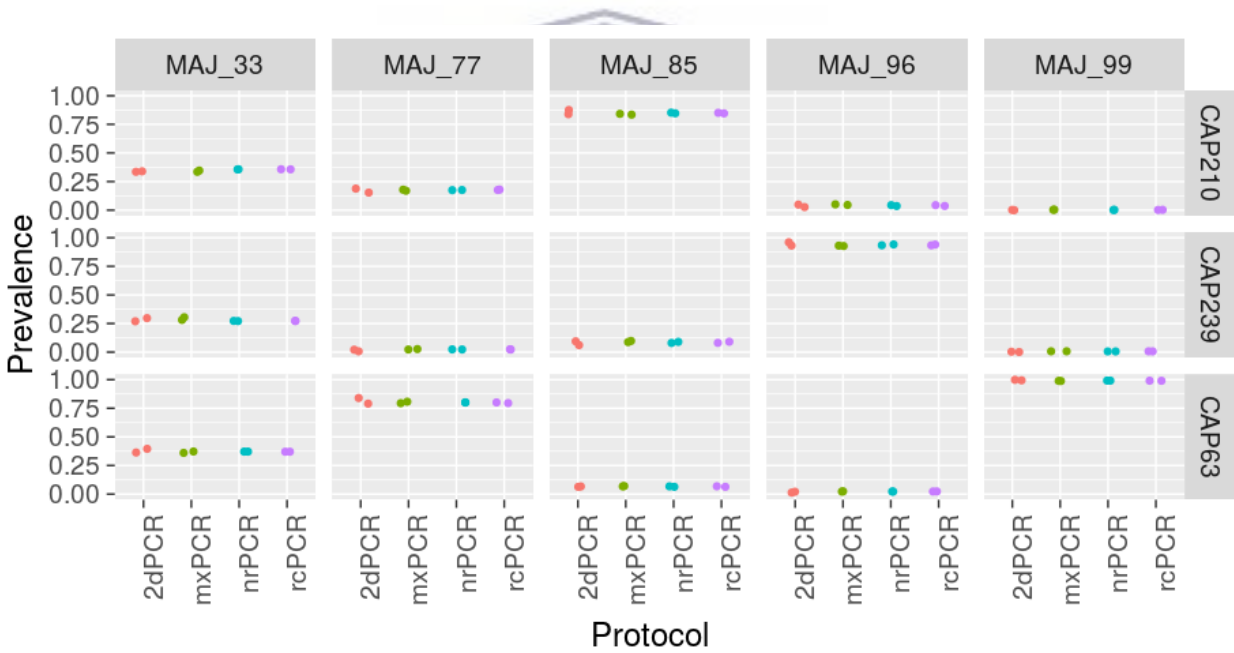


Figure 54: Individual prevalence estimates for each variant in each sample based on consensus sequences by protocol. There are only two estimates for each quantity of interest, one for each elongation time.

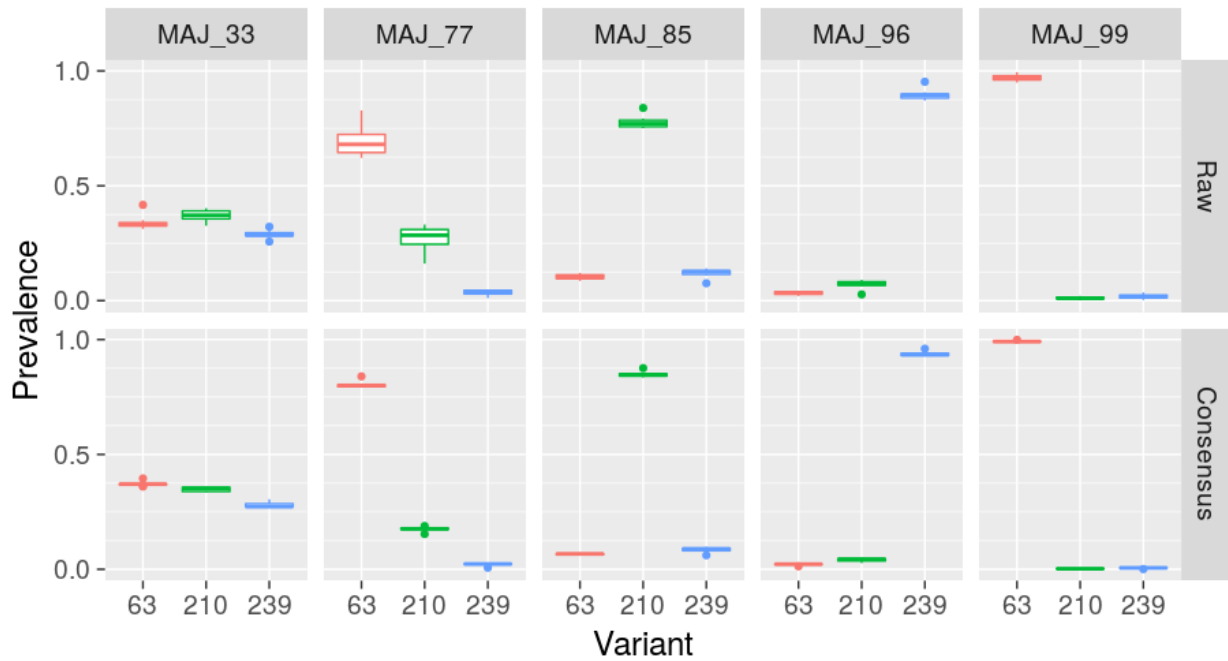


Figure 55: Box and whisker plots of the prevalence estimates based on the raw and consensus sequences. These plots summarize the distribution of the estimates with 5 summary statistics and individually presents observations deemed to be outliers. The median is shown as the central thick horizontal bar and the 25th and 75th percentiles form the upper and lower ends of the box. The upper and lower edges of the box are called the upper and lower hinges. From the upper and lower hinges, lines extend upwards and downward up to the largest (smallest) value that greater (smaller) than the 75th (25th) percentile but within 1.5 times the length of the interquartile range (IQR, the distance from the 25th to the 75th percentiles) of the upper (lower) hinges. Any values that are more than 1.5 times the length of the IQR away from the closest hinge are considered outliers and are individually represented with points.

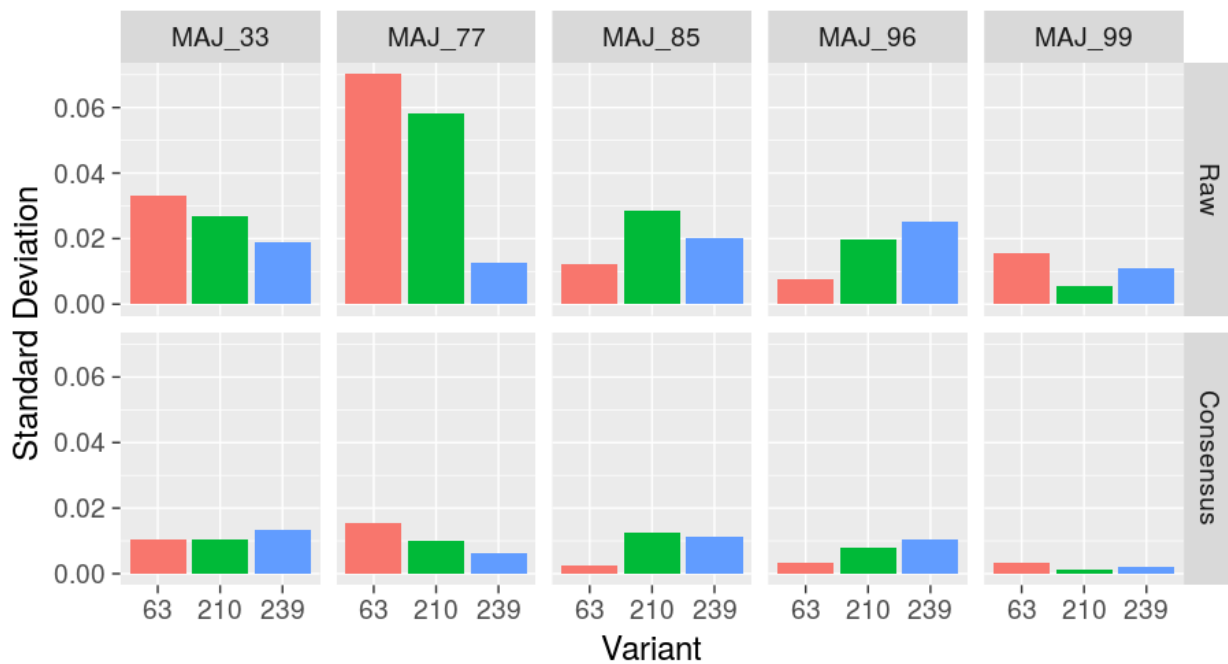


Figure 56: Variability of prevalence estimates.

Table 26: Prevalence estimates, in percentages, for each variant for both sequence types on each sample by each protocol and elongation time.

Variant	Seq. Type	Sample	2dPCR		mxPCR		nrPCR		rcPCR	
			Long	Short	Long	Short	Long	Short	Long	Short
63	Cons.	MAJ_33	36.34	39.53	36.02	37.21	37.13	37.11	37.02	37.01
63	Raw	MAJ_33	35.01	41.67	32.85	33.59	33.26	31.43	33.19	31.22
210	Cons.	MAJ_33	33.95	33.53	33.5	34.62	35.75	35.62	35.77	35.68
210	Raw	MAJ_33	35.81	32.63	35.01	35.82	38.72	40.21	38.53	40.08
239	Cons.	MAJ_33	29.71	26.94	30.48	28.17	27.12	27.26	27.21	27.3
239	Raw	MAJ_33	29.18	25.7	32.15	30.59	28.01	28.36	28.28	28.7
63	Cons.	MAJ_77	79.05	83.96	79.44	80.74	80.19	79.98	80.09	79.5
63	Raw	MAJ_77	70.52	82.78	74.5	71.62	65.5	62.48	65.13	62.15
210	Cons.	MAJ_77	18.83	15.35	17.89	16.97	17.49	17.62	17.64	18
210	Raw	MAJ_77	26.77	16.12	22.52	25.19	30.18	33.12	30.27	32.97
239	Cons.	MAJ_77	2.12	0.69	2.67	2.29	2.32	2.4	2.27	2.49
239	Raw	MAJ_77	2.71	1.09	2.98	3.19	4.33	4.4	4.6	4.87
63	Cons.	MAJ_85	6.68	6.33	6.75	7.04	6.36	6.75	6.32	6.81
63	Raw	MAJ_85	9.74	8.58	9.23	11.64	9.77	11.06	10.25	12.09
210	Cons.	MAJ_85	83.88	87.56	83.47	84.19	84.65	85.23	84.59	85.08
210	Raw	MAJ_85	79.29	83.86	78.27	75.63	76.85	77.33	75.71	75.04
239	Cons.	MAJ_85	9.44	6.12	9.78	8.77	8.99	8.02	9.09	8.11
239	Raw	MAJ_85	10.97	7.56	12.5	12.73	13.38	11.61	14.04	12.87
63	Cons.	MAJ_96	2.03	1.3	2.19	2.39	2.2	2.25	2.26	2.32
63	Raw	MAJ_96	2.84	2.06	2.53	4.26	3.49	3.29	3.86	3.95
210	Cons.	MAJ_96	4.84	2.7	5.09	4.56	4.42	3.62	4.39	3.68
210	Raw	MAJ_96	7.67	2.69	6.77	6.22	8.24	7.1	9.04	8.21
239	Cons.	MAJ_96	93.13	96.01	92.72	93.05	93.38	94.13	93.35	94
239	Raw	MAJ_96	89.49	95.25	90.7	89.52	88.28	89.62	87.09	87.85
63	Cons.	MAJ_99	99.43	99.88	98.83	98.94	99.16	99.21	99.04	99.1
63	Raw	MAJ_99	97.65	99.47	98.3	97.73	96.44	96.95	94.91	95.06
210	Cons.	MAJ_99	0.23	0.02	0.44	0.35	0.25	0.2	0.31	0.2
210	Raw	MAJ_99	1.15	0.21	0.68	0.89	1.46	0.89	2.05	1.36
239	Cons.	MAJ_99	0.34	0.1	0.73	0.71	0.6	0.6	0.65	0.7
239	Raw	MAJ_99	1.21	0.32	1.02	1.38	2.09	2.16	3.04	3.58

Table 27: Deviation of the individual prevalence estimates from the average prevalence estimate for that variant and sequence type in that sample across all protocols and elongation times. Each cell shows the difference between that cell and the row average in Table 26.

Variant	Seq. Type	Sample	2dPCR		mxPCR		nrPCR		rcPCR	
			Long	Short	Long	Short	Long	Short	Long	Short
63	Cons.	MAJ_33	-0.85	2.35	-1.15	0.05	-0.05	-0.05	-0.15	-0.15
63	Raw	MAJ_33	0.975	7.675	-1.225	-0.425	-0.725	-2.625	-0.825	-2.825
210	Cons.	MAJ_33	-0.8125	-1.3125	-1.3125	-0.2125	0.9875	0.7875	0.9875	0.8875
210	Raw	MAJ_33	-1.2875	-4.4875	-2.0875	-1.2875	1.6125	3.1125	1.4125	3.0125
239	Cons.	MAJ_33	1.675	-1.125	2.475	0.175	-0.925	-0.725	-0.825	-0.725
239	Raw	MAJ_33	0.325	-3.175	3.225	1.725	-0.875	-0.475	-0.575	-0.175
63	Cons.	MAJ_77	-1.3625	3.6375	-0.9625	0.3375	-0.1625	-0.3625	-0.2625	-0.8625
63	Raw	MAJ_77	1.1625	13.4625	5.1625	2.2625	-3.8375	-6.8375	-4.2375	-7.1375
210	Cons.	MAJ_77	1.325	-2.075	0.425	-0.475	0.025	0.125	0.125	0.525
210	Raw	MAJ_77	-0.35	-11.05	-4.65	-1.95	3.05	5.95	3.15	5.85
239	Cons.	MAJ_77	-0.0625	-1.4625	0.5375	0.1375	0.1375	0.2375	0.1375	0.3375
239	Raw	MAJ_77	-0.825	-2.425	-0.525	-0.325	0.775	0.875	1.075	1.375
63	Cons.	MAJ_85	0.075	-0.325	0.075	0.375	-0.225	0.175	-0.325	0.175
63	Raw	MAJ_85	-0.5875	-1.6875	-1.0875	1.3125	-0.4875	0.8125	-0.0875	1.8125
210	Cons.	MAJ_85	-0.95	2.75	-1.35	-0.65	-0.15	0.35	-0.25	0.25
210	Raw	MAJ_85	1.55	6.15	0.55	-2.15	-0.85	-0.45	-2.05	-2.75
239	Cons.	MAJ_85	0.8625	-2.4375	1.2625	0.2625	0.4625	-0.5375	0.5625	-0.4375
239	Raw	MAJ_85	-0.9625	-4.3625	0.5375	0.7375	1.4375	-0.3625	2.0375	0.9375
63	Cons.	MAJ_96	-0.1125	-0.8125	0.0875	0.2875	0.0875	0.0875	0.1875	0.1875
63	Raw	MAJ_96	-0.4875	-1.1875	-0.7875	1.0125	0.2125	0.0125	0.6125	0.6125
210	Cons.	MAJ_96	0.6375	-1.4625	0.9375	0.4375	0.2375	-0.5625	0.2375	-0.4625
210	Raw	MAJ_96	0.7125	-4.2875	-0.1875	-0.7875	1.2125	0.1125	2.0125	1.2125
239	Cons.	MAJ_96	-0.6125	2.2875	-1.0125	-0.7125	-0.3125	0.3875	-0.3125	0.2875
239	Raw	MAJ_96	-0.225	5.575	0.975	-0.225	-1.425	-0.125	-2.625	-1.925
63	Cons.	MAJ_99	0.2125	0.7125	-0.3875	-0.2875	0.0125	0.0125	-0.1875	-0.0875
63	Raw	MAJ_99	0.55	2.45	1.25	0.65	-0.65	-0.15	-2.15	-1.95
210	Cons.	MAJ_99	-0.025	-0.225	0.175	0.075	-0.025	-0.025	0.075	-0.025
210	Raw	MAJ_99	0.0125	-0.8875	-0.3875	-0.1875	0.4125	-0.1875	0.9125	0.3125
239	Cons.	MAJ_99	-0.25	-0.45	0.15	0.15	0.05	0.05	0.15	0.15
239	Raw	MAJ_99	-0.65	-1.55	-0.85	-0.45	0.25	0.35	1.15	1.75
Average deviation			-0.0112	0.0088	-0.0046	-0.0046	0.0088	-0.0012	-0.0012	0.0054

4.2.5 PCR recombination is influenced by sequence homology, but effectively removed by the PID approach.

By comparing the raw and consensus sequences to the original input templates using edit distances, each sequence was flagged as either one of the variants or as resulting from a recombination event between two variants in a certain region. This detailed classification of each read (raw or consensus) enables a careful study of recombination. For detectable recombination to occur in a consensus sequence, either the event had to occur in an early cycle or a large number of recombination events had to occur in sequences with the same PID. Thus, as shown in Table 28 and Figure 57, the rate of detectable recombination is reduced substantially by generating consensus sequences. For all the protocols except the 2dPCR protocol, only 0.02% or 0.01% of the consensus sequences are chimeras, more than two orders of magnitude lower than in the raw sequences in which 6.5% to 11.65% of the sequences are chimeras.

dPCR partitions the reaction into multiple isolated droplets which should theoretically reduce the rate of recombination. The opposite effect was observed in both the consensus and raw sequences when dPCR was used for both 1st and 2nd rounds of amplification (2dPCR) (Table 28 and Figure 57). We hypothesize that the increased recombination is driven by the high occupancy of the droplets during the second round of amplification. Due to the low input number of viral templates, most droplets during the first round PCR should be seeded with only a single molecule, theoretically preventing recombination in the first round of PCR. However, partitioning the second round PCR might have promoted recombination events due to the high occupancy of the droplets (~8 billion templates distributed among ~16 000 droplets). This hypothesis is supported by the lower rates of recombination seen when using the mxPCR protocol (which utilized dPCR for round 1 and normal PCR for round 2). Both protocols using only normal PCR performed well.

Longer elongation times decreases the amount of partially amplified product in the reaction and should result in lower recombination rates. As seen in Table 28, longer elongation times consistently lead to lower recombination rates with the exception of a minor increase in the consensus sequences produced with the rcPCR protocol. However, this effect was suppressed in samples where a single variant accounted for more than 95% of the sample and in the rcPCR protocol (Figure 57). The theoretical processivity of Q5 polymerase is about 30 seconds per kb, hence the 2.5 min elongation time we chose as a minimum was already conservative limiting the potential for improvement by further increasing the elongation time.

PCR recombination is amplified in the later cycles (Kanagawa, 2003), when the template concentration is high. The rcPCR protocol reduced the number of cycles in the 2nd round of amplification, resulting

in fewer chimeric raw sequences as compared to the nrPCR protocol when using short elongation times (11.65% vs 9.88%) as shown in Table 28. The opposite effect was observed when looking at long elongation times (8.33% vs 9.29%), suggesting that the longer elongation times reduces PCR recombination by affecting similar mechanisms as reducing the number of cycles. Since there were so few chimeric consensus sequences, the effect on the consensus sequences was negligible.

MotifBinner introduces Ns into consensus sequence when certain quality conditions are not met (section 3.1.77). We performed one analysis on the consensus sequences that do contain Ns. To explore whether or not consensus sequences containing Ns should be removed and what drives the formation of Ns in consensus sequences, the percentages of chimeric and non-chimeric consensus sequences that contain Ns are shown in Figure 58. The percentage of non-chimeric sequences that contain Ns are very low. In chimeric consensus sequences, the sample composition strongly affects the chance of the consensus sequence containing Ns, starting from 95.8% in the samples where the three variants are prevalent at similar rates dropping to 32.3% in samples where the majority variant accounts for 99% of the sample. The relationship between the production of chimeric consensus sequences and sample composition is explored later in this section. This relationship is also observed when the data from 2dPCR protocol, which accounts for 254 of the 373 (68%) chimeric consensus sequences that contain Ns, is removed from this analysis. This correspondence between the occurrence of N's and the likelihood of consensus sequences to be chimeric justifies the exclusion of such sequences from any further analysis.

The primary mechanism by which recombination occurs is when partially amplified products, resulting from imperfect polymerase processivity, act as primers during subsequent PCR amplification cycles (J. Liu et al., 2014). The template to prime to is randomly selected from the most proximate template molecules in the reaction mix and mediated by homology between the template and partial product. Hence, minority variants are at higher risk of being transformed into chimeras due to the low availability of identical templates in the reaction. As shown in Figure 59, raw sequences of which the start or end belonged to a minority variant (>1% prevalence), had a 26 to 46% probability of being chimeric. As patient samples frequently contain a dominant variant with many more low prevalence variants, this implies that minority variant sequences detected by NGS are likely to be chimeric. For high prevalence templates, the chance that the partial product will prime an identical template is high, so that the detected level of recombination is low. Even though the homology between the sequences in this dataset was low, the effect of prevalence on detectable recombination was still very high.

We explored the relationship between the chimerism rate in the consensus sequences and the chimerism rate in the raw sequences and the composition of the sample (Figure 60). Counter-

intuitively, as the chimerism rate in the raw sequences increases, the chimerism rate in the consensus sequences decreases. However, the composition of the sample has a strong influence on this relationship where samples with a more equal distribution of the three variants have the highest raw recombination rates and the lowest consensus recombination rates. Having a single low prevalence variant and a single high prevalence variant serves to suppress the chimerism rate in raw sequences because recombination is primarily between templates both originating from the high prevalence variant. Simultaneously, it amplifies the chimerism rate in the consensus sequences since the probability of chimeric recombination in low prevalence variants is high enough to introduce multiple chimeric recombination events into a single bin from which a consensus sequence is constructed. The recombination rate in the consensus sequences was low, with the exception of the 2dPCR protocol, implying low power to measure any relationships involving the recombination rate in the consensus sequences.

Recombination events can only be detected at positions where the sequences differ from each other. This implies that each detectable recombination event can only be assigned to an interval in which it occurred and not to a precise location. The homology between two sequences for such a recombination interval can be described as the length of the recombination interval plus the lengths of the two such intervals flanking it. To investigate if there were specific recombination hotspots, we assigned each detectable recombination event to an interval and plotted the rate at which detectable recombination events occurred by position in the sequence as shown in Figure 61. At the 5' end, CAP210 and CAP239 have very high homology resulting in wide boxes on the left hand side of the 210_239 column in Figure 61. The height of the boxes show the recombination rate within that interval, with higher boxes indicating higher recombination rates. The y-axis is scaled so that it shows the likelihood that in a chimeric sequence, the recombination event will occur at the specific position (assuming that recombination event occur uniformly within the recombination intervals). The boxes are colored according the homology score calculated as the sum of the length of the recombination interval and the lengths of the intervals flanking it.

Since recombination is driven by partially amplified PCR products acting as primers we expect the recombination rate to be elevated in regions of high homology between the different variants. The higher homology makes it more likely for a sequence to act as a primer since hybridization is more likely to occur between similar sequences than divergent sequences. This relationship is present in Figure 61, in areas where the bars are light blue (signifying high homology), the bar are also high (indicating a high likelihood of recombination) as can be seen around position 80. Conversely, when the bars are dark blue (signifying low homology), the bars are also low (indicating a low likelihood of recombination) as can be seen around position 200. Figure 62 shows the relationship between the

homology score and the likelihood of recombination more directly. While the data is noisy, there is a clear trend for a higher homology score to be associated with a higher likelihood of recombination events occurring.

By constructing the dataset in which each sequence was classified as either originating from one of the input templates or as resulting from a recombination event in a certain interval between two of the input templates we were able to investigate PCR induced recombination. Constructing consensus sequences drastically reduced the percentage of sequences that are flagged as chimeric (over 5% of raw sequences, but less than 0.03% of consensus sequences). Recombination is strongly influenced by the prevalence of the variants, with sequences from minority variants being at higher risk of being recombinants. Lastly, recombination events are concentrated in regions of high homology. Since many patient samples contain a complex quasispecies with many closely related minority variants we expect the chimerism rates in patient samples to be higher than we observed here.

Table 28: Recombination rates presented by protocol, elongation time and sequence type. The total number of sequences across all samples are listed (In the Total columns) for each of the eight protocols further split by raw and consensus sequences. The number of sequences that were classified as recombinants are listed (In the Recombinants columns) as well as what percentage of all the sequences were classified as recombinants (shown in parentheses).

Seq. Type	Protocol	Short Elongation Time		Long Elongation Time	
		Total	Recombinants	Total	Recombinants
Raw	nrPCR	1082481	126158 (11.65%)	1254552	104543 (8.33%)
Raw	rcPCR	709230	70084 (9.88%)	859162	79808 (9.29%)
Raw	mxPCR	794737	70908 (8.92%)	692394	45408 (6.56%)
Raw	2dPCR	670688	100229 (14.94%)	467476	60580 (12.96%)
Cons.	2dPCR	16992	74 (0.44%)	18169	45 (0.25%)
Cons.	nrPCR	34093	7 (0.02%)	34808	7 (0.02%)
Cons.	rcPCR	32918	3 (0.01%)	33943	4 (0.01%)
Cons.	mxPCR	17693	2 (0.01%)	19089	2 (0.01%)

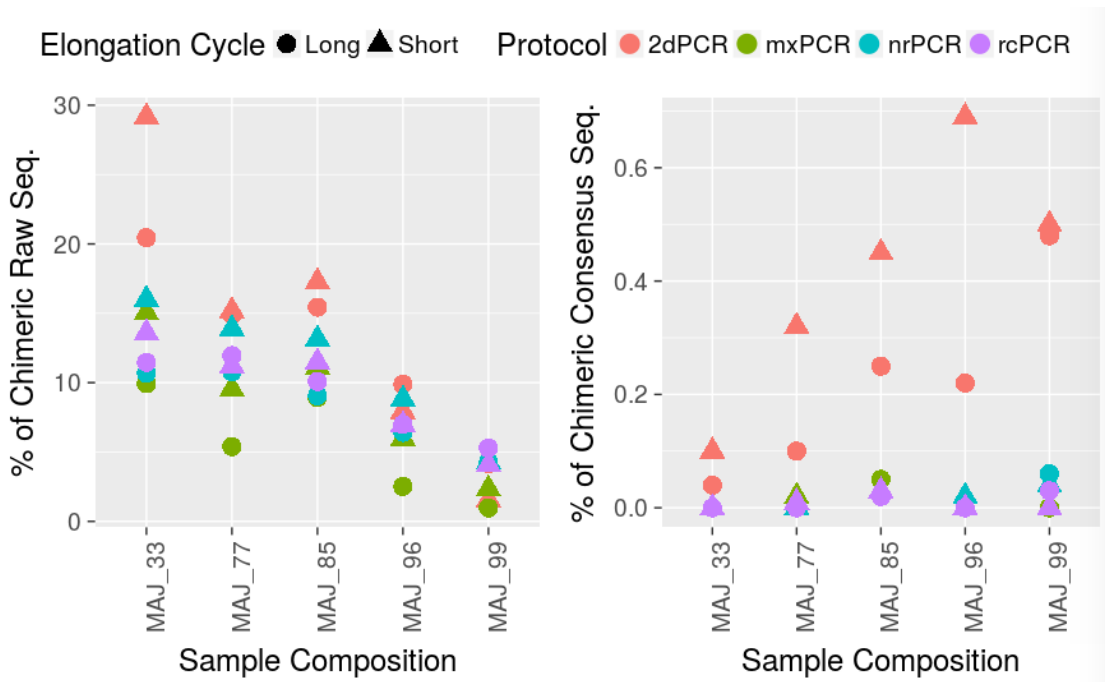


Figure 57: The percentages of raw (left) and consensus (right) sequences that are recombinants. Due to the large difference in the prevalence of chimeric sequences between raw and chimeric sequences, the ranges on the y-axes were not restricted to the same values.

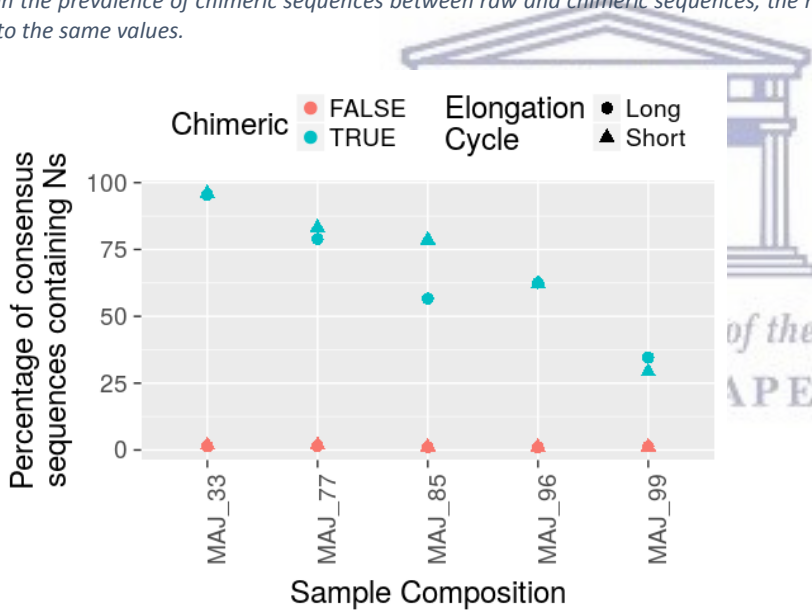


Figure 58: The percentages of chimeric and non-chimeric consensus sequences that contain Ns.

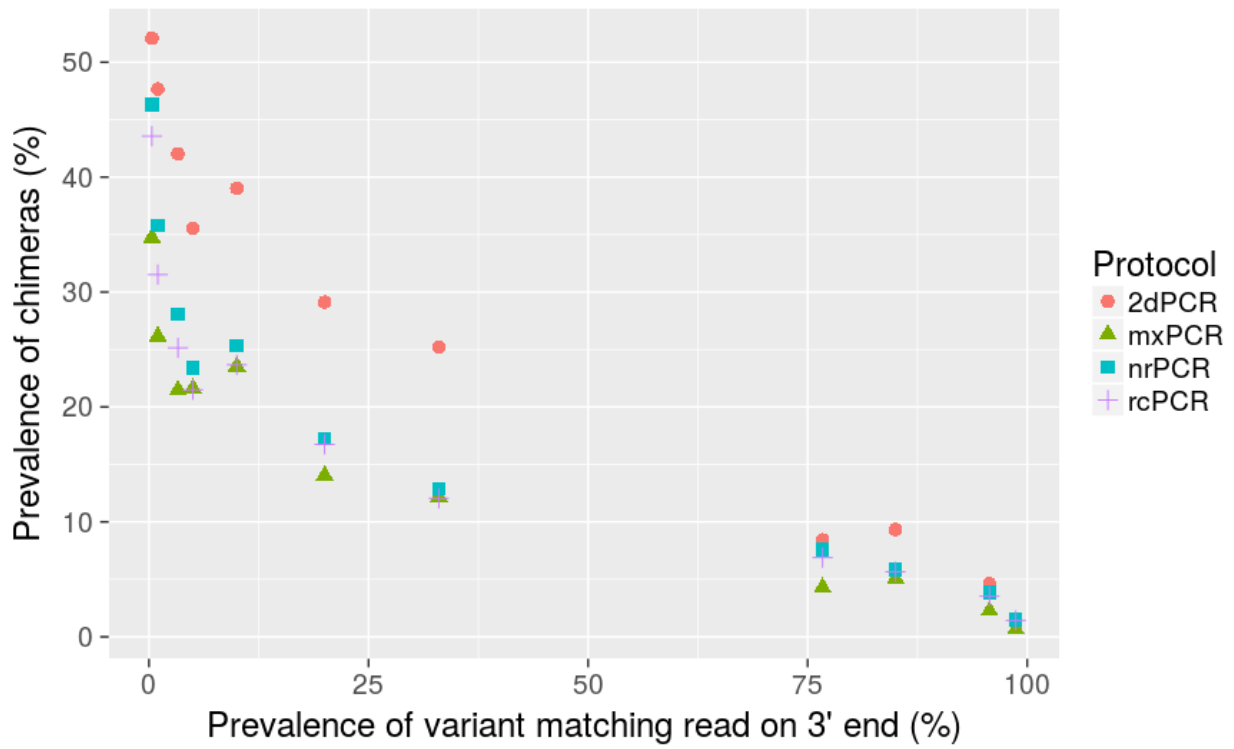


Figure 59: The percentage of raw reads that are recombinants based on the prevalence of the variant that matches the 3' end of the read.



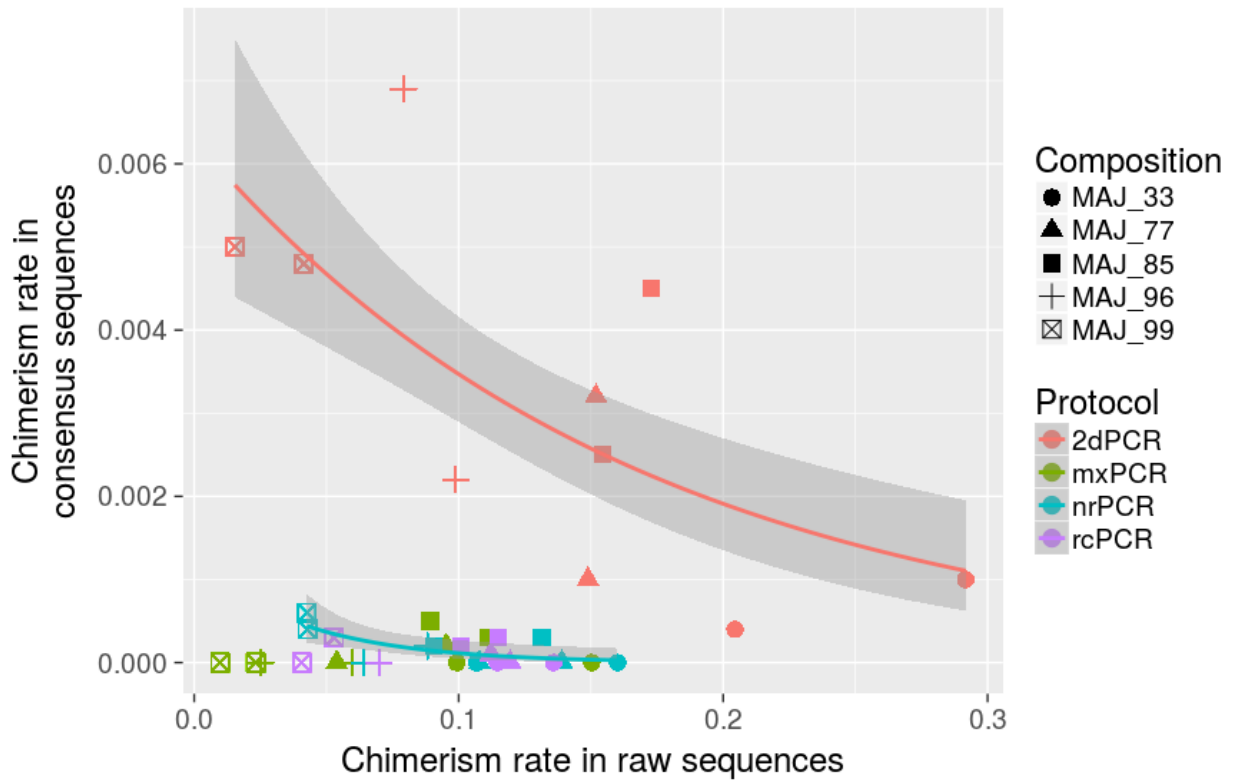


Figure 60: Relationship between the recombination rate in the raw and consensus sequences presented by sample preparation protocol. Trend lines added using binomial regression. Recombination rates in the consensus sequences of the mxPCR and rcPCR protocols are too low to model.



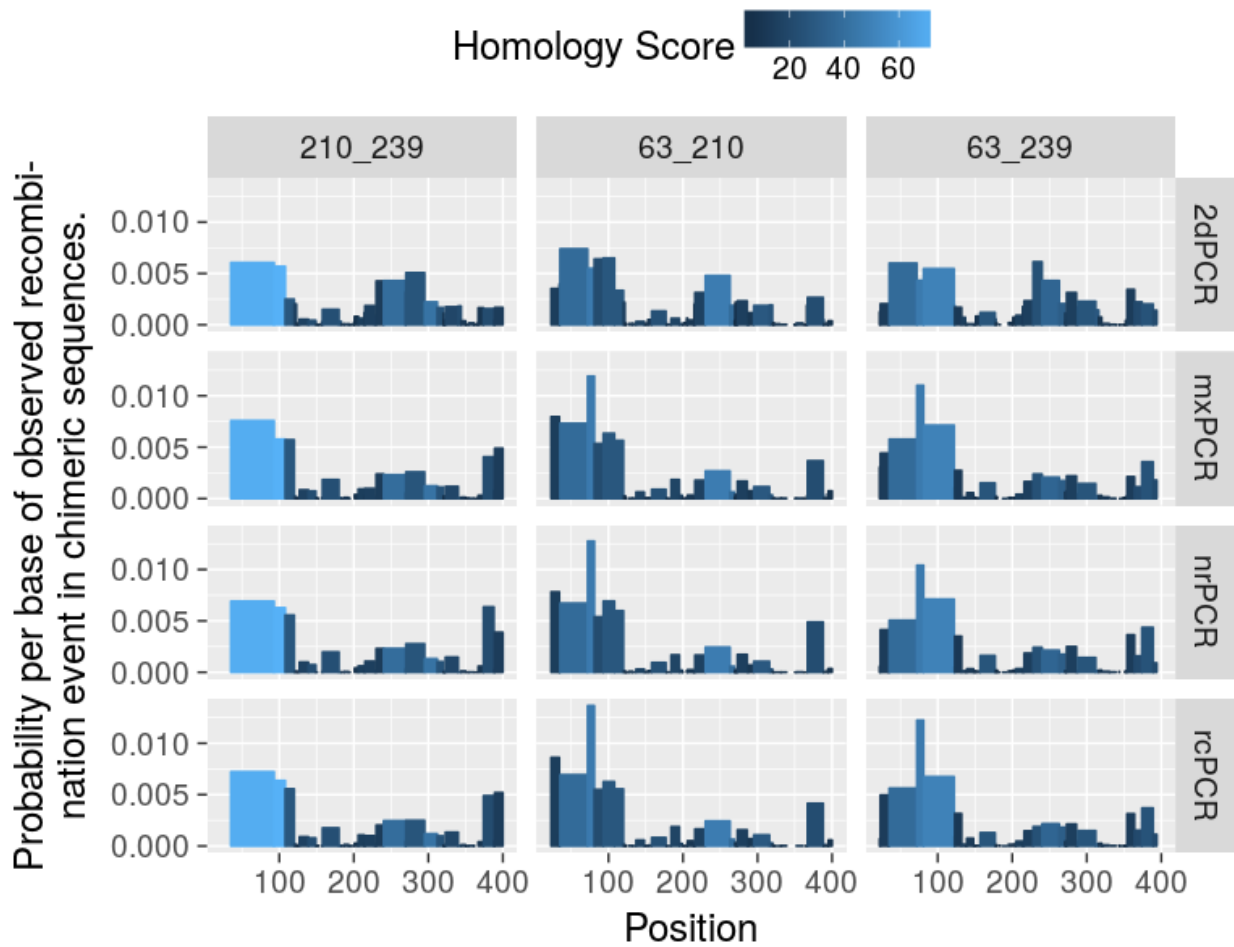


Figure 61: Likelihood of recombination events occurring in raw recombinant sequences by position. Each column presents data for a specific type of chimeric sequence, the first column show chimeras composed of CAP210 and CAP239. Color indicates the homology between the two sequences in the area surrounding the interval.

UNIVERSITY of the
WESTERN CAPE

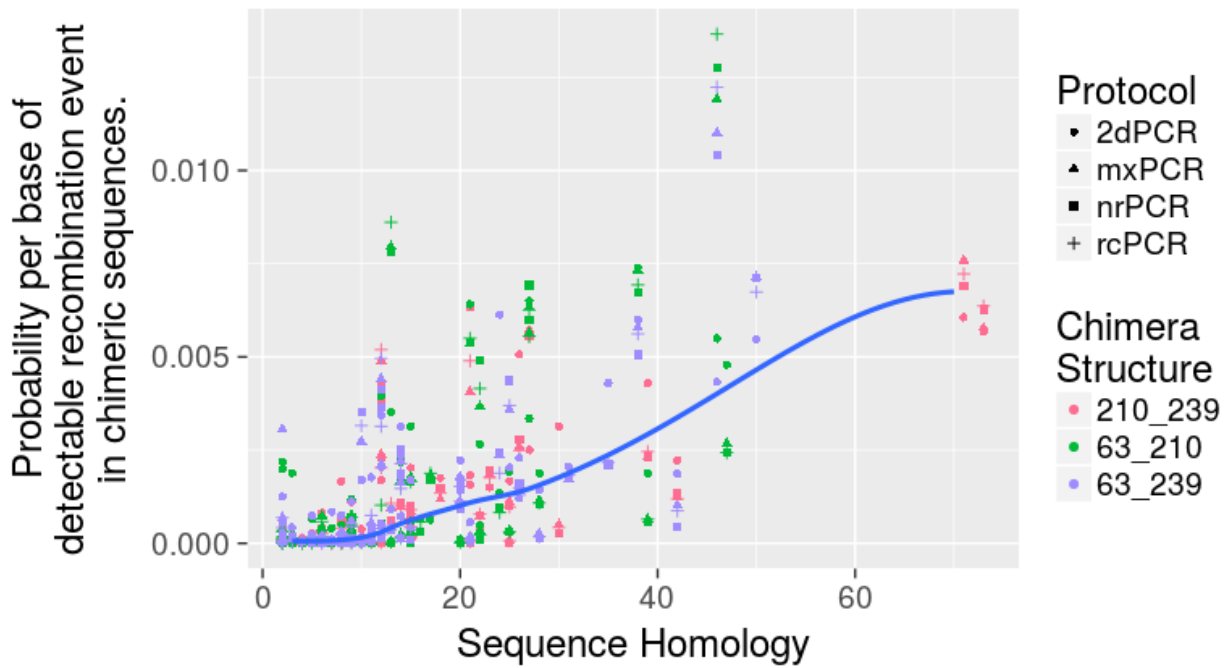


Figure 62: The relationship between the local homology around an interval to which a recombination event can be assigned (x-axis) and the probability per base of a recombination break occurring at that position in raw recombinant sequences (y-axis). Trend line added using loess regression.



4.2.6 The loss of minority variants in consensus sequences at the 5' end was minimal.

A potential limitation of the PID approach is that the generation of consensus sequences may remove variations on the 5' end that are unique to minority variants from the dataset. As the distance between two positions increase, the probability that a recombination event occurred between them increases. Thus, when considering minority variants, positions on the 5' end (furthest from the PID) are likely to be from majority variants (Figure 59). This is because the large distance means that it is likely that a recombination event occurred between these positions and the PID, and since the majority variants are highly prevalent, it is likely that the minority variant recombined with a majority variant. When considering consensus sequences, this effect has the potential to hide minority variants (Boltz et al., 2016). The prevalence of each variant was established at each of the 12 positions where none of the three variants matched each other. The loss of minority variants at the 5' end of the consensus sequences was not seen in these datasets (Figure 63), except for minor decreases in prevalence in the extremely low prevalence variants in the 2dPCR protocol (Figure 64).

A potential technique to reduce the impact of such a loss of minority variants would be to perform detailed analyses across all bins at specific positions. If the prevalence of a minority variant is suppressed due to chimera formation, one expects to observe a number of bins in which the minority variant is present at levels insufficient to make it into the consensus sequences, but high enough to be unlikely to be explained by sequencing error. This dual approach, comparing data present in the raw sequences only with that provided by uniquely tagging each molecule, may be useful in limited cases where a large amount of PCR induced recombination might obscure minority variants. However, it comes at the cost of a high false positive rate as illustrated by the examples that follow.

Consider position 36, where CAP210 has the letter 'A' mismatching the other two variants and is only prevalent at 0.33% in sample MAJ_99 as presented in Table 29. In the 2dPCR protocol with short elongation times, only 2 consensus sequences in the analysis dataset had an 'A' at this position. Another 4 consensus sequences also had an 'A' at this position, but they were excluded from the final dataset because these 4 consensus sequences contained 'N's at other positions. Furthermore, if one were to scan for bins such that the number of 'A's at position 36 are such that that only 1 in 10000 bins would contain that number of 'A's due to sequencing errors alone, then 27 such bins are found. Taken together, the consensus sequences removed due to the presence of 'N's and the bins with over represented levels of 'A's at position 36, it appears that prevalence of this minority variant was suppressed (from 27 sequences to 2 sequences) by recombination in this dataset. However, using this approach with the same 1 in 10000 bins cutoff for the error rate, one would also find that 7 bins contained 'T's at position 35 and 4 bins contained 'C's at position 37, neither of which match any of

the input variants and are thus erroneous. Similar results are obtained when considering this position for the rcPCR with a short elongation time dataset.

The utility of this minority variant recovery technique diminished greatly as the distance to the PID shrinks. Consider position 296 where CAP210 has the letter 'G' mismatching the other variants and is only prevalent at 0.33% in sample MAJ_99 as presented in Table 30. In the 2dPCR protocol with short elongation times, only 10 consensus sequences had a 'G'. Another 8 consensus sequences also had a 'G' at this position, but they were excluded from the final dataset because they contained 'N's at other positions. Furthermore, if one were to scan for bins such that the number of 'G's at position 296 are such that that only 1 in 10000 bins would contain that number of 'G's due to sequencing errors alone, then 18 such bins (the 10 already in the dataset and the 8 that formed the bins that yielded the consensus sequences with N's) are found. However, when using this approach, in positions 294 to 298 alone, 140 mistakes would be introduced (Table 30).

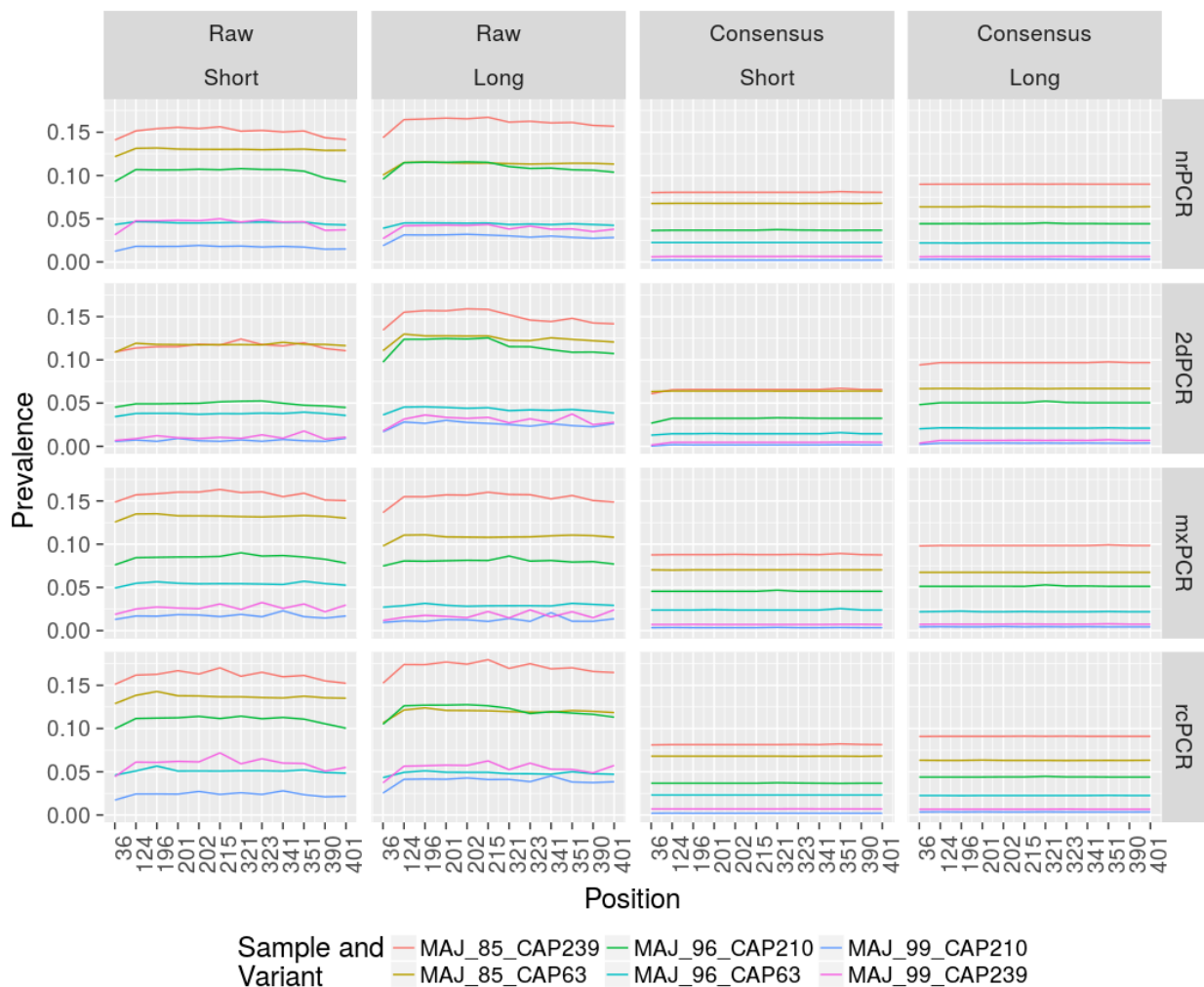


Figure 63: Prevalence by position of minority variants. Only positions where all three variants are distinct from each other are shown.

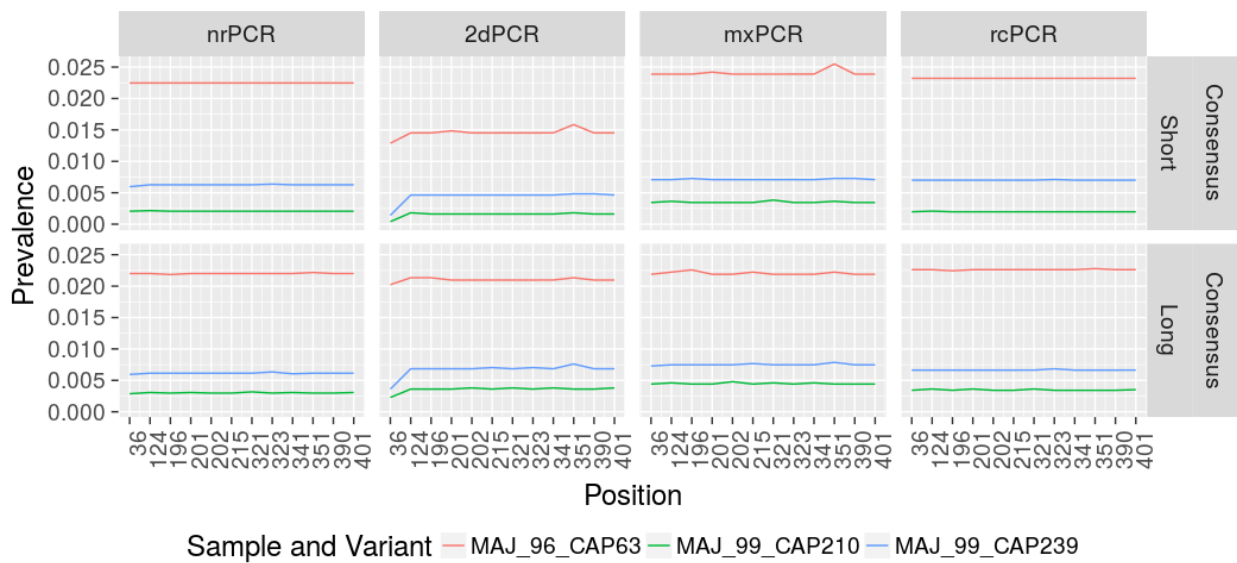


Figure 64: Prevalence by position of the extremely low prevalence minority variants. Only positions where all three variants are distinct from each other are shown.



Table 29: Number of bins in which the stated letter (Column 3) occurs at the specific position (Column 2), close to position 36, at a frequency whose probability of arising due to sequencing error is less than the given probability for the sample MAJ_99. The row highlighted in blue shows a correct call. Position 36 should indeed be an A. The cells highlighted in red, show errors that would be made at the 1 in 10000 cutoff. Using this cutoff, 3 bins will be flagged as having an A at position 34, and there should not be an A at position 34.

Protocol	Position	Letter	Chance of resulting from sequencing error alone is smaller than						
			10^0	10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}
2dPCR with short elongation time	34	A	5078	3461	39	9	3	1	1
	34	G	5078	3454	14	1	0	0	0
	34	T	5078	3511	110	17	2	1	0
	35	G	5078	3462	38	5	1	1	1
	35	T	5078	3574	215	46	7	2	0
	36	A	5078	3626	341	87	27	17	17
	36	T	5078	3465	25	2	0	0	0
	37	A	5078	3486	65	15	1	0	0
	37	C	5078	3580	237	51	4	2	1
	37	G	5078	3489	70	16	2	1	1
	38	C	5078	3445	17	3	0	0	0
	38	G	5078	3550	193	54	4	1	1
	38	T	5078	3465	40	8	1	0	0
rcPCR with short elongation time	34	A	9293	9210	139	68	3	0	0
	34	G	9293	9202	18	9	0	0	0
	34	T	9293	9204	47	31	0	0	0
	35	G	9293	9205	42	22	1	0	0
	35	T	9293	9202	84	44	3	0	0
	36	A	9293	9233	528	298	43	25	21
	36	T	9293	9203	67	38	0	0	0
	37	A	9293	9201	33	12	0	0	0
	37	C	9293	9205	54	22	2	1	1
	37	G	9293	9201	33	20	1	0	0
	38	C	9293	9222	261	128	7	1	0
	38	G	9293	9208	127	65	2	0	0
	38	T	9293	9205	76	43	0	0	0

Table 30: Number of bins in which the stated letter (Column 3) occurs at the specific position (Column 2), close to position 296, at a frequency whose probability of arising due to sequencing error is less than the given probability for the sample MAJ_99. The row highlighted in blue shows a correct call. Position 296 should indeed be a G. The cells highlighted in red, show errors that would be made at the 1 in 10000 cutoff. Using this cutoff with rcPCR protocol with short elongation times, 52 bins will be flagged as having an A at position 296, and there should not be an A at position 296.

Protocol	Position	Letter	Chance of resulting from sequencing error alone is smaller than						
			10^0	10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}
2dPCR with short elongation time	294	C	5078	3460	32	8	1	1	1
	294	A	5078	3612	256	48	9	4	3
	294	T	5078	3499	96	31	17	15	14
	295	G	5078	3714	459	100	21	8	6
	295	C	5078	3514	93	14	4	1	1
	295	T	5078	3489	82	16	4	0	0
	296	A	5078	3620	286	71	22	16	15
	296	G	5078	3536	150	40	18	18	18
	296	T	5078	3504	109	19	5	2	2
	297	C	5078	3586	213	39	5	2	2
	297	T	5078	3523	144	32	16	13	13
	297	G	5078	3712	420	106	17	6	3
	298	C	5078	3488	55	7	0	0	0
	298	G	5078	3647	318	70	17	5	3
	298	T	5078	3471	52	9	2	1	1
rcPCR with short elongation time	294	C	9293	9210	131	74	5	0	0
	294	A	9293	9219	434	224	22	8	4
	294	T	9293	9233	599	312	43	16	12
	295	G	9293	9207	144	67	1	0	0
	295	C	9293	9206	103	54	6	1	0
	295	T	9293	9206	76	41	2	0	0
	296	A	9293	9226	601	341	52	23	15
	296	G	9293	9228	635	338	41	27	20
	296	T	9293	9212	68	35	4	0	0
	297	C	9293	9210	171	99	3	1	0
	297	T	9293	9205	67	43	13	12	12
	297	G	9293	9209	98	57	2	0	0
	298	C	9293	9202	30	13	0	0	0
	298	G	9293	9201	66	28	1	0	0
	298	T	9293	9202	14	10	1	0	0

4.3 Conclusion and Future Work

The amount and quality of data produced by amplicon-based NGS, is affected by the amplification bias and the amount of recombination occurring during PCR. We attempted to reduce these effects by using droplet instead of standard PCR, reducing the number of PCR cycles and by increasing PCR elongation time. We found that PCR recombination rate is strongly affected by the sample composition and the homology between the sequences. Using droplet PCR for the first round and normal PCR for the second round had the lowest recombination rate, but this protocol picked up 50% less unique templates than the protocols involving only normal PCR. Additionally, the non-recombinant consensus sequences produced by droplet PCR protocols had higher error rates. Future work on using droplet PCR to reduce PCR induced recombination might benefit from exploring less stable emulsions.

The effect of longer elongation times was minimal. We were able to show that prevalence estimates based on consensus sequences are less variable than estimates based on raw sequences. However, we were unable to measure bias due to the high variability of the viral loads used to construct the samples. Finally, we show that the PID approach not only corrects PCR and sequencing errors but also corrects PCR recombination extremely efficiently and that bins as small as size 10 can yield highly accurate consensus sequences. Overall our data supports the use of this approach in NGS as it enables high-throughput sequencing of viral populations with high levels of accuracy.

In real world samples, the chimerism rates are expected to be much higher than what was found in this study. Real world samples contain many minority variants which we showed to be at much higher risk of chimerism formation. Amplifying and sequencing from blood plasma also poses extra challenges not encountered when working with clones with known sequences. Hence one cannot be certain that the primers will bind with high affinity causing uncertainty in the amplification efficiency. Taken together, the large numbers of different minority variants and the additional amplification and sequencing challenges imply that controlling PCR recombination should be a priority.

The recommended approach will thus depend on the ultimate goal of one's study. If the goal is to detect as many minority variants as possible, then two rounds of normal PCR should be used. The number of cycles of PCR should be carefully optimized to ensure that the reaction is stopped prior to saturation or depletion of reagents. If the goal is to recover sequences with the explicit goal of minimizing the probability that any sequence is the result of PCR induced recombination, then the mxPCR protocol may be considered which uses dPCR for the first round and normal PCR for the second round. Again, optimizing the PCR reaction to ensure that they stop before saturation or depletion of the reagents will decrease error and recombination rates. This approach reduces PCR recombination,

but yield is suppressed and the rate of non-recombination errors are increased. Lastly, if the goal is to search for extremely rare (1% or below) single nucleotide polymorphisms (SNP) at known positions, then a re-analysis of the data, scanning for bins in which that specific SNP is present at levels higher than what is expect due to sequencing error is a method to improve detection. However, careful exploration of the false positive rate will be required when using this technique. Lastly, increasing the elongation times to 10min yields little improvement, suggesting that for most cases using 2.5min elongation times will yield adequate results.



5 Infection Timing

The ongoing AMP studies (HVTN703 <https://clinicaltrials.gov/ct2/show/NCT02568215> and HVTN704 <https://clinicaltrials.gov/ct2/show/NCT02716675>) infuses HIV-negative patients with the VRC01 antibody (T. Zhou et al., 2010) and monitors for HIV infections. Accurately timing the infection event and reconstructing the founder viruses of these infections are critical for relating infection risk to antibody titer and homology between the founder virus and antibody binding sites (L. Zhang et al., 2018).

In the case of a single founder and no selective pressure, a star like phylogeny arises, under which the time of infection can be computed from the number of HIV replication cycles (number of generations) that has occurred since the infection. The generation number is related to the distribution of the inter sequence Hamming distances which follows a Poisson distribution under a star like phylogeny. In the presence of selective pressure or multi-founder infection, the inter sequence Hamming distances will diverge from the Poisson distribution. Hence, it is possible to test for multiplicity of infection by comparing the inter sequence Hamming distances to a Poisson distribution (Elena E Giorgi et al., 2010).

Several biological processes can affect the data leading to inaccurate results. Hypermutation artificially inflates the mutation rate leading to over estimates since a star like phylogeny implies that the current population all arose from a single individual with no selective pressure (Rose & Korber, 2000). When assuming a tree-like data structure, as most phylogenetic techniques do, recombination can lead to serious complications (Schierup & Hein, 2000). However, the Poisson Fitter approach targets datasets before the tree-like structure arises so the effect of recombination is unclear. Lastly, the generation number is influenced by the number of mutations that occurred, but the mutation rate varies along the HIV-1 genome requiring calibration based on the amplicon location or sophisticated models allowing for position specific mutation rates (Zanini et al., 2015).

The AMP trial is a large collaborative project comprising multiple research teams spread worldwide. As part of the founder virus characterization component Dr Paul Edlefsen at the Fred Hutch Cancer Research Institute developed a pipeline that performs infection timing and founder reconstruction. Continuing on with this project we produce detailed tests for a portion of the pipeline, investigate the robustness of some of the external tools used in the pipeline to violations in the assumptions used during development and produce comprehensive documentation of the pipeline.

5.1 Overview and Design

The goal of the pipeline is to estimate the time since infection, the multiplicity of infection and the sequence(s) of the founder virus(es) from a single alignment sequenced from viral RNA isolated from

blood plasma of a single patient at a single time point. A diagram of the full pipeline is shown in Figure 65. The middle column of the diagram shows all of the programs and scripts used to process the data and perform the estimates. A large number of intermediate results are produced allowing more detailed investigation and enabling more nuanced masking and grouping of data to deal with multiple founder infection and selective pressure.

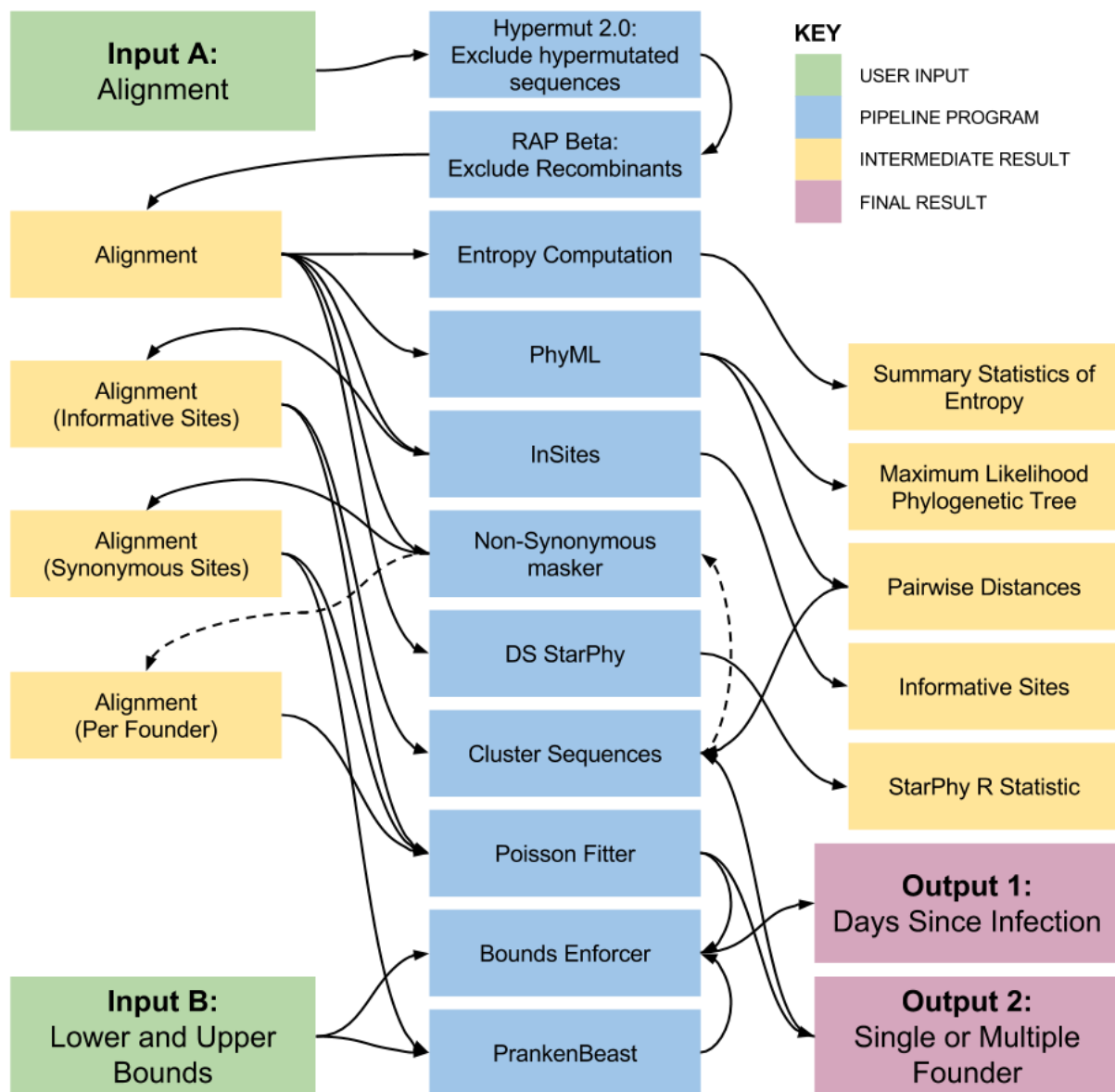


Figure 65: Diagram of the HIV founder pipeline.

The pipeline is implemented in perl. A single main script `identify_founders.pl` contains a main loop iterating over all the input files. Each iteration processes a single time point for a single patient. The variable `fasta_file` contains the path and name to the file containing the data for the current iteration. As the data is manipulated, this variable is updated to always reflect the current dataset that

is analyzed. For example, if hypermutated sequences are removed, then a new file is produced that excludes all sequences with hypermutations and the `fasta_file` variable is updated to point to this new file, leaving the original in place, producing an audit trail.

Relevant metrics are printed to the `identify_founders.tab` file as soon as they are available. This file reports the main results. It has a large number of columns that depends on which steps the users decides to run. For example, if the step dealing with hypermutation is run with the setting to “fix” instead of remove hypermutated sequences, then the file will contain a column called ‘fixed-hypermut’ as opposed to the column ‘removed-hypermut’.

Each program listed in Figure 65 has a wrapper script written in either R or perl that calls the required programs and formats the data. These scripts are called using system calls from perl implemented using the back tick symbol based syntax. Information, including the `fasta_file` variable, is passed to these wrappers by setting environmental variables. Information is passed back to the main script via STDOUT and by producing output files in specified locations.

In this thesis, only a portion of this pipeline will be explored and some alterations to the pipeline will be proposed. The recombination removal step will be removed from the pipeline. When dealing with short sequences, such as those proposed by Illumina’s MiSeq, the power to detect recombination is very low. Additionally, recombination is a constant feature in the HIV quasispecies. The step that removes the hypermutation has utility in many other applications, so we split it off of the pipeline and produce an R package that implements the algorithm. A detailed investigation of the PhyML step that produces a pairwise diversity metric reveals that the same result can be obtained using much simpler processes such as a Hamming distance based metric. Lastly, this thesis only considers the case of single founder datasets, thus excluding all the components designed to process multi-founder datasets.

5.2 hypermutR

Hypermutation is a phenomenon that affects HIV-1 by introducing large numbers of mutations into some sequences. It manifests in the datasets as sequences in which large numbers of Guanine was mutated to Adenine, specifically when that Guanine was surrounded by a particular pattern (Rose & Korber, 2000). The Hypermut 2.0 tool available from <https://www.hiv.lanl.gov/content/sequence/HYPERMUT/hypermut.html> is a frequently used tool to detect and remove hypermutated sequences. However, this tool is only available as a web interface, making batch processing of large volumes cumbersome. Since this tool is useful outside this pipeline, we decided to remove it from the pipeline and implemented the Hypermut 2.0 algorithm in an R package called hypermutR.

5.2.1 Algorithm

The Hypermut algorithm compares each sequence in an alignment to some ancestral sequence (usually approximated by the consensus sequence of the alignment), tallying the frequency of specific mutations. Hypermutation occurs when a G which is followed by an A or G (denoted by R in the IUPAC convention) and then by an A, G or T (denoted by a D in the IUPAC convention) mutates to an A. More compactly, when GRD become ARD, the mutation is flagged as possibly due to hypermutation. In order to distinguish between true hypermutation and the generally expected level of mutation, a baseline must be established. The baseline is established by tallying G to A mutations when the G is followed immediately by either a C or T (denoted by a Y in the IUPAC convention) or when the G is followed by an A or a G (denoted by R in the IUPAC convention) and then a C. More compactly, when GY becomes AY or GRC becomes ARC, the mutations are tallied as the baseline mutation rate against which the potential hypermutations must be compared.

A one-sided Fisher's exact test is used to compare the proportion of GRD positions that became ARD positions to the proportion of GY or GRC positions that became either AY or ARC positions. When the p-value of the test is smaller than some threshold, with the default set to 0.1 as in (Abrahams et al., 2009), then the individual sequence is flagged as a hypermutant and (depending on user input) either the sequence is removed from the dataset, or the mutated bases (the A's followed by RD) are replaced by an R to indicate that we are uncertain whether the mutation was a random mutation or if it was the result of hypermutation.

In order to be a position of interest (either a control or hypermutation position), all that is required is a G in the ancestral sequence. To classify the position into either a hypermutation or control position, only the query sequence is considered. If the two positions following the position that contains the G in the query sequence matches RD, then it is a hypermutation position, else it is a control position.

The two downstream positions in the ancestral sequence are not considered. Note that, since the mechanism that induces hypermutation only affects positions that match GRD, the fact that only the query sequence is considered, means that if the downstream positions in the ancestral sequence does not match RD, then the Hypermut algorithm implicitly assumes that those downstream positions in the query sequence derived from the ancestral sequence by first mutating these positions to match RD, and then the hypermutation mechanism operated on the GRD signal to mutate the G into an A.

5.2.2 Implementation

The implementation of the algorithm can be found in the hypermutR R package. It has a command line interface built with the optparse package (Davis, 2017) providing control over 5 variables (Table 31). The `remove_hypermutation` function is a wrapper that calls `ancestor_processing` to obtain the ancestral sequence to compare the query sequences to, calls `deduplicate_seqs` to remove duplicate sequences for performance reasons, then loops over each unique sequence, comparing it to the ancestral sequence with `scan_seq` and finally collates the results. [Detailed installation and usage instructions are provided in appendices 8.4 and 8.5.](#)

Table 31: Parameters that can be controlled by the command line user interface of the hypermutR package.

Name	Description
<code>input_file</code>	String specifying the path to and the name of the fasta file containing the alignment of the sequences.
<code>ouput_file</code>	String specifying the path to and the name of the file that will contain the resulting sequences (with hypermutated sequences either removed or corrected).
<code>p_value</code>	The threshold to use when deciding if the p-value produced by the Fisher test indicates that there is hypermutation present in the sequence.
<code>ancestor</code>	Either 'consensus' to indicate that the consensus sequences must be computed, or 'first' to indicate that the first sequence in the dataset should be considered to be the ancestral sequence, or the ancestral sequence itself.
<code>fix_with</code>	If omitted, hypermutants will be removed. If a single letter is specified, then hypermutants will be corrected by replacing the hypermutated base with the specified letter.

The package is designed to depend exclusively on packages from CRAN (and none from Bioconductor), meaning that the `seqinr` (Charif & Lobry, 2007) package is used to read and write fasta

formatted files. The `seqinr` package stores sequence data in objects of class `SeqFastadna`. Formatting data as `SeqFastadna` objects yields one of two configurations. The first configuration is a vector of character strings, in which each character string is a sequence, with the optional attributes: `names`, `Annot`, and `class`. The alternate structure is a `list` in which each element represents a single sequence. Each element consists of a vector of single letters of class `character` with the same optional attributes as the first configuration. The `seqinr` package provides fasta file access with the `read.fasta` function which stores data in the second format, the `list` of vectors of single letters. For consistency, `hypermurR` also uses the `list`-based format to store sequence data.

Three options exist for specifying the ancestral sequence to compare the query sequences in the dataset to. If the value 'consensus' is specified via the `ancestor` parameter, a consensus sequence will be computed from the sequences in the input file. The letter that most frequently occurs is placed in the consensus sequence. In the case of ties, the first letter, when arranged alphabetically, is used. If at a specific position the true, but unknown, ancestral sequence contained a G, but hypermutation changed more than 50% of the sequences at that position into an A, then this consensus approach will assign an A to that position in the sequence used as an ancestral reference sequence. The second option is to include the ancestral sequence as the first sequence in the input file and to set the value of the `ancestor` parameter to 'first'. In this case, the first sequence will be removed from the dataset before proceeding. Lastly, the `ancestor` parameter can be assigned the ancestral sequence itself. The only validation that is performed on the last of the three options is to check that the sequence assigned to `ancestor` has the same length as the sequences in the input file.

The `scan_seq` operation is slow and is entirely deterministic (so yields the same result for duplicate inputs), so the dataset is deduplicated with the `deduplicate_seqs` function to improve performance. The dataset is converted to a vector of character strings and the unique sequences are selected with the `unique` function. Looping over the unique sequences, a `list` is constructed in which each element corresponds to a unique sequence. Each element is also a `list` with the elements `the_seq` containing the actual sequences and `dup_names`, a vector of character strings listing the names of all sequences that matches the unique sequences stored in `the_seq`.

The `remove_hypermutation` function loops over the unique sequences returned by the `deduplicate_seqs` function. On each unique sequence, the `scan_seq` function is called. The ancestral sequence provided by the `process_ancestors` function is also passed to the `scan_seq` function. The `scan_seq` function simultaneously passes two sliding windows along the ancestral and query sequences. The sliding window is of length 3, corresponding to the potentially hypermutated position and the 2 downstream positions.

At each position, the size of the window is increased until it covers 3 non-gap characters in the query sequence. If a G is located at the first position of the window, the position is considered a position of interest and the query sequence is inspected to classify it as either a hypermutation or control position, incrementing either the `num_potential_mut` variable or the `num_potential_control` variable. The query sequence is checked next and if the G mutated to an A, then the tally of the number of possible hypermutations (`num_mut`) or the number of control mutations (`num_control`) is incremented.

The return value from `scan_seq` is a `list` that contains the number of mutated hypermutation and control positions, the total number of potential hypermutation and control positions, the p-value of the one-sided Fisher's exact test, the (possibly corrected) query sequence and the `data.frame` that catalogs each individual position. As noted above, the return sequence will only be corrected if the `fix_with` option is specified (typically we choose "R" for this option to indicate residual uncertainty between "A" and "G" in these cases, but the `fix_with` value might instead be specified as "G" to undo the hypermutation change).

The `remove_hypermut` function binds the `data.frames` that catalog each position together into a full log, called `all_mut_pos`, of all positions of interest in all sequences. After comparing the p-value to the p-value cutoff passed into the `remove_hypermutation` function, each sequence is stored in either a `list` that contains all hypermutated sequences (`seq_hypermutants`) or a `list` that contains all non-hypermutated sequences (`seq_result`). The `remove_hypermut` function returns these three results: `all_mut_pos`, `seq_result`, and `seq_hypermutants`.

The user interface (UI) script, `hypermut.R`, located in the `inst` folder in the package root, writes the return values from `remove_hypermut` to disk. The value of the `input_file` parameter dictates the file name used for `seq_result`. The `'fasta'` extension on the value of `input_file` is replaced with `'_hypermutants.fasta'` to construct the file name for `seq_hypermutants`. Lastly, the file name for the `all_mut_pos` `data.frame` is obtained by replacing the `'fasta'` extension of the `input_file` parameter with `'_mut_pos.csv'`.

5.2.3 Tests

The `hypermutR` package has a full suite of unit tests built with the `testthat` package. As per the guidelines of `testthat`, the testing code is located in the `tests/testthat/` subfolder of the package root. The modular design of `hypermutR` allows the construction of tests that precisely test the functioning of small specialized pieces of code. The organization of the tests mirror that of the code, with matching file names, but `'test_'` prepended to the names of the files that contain the test code. The contents of each test file is organized hierarchically into `contexts`, `tests` and

expectations (Wickham, 2011). An expectation is a single simple requirement that a return value of one of the functions of hypermutR must meet. For example, the class of the return value from the `remove_hypermut` function must be `list`. Expectations that cover a set of tightly related operations are grouped together into tests. Tests are further grouped into contexts which provides extra information to help locate the code covered by the context in question.

Each function in hypermutR, except those designed to simulate test scenarios, are covered by a number of expectations checking the format of the output as well as the correctness of a sample of the elements of the return value. A number of tests check the result of applying the wrapper function `remove_hypermut` to the `ld_seqs` and `hd_seqs` data sets, described later in section 5.4.4.4, in which some sequences were hypermutated. These tests serve as integration tests ensuring that the entire process of removing hypermutated sequences works as expected.

Hypermutation is simulated with the `sim_hyper` function. Given a sequence dataset, `sim_hyper` will mutate a specified number of hypermutation and control positions in a given number of sequences. The number of sequences in which to mutations are to be introduced is specified by the parameter `n1`. The `n2` and `n3` parameters control the number of hypermutation and control positions to mutate respectively. Each of the parameters may be between zero and one to specify a proportion of sequences or positions. If the parameter values are equal to or larger than one then they specify the exact number of sequences or positions. The parameter can also be assigned the value 'all' in which case it will signify that all the positions or sequences should be affected. The return value is a named vector of type `atomic` assigned the class `SeqFastadna` in which each element of the vector is a DNA sequence.

5.2.4 Benchmarks and Comparisons

The Hypermut tool on the LANL website is a well-established tool with more than 200 citations. To ensure that the implementation of hypermutR is correct and faithfully reproduced the results of the latest version of the Hypermut tool on the LANL website and to document any discrepancies, a large number of edge cases were constructed and processed with both hypermutR and the Hypermut 2.0 tool. The results of this comparison are shown in Table 32. Very short sequences were chosen so that each result could be computed manually to ensure that the result is consistent with the description of the algorithm.

The construction of the windows to compare between the ancestral and query sequences is the most complex step. Decisions had to be made regarding the handling of insertions, deletions, mutations in the downstream pattern, overlaps between potential sites and the handling of very short sequences. The first eight cases checks the behavior when the position of interest is either at the start or end of

the sequence and what happens when the position is demolished by an insertion or deletion. In one of these cases, the hypermutR package and the LANL implementation yields different results. This is the case where a control position was deleted in the query sequence. We chose to maintain this mismatch because it is consistent with the behavior when a hypermutation position gets deleted from the query sequence. Furthermore, this is an extremely rare edge case requiring a frameshift deletion in the query sequence.

Should the ancestral or query sequences be scanned for the pattern? Cases 9 and 10 investigate this question. The pattern in the control sequence is used to classify the position in to a hypermutation or a control position. This implies the assumption that the two downstream positions in the ancestral sequence mutates before the position of interest. Cases 17 and 18 duplicates this investigation, but the description is written from the context of a hypermutation position instead of a control position as with cases 9 and 10.

Gaps in the pattern does not affect the classification of the position irrespective of the number and the location of the gaps as illustrated in cases 11 through 16.

Fisher's exact test can test the two-sided hypothesis that the ratio of mutated to non-mutated control positions is unequal to the ratio of mutated to non-mutated hypermutation positions, or it can test both versions of the one-sided hypothesis (that the ratio of mutated to non-mutated control positions is greater than (or less than) the ratio of mutated to non-mutated hypermutation positions). Cases 19 and 20 confirms that it is a one-sided test checking that the ratio of mutated to non-mutated control positions is equal to or less than the ratio of mutated to non-mutated hypermutation positions.

As shown by cases 21 and 22, positions are allowed to overlap. Letters that form part of the downstream pattern for one positions can themselves be a position of interest and can also be part of the downstream pattern of another position.

If the sequences are shorter than the length of the pattern of a single position, then it is treated as if there are no positions in the sequence (Case 23). Very short sequences (lengths 3 and 4) are treated as normal sequences which can have up to 1 (or 2 in the case of a sequence of length 4) positions of interest (Cases 24 to 27).

Finally, cases 28 to 31 show that gaps at the starts of ends of sequences have no effect on the algorithm aside from the fact that they exclude the first and last positions from being considered as positions of interest.

Table 32: Edge cases evaluated and compared with the Hypermut 2.0 evaluation on the LANL website. *The result column reports the number of hypermutation detected, the number of potential hypermutation positions in the sequence, the number of control mutations detected and the number of potential control positions. The comment column indicates whether hypermutR and LANL's implementation of the hypermut 2.0 algorithm obtained the same result.*

Case	Ancestral sequence	Query Sequence	Result	p-value	Comment
1. A control position at the first position.	GCACTCAAT	CACTCAAT	0, 0, 1, 1	1	match
2. A control position at the last position.	CCACTCGCT	CCACTCACT	0, 0, 1, 1	1	match
3. A control position was deleted in the ancestral sequence.	ACT-CTACTACT	ACTACTACTACT	0, 0, 0, 0	1	match
4. A control position was deleted in the query sequence.	ACTGCTACTACT	ACT-CTACTACT	0, 0, 0, 1	1	LANL Result: 0, 0, 0, 0
5. A hypermutation position at the first position.	GAACTCAAT	AACTCAAT	1, 1, 0, 0	1	match
6. A hypermutation position at the last position.	CCACTCGAT	CCACTCAAT	1, 1, 0, 0	1	match
7. A hypermutation position was deleted in the ancestral sequence.	ACT-AACTACT	ACTAAACTACT	0, 0, 0, 0	1	match
8. A hypermutation position was deleted in the query sequence.	ACTGAACTACT	ACT-AACTACT	0, 1, 0, 0	1	match
9. Control pattern only in the ancestral sequence.	ACTGCTACT	ACTAAACT	1, 1, 0, 0	1	match
10. Control pattern only in the query sequence.	ACTGATACT	ACTACCACT	0, 0, 1, 1	1	match
11. Gaps in a control pattern in both sequences.	ACTGC-ACT	ACTAC-ACT	0, 0, 1, 1	1	match
12. Gaps in a control pattern in the ancestral sequence.	ACTGC-ACT	ACTACTACT	0, 0, 1, 1	1	match
13. Gaps in a control pattern in the query sequence.	ACTGCTACT	ACTAC-ACT	0, 0, 1, 1	1	match
14. Gaps in a hypermutation pattern in both sequences.	ACTGA-ACT	ACTAA-ACT	1, 1, 0, 0	1	match
15. Gaps in a hypermutation pattern in the ancestral sequence.	CCAGA-TACT	CCAAAATACT	1, 1, 0, 0	1	match
16. Gaps in a hypermutation pattern in the query sequence.	ACTGAACT	ACTAA-ACT	1, 1, 0, 0	1	match
17. Hypermutation pattern only in the ancestral sequence.	ACTGATACT	ACTACTACT	0, 0, 1, 1	1	match
18. Hypermutation pattern only in the query sequence.	ACTGCTACT	ACTAAACT	1, 1, 0, 0	1	match
19. More control mutations than hypermutations.	GAGAGAGAGAGAGC GCGCGCGCGCGC	GAGAGAGAGAGAAC ACACACACACAC	0, 6, 6, 6	1	match
20. More hypermutation mutations than control mutations.	GAGAGAGAGAGAGC GCGCGCGCGCGC	AAAAAAAAAAAAAGC GCGCGCGCGCGC	6, 6, 0, 6	0.0011	match
21. Overlapping control positions in the ancestral sequence.	ACTGGCACT	ACTAACACT	0, 0, 2, 2	1	match
22. Overlapping hypermutation positions in the ancestral sequence.	ACTGGAECT	ACTAAAECT	2, 2, 0, 0	1	match

23. The alignment is of length 2.	GA	AA	0, 0, 0, 0	1	match
24. The alignment is of length 3 with hypermutation.	GAA	AAA	1, 1, 0, 0	1	match
25. The alignment is of length 3 without hypermutation.	CAA	AAA	0, 0, 0, 0	1	match
26. The alignment is of length 4 with hypermutation.	CGAA	CAAA	1, 1, 0, 0	1	match
27. The alignment is of length 4 without hypermutation.	ACAA	AGAA	0, 0, 0, 0	1	match
28. The ancestral sequence ends with gaps.	ACTGCTGAAA--	ACTACTAAAAC	1, 1, 1, 1	1	match
29. The ancestral sequence starts with gaps.	--TGCTGAAAC	ACTACTAAAAC	1, 1, 1, 1	1	match
30. The query sequence ends with gaps.	ACTGCTGAAAC	ACTACTAAAA--	1, 1, 1, 1	1	match
31. The query sequence starts with gaps.	ACTGCTGAAAC	--TACTAAAAC	1, 1, 1, 1	1	match



In all except a single case the results produced by hypermutR exactly matches with the results produced by the implementation of the Hypermut 2.0 algorithm available from LANL. In the case where a control position was deleted in the query sequence, i.e. GCT became -CT, LANL does not count it as a control position while our implementation does count it as a control position.

5.2.5 Future Work and Conclusions

The hypermutR package is a high quality implementation of the Hypermut 2.0 algorithm that can be used offline. It has a comprehensive suite of unit tests and detailed documentation. Many edge cases were evaluated against the version that is available from the LANL website and all except one were found to match. In this case, if a control position (GY or GRC) is deleted in the query sequence so that it becomes -Y or -RC, then hypermutR will tally this as an unmutated control position, but LANL will not report it as a control position (mutated or unmutated). This behavior was left in hypermutR, since this is consistent with the behavior if the position is a hypermutation position. In both the LANL and hypermutR implementations, when GRD becomes -RD, it is considered as an unmutated hypermutation position. When performing the test to determine if the sequence is hypermutated, the percentage of control positions that mutated is compared to the percentage hypermutation positions that mutated. By keeping this discrepancy with LANL's handling of this case in hypermutR, we make the processes that calculate these two ratios more consistently with each other.

Currently, hypermutR lacks the ability to specify custom downstream patterns for classifying a position into either a hypermutation or control position. The data formats implemented in the popular Biostrings package (Pages et al., 2017) are not supported. A more sophisticated consensus sequence generation approach should be implemented that can handle IUPAC sequences and that has some more nuanced options for dealing with positions with many A's and G's.

5.3 Entropy Calculation

Diversity of the sequences in the alignment are summarized by computing the Shannon entropy at each position in the alignment and then reporting summary statistics. This section first explains what Shannon entropy is with the aid of several examples. The implementation of the entropy calculation is carefully described and lastly, a number of tests designed to check that the computation functions as expected are presented.

5.3.1 Definition and Explanation of Shannon Entropy

The Shannon entropy measures the amount of information present in a set of observations from a random variable. It is defined so that the maximum value is achieved when the observations are uniformly distributed across the entire domain and minimized (at value zero) if all observations are of the same value. The explicit formula is given by

$$H(X) = - \sum_{i=1}^n P(x_i) \cdot \log_b(P(x_i)).$$

In this equation, $P(x_i)$ is the probability of observing the value x_i which is estimated by the number of times that observation occurs divided by the total number of observations of all values, and b is the base of the logarithm used. Entropy also has the property that it can be summed across independent positions and still be interpretable.

When b is chosen as 2, the interpretation of H is that it counts the smallest number of yes/no questions that you have to ask before you can accurately state the value of the random variable. Consider a position where A, C, G and T are equally likely. Figure 66 shows an optimal scheme in the form of a decision tree for finding out the nucleotide using only yes/no questions. In all cases, two questions are needed, hence the Shannon entropy is 2. More algorithmically, in 25% of the cases, there will be an A and using the tree in Figure 66, two questions are needed to deduce that the base is an A. In another 25% of the cases, there will be a C and using the tree in Figure 66, two questions are needed to deduce that the base is a C. The base G will occur in another 25% of the cases, when again, 2 questions will be needed according to the tree in Figure 66. Similarly, the last 25% of cases will be a T, also requiring 2 questions. Putting this together, the Shannon entropy is computed as $2 = 2 \cdot 0.25 + 2 \cdot 0.25 + 2 \cdot 0.25 + 2 \cdot 0.25$. This is the same answer as obtained by plugging the values into the equation of $H(X)$. In this case, $P(x_i) = 0.25$ for all values of x_i and $\log_2(P(x_i)) = -2$ for all values of x_i .

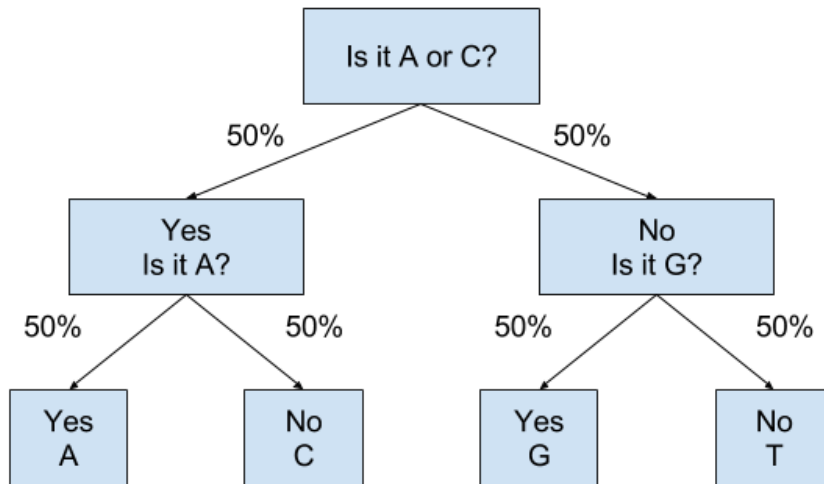


Figure 66: An optimal decision tree to accurately state the base for a single position under a uniform distribution. In all cases, two questions are needed so that the Shannon entropy (when defined using a logarithm of base 2) for this case is $2 = 2 \cdot 0.25 + 2 \cdot 0.25 + 2 \cdot 0.25 + 2 \cdot 0.25$.

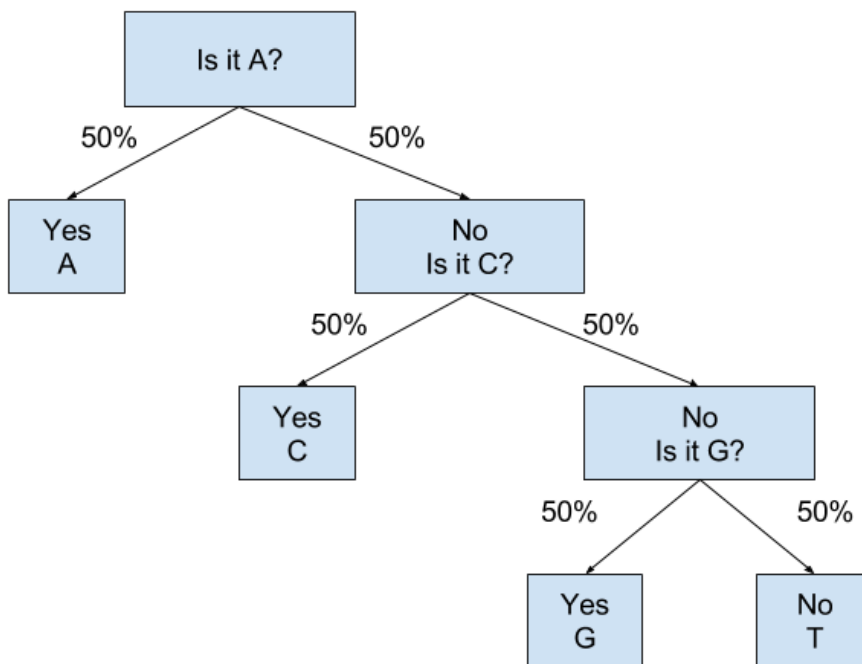


Figure 67: An optimal decision tree to accurately state the base for a single position when the bases A, C, G and T are distributed with frequencies (0.5, 0.25, 0.125, 0.125). In half the cases (A), one question is needed, in a quarter of the cases (C), two questions are needed, in an eighth of the cases (G), three questions are needed and the last eighth of the cases (T) also requires three questions so that the Shannon entropy (when defined using a logarithm of base 2) for this case is $1 \cdot 0.5 + 2 \cdot 0.25 + 3 \cdot 0.125 + 3 \cdot 0.125 = 1.75$.

A more complex example is presented in Figure 67 where the distribution is no longer uniform. By designing a more complex decision tree, the average number of yes/no questions needed to ascertain the base can be reduced. In 50% of the cases, there will be an A and using the tree in Figure 67, only

one question is needed to deduce that the base is an A. In 25% of the cases, there will be a C and using the tree in Figure 67, two questions are needed to deduce that the base is a C. The base G will occur in 12.5% of the cases and 3 questions will be needed according to the tree in Figure 67. Similarly, the last 12.5% of cases will be a T, also requiring 3 questions. Putting this together, the Shannon entropy is computed as $1 \cdot 0.5 + 2 \cdot 0.25 + 3 \cdot 0.125 + 3 \cdot 0.125 = 1.75$. This smaller number reflects the fact that there is less information in that case since on average you need to ask fewer yes/no questions to obtain the correct answer.

Shannon entropy has a useful property in that entropy of independent variables can be added together to obtain the entropy of their joint distribution. This is illustrated in Figure 68 where two positions are independently each uniformly distributed. Parallel to the case presented in Figure 68, all cases are resolved using the same number of questions, in this case 4, so that the Shannon entropy is 4, exactly double that of the case presented in Figure 68.

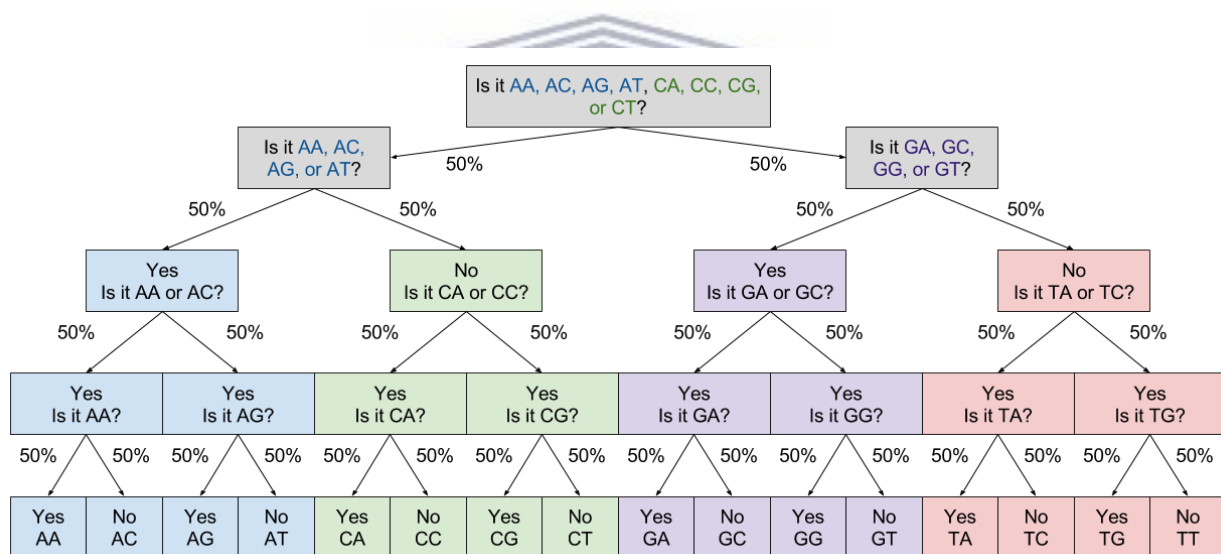


Figure 68: An optimal decision tree to accurately state the base for two positions when bases at the two positions are independently uniformly distributed. In all cases, four questions are needed, so that the Shannon entropy (when defined using a logarithm of base 2) for this case is 4, which is the sum of the entropy of two positions each of which is uniformly distributed.

5.3.2 Implementation

The pipeline includes a step that computes the Shannon entropy as a convenient measure of the diversity in the alignment. The average entropy (across the positions) and the standard deviation of the entropy (across the positions) is reported. The entropy calculation is performed by a script called `computeEntropyFromAlignedFasta.R` and it is controlled by setting the environment variables `computeEntropyFromAlignedFasta_inputFilename` and `computeEntropyFromAlignedFasta_outputDir`. The

`computEntropyFromAlignedFasta.R` scripts internally sets the values of `inputFilename` and `outputDir` from the aforementioned environment variables. `inputFilename` is parsed to obtain the path to the input file, the name of the file with its extension but with the path excluded, the name of the file without its extension and the extension only. The `outputDir` variable is checked to ensure that it specifies a valid dir. If it is not specified, then the path deduced from the `inputFilename` is used. The name for the output file is constructed by appending “.entropy.txt” to the end of the `inputFilename`.

The fasta file is read in and a consensus matrix is constructed using the `consensus` function from the `seqinr` package by specifying the `method` argument as “profile”. The consensus matrix contains a count of the number of times each character occurs at each position. IUPAC characters are handled by adding fractional amounts of the letters they represent to the consensus matrix. For example, if at position 10, a sequence had an N (representing either an A, C, G or T), then for position 10, 0.25 is added to the count for the number of A’s, 0.25 is added to the count for the number of C’s, 0.25 is added to the count for the number of G’s and 0.25 is added to the count for the number of T’s. Gaps are ignored from entropy calculations.

The `entropy.empirical` function from the package `entropy` with the logarithm base 2 as unit is applied to each position in the consensus matrix to compute the Shannon entropy for each position. The summary statistics listed in Table 33 is computed and saved to the output file. The last step in the script is to print the name (including the path) to the output file.

*Table 33: Summary statistics calculated on the entropies. An entropy is computed at each position in the alignment and these statistics summarize those per position entropies. *The R script writes the results to a file, and changes to the variable names to conform to R variable naming restrictions. Variable names that start with a number gets an ‘X’ prepended to then and spaces are replaced by underscores.*

Explanation	Name in R script*	Name in pipeline script
The number of sequences in the alignment.	N	entropy_seqs
The number of positions in the alignment.	K	entropy_sites
Across all positions, then minimum entropy.	Min.	entropy_min
The 1 st quartile (25 th percentile) of the per position entropies.	1st Qu.	entropy_q1
The median of the entropies across all positions.	Median	entropy_median
The average of the entropies across all positions.	Mean	mean_entropy
The 3 rd quartile (75 th percentile) of the per position entropies.	3rd Qu.	entropy_q3
The maximum entropy found at any of the positions.	Max.	entropy_max
The standard deviations of the entropies calculated at each position.	SD	sd_entropy

The pipeline calls the `computeEntropyFromAlignedFasta.R` script using the backtick notation. The output (to `STDOUT`) from the script is parsed for the name of the output file. The output file is read and parsed, initializing the variables listed in the last column of Table 33. Of the metrics listed in Table 33, average of the entropies and the standard deviation of the entropies is added to the `identify_founders.tab` with the names `mean.entropy` and `sd.entropy`.

5.3.3 Tests and Examples

A number of alignments were generated to test the behavior of the entropy calculation script as shown in Table 34. Five edge cases were considered. Tests 1 to 3 check the behavior of the script when the sequences are just a single repeated base. When all the sequences are only As, all entropies are zero since there is no variation. In the case where each sequence is a different base repeated (as in test 2), the entropy is maximized as illustrated in the example presented in Figure 66. This maximization occurs due to the fact that the positions are assumed to be independent, while they often are not, highlighting a shortcoming of measuring diversity in this way. Tests 1 and 2 minimized and maximized entropy using sequences that are just a single repeated base. Test 3 is based on a dataset with an intermediate amount of entropy while also still consisting of sequences that are just a single repeated base. Tests 10 and 11 check that adding either a sequence composed only of gaps, or a position composed only of gaps, has no effect on the results. Gaps are ignored when entropy is computed by design.

Four cases were generated by randomly sampling bases from a uniform distribution in which each base has a 25% chance to occur at any position in any sequence. The four cases differ in the number of sequences and positions they include. As is expected, when using a small number of sequences (only four sequences as in test cases 4 and 6), large variances are observed in the per position entropies. The chance of sampling the same letter four times is 0.0039 or approximately one in 256 which did not occur in the case with only 10 positions. The minimum entropy when looking at only 10 positions was 0.8113, achieved when one letter occurs 3 times, another letter occurs 1 time and the other two letters do not occur. However, when 1000 positions were considered, an entropy of zero was observed among the 1000 four-letter draws.

As the number of sequences increases, it is expected that all the summary measures of entropy (all the quantiles as well as the average), will converge to the entropy of the population distribution from which sampling occurred. Since the uniform distribution under which each base has a 25% chance of occurring implies an entropy of 2 (as illustrated in the examples presented in Figure 66 and Figure 68), it is expected that with enough uniformly random sequences, these statistics will converge to 2. As expected, in both cases involving 400 sequences the minimum entropy exceeds 1.95 and the averages

exceeds 1.99. Together cases 4 through 7 show that the calculations performed by the pipeline behaves as expected when the amount of sequences are increased or decreased, providing another basic check of these properties.

Cases 8 and 9 introduce gaps randomly across the sequence over all positions. Since gaps are ignored, the expected effect is only that the variance might increase by a small amount due to the reduced number of observations. Indeed, the statistics are comparable between the cases with gaps and those similar to them but without gaps (Both cases 4 and 8 concern 4 short sequences and both cases 7 and 9 concerns 400 long sequences), but with slightly elevated standard deviations (0.46 vs 0.56 and 0.0051 vs 0.0061).

The last two cases (numbers 12 and 13) are the datasets simulated based on trees derived from real-world data. The low diversity sample is extremely homogeneous, with the most common sequence accounting for 97.6% (851 out of 872) of the dataset. In total the low diversity sample contains only 20 unique sequences. The high diversity timepoint sample is also highly homogeneous when compared with the contrived test cases. In the high diversity dataset, there are a total of 183 unique sequences (out of 691) with the most frequent sequence accounting for 12.2% (84 of 691) of the sample. The entropy statistics reflect these relatively low levels of diversity well, with average entropies of 0.0006 and 0.1853 respectively.

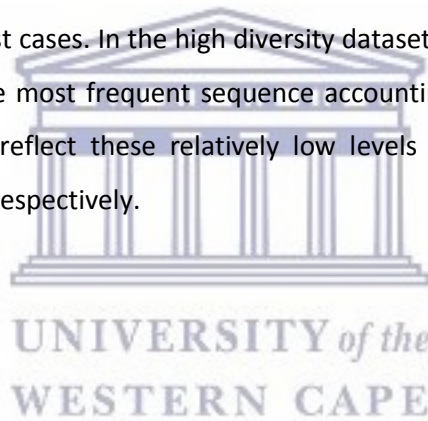


Table 34: Results from running the entropy calculation script on a set of simulated datasets.

Test Description	N	K	Min.	1 st Qu.	Median	Mean	3 rd Qu.	Max.	SD
1. Three sequences of length 37 composed entirely of A's	3	37	0	0	0	0	0	0	0
2. Four sequences of length 45, one only A's, one only c's, one only G's and the last only T's.	4	45	2	2	2	2	2	2	0
3. Eight sequences of length 13 in the proportions used in Figure 67 where each sequence is just a single base repeated 13 times.	8	13	1.75	1.75	1.75	1.75	1.75	1.75	0
4. Four sequences of length 10 each where each base is randomly sampled from a uniform ($\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}$) distribution.	4	10	0.8113	1	1	1.2623	1.5	2	0.4575
5. Four hundred sequences of length 10 each where each base is randomly sampled from a uniform ($\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}$) distribution.	400	10	1.9749	1.9914	1.9938	1.9926	1.9967	1.9997	0.007
6. Four of length 1000 each where each base is randomly sampled from a uniform ($\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}$) distribution.	4	1000	0	1	1.5	1.3129	1.5	2	0.4
7. Four hundred sequences of length 400 each where each base is randomly sampled from a uniform ($\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}$) distribution.	400	400	1.9681	1.9924	1.996	1.9944	1.9979	2	0.0051
8. Four sequences of length 10 each where each base is randomly sampled from a uniform (0.2, 0.2, 0.2, 0.2, 0.2) distribution where the fifth option is for a gap.	4	10	0	0.9183	1	1.1318	1.5637	2	0.5587
9. Four hundred sequences of length 400 each where each base is randomly sampled from a uniform (0.2, 0.2, 0.2, 0.2, 0.2) distribution where the fifth option is for a gap.	400	400	1.9603	1.9902	1.9947	1.9929	1.9973	2	0.0061
10. Four sequences of length 10 each where each base is randomly sampled from a uniform ($\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}$) distribution. A fifth sequence consisting only of gaps were also added.	5	10	0.8113	1	1	1.2623	1.5	2	0.4575

11. Four sequences of length 10 each where each base is randomly sampled from a uniform ($\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}$) distribution. An eleventh position in which all sequences had only gaps were also added.	4	11	0.8113	1	1	1.2623	1.5	2	0.4575
12. Simulated low diversity sample.	872	471	0	0	0	0.0006	0	0.0331	0.0029
13. Simulated high diversity sample.	691	471	0	0.0157	0.0619	0.1853	0.2343	1.2008	0.2701



5.3.4 Future Work and Conclusions

The `computeEntropyFromAlignedFasta.R` script is a basic wrapper for the `entropy.empirical` function from the `entropy` package, with handling for IUPAC characters as described in section 5.3.2, gaps, and performing some basic result processing. This script provides the entropy statistics to the pipeline, where they are employed as measures of sequence diversity in the alignments. We performed a number of tests and they demonstrate that the entropy computation step works as expected. A key shortcoming of using statistics computed on entropies calculated at each position is the underlying assumption of independence between the positions. In the case in which a small number of sequences that diverge significantly from each other each occur many times in the sample, entropy will be very high. In such cases, entropy immediately following dual infection can possibly be higher than it will be for later samples when each of the two strains have developed a diverse quasispecies. Including different measures of diversity, as described in (Gregori et al., 2016), may ameliorate this problem.



5.4 PhyML

PhyML is phylogeny software designed to compute tree topology based on the maximum-likelihood principle (Guindon et al., 2010). PhyML is incorporated into the pipeline to produce trees that can be manually reviewed by experts as part of an audit of the data processing. Additionally, the average of the pairwise distances computed from the tree informs the classification of the sample as either a single founder or multi-founder sample.

5.4.1 Overview of PhyML

The main goal of PhyML is to estimate maximum likelihood phylogenies from aligned sequences. Briefly, a set of rules relating the probabilities of the different mutations to each other is specified. An algorithm is then used to search through different parameterizations of the model (tree shapes, branch lengths and mutation rates satisfying the specified rules) for the parameter values under which the data is most likely. The most flexible set of rules for the probabilities is to assume that all probabilities are unconstrained of each other and is called the General Time-Reversible (GTR) model.

PhyML produces two output files: A file with a number of relevant statistics (file name ends with `_phyml_stats.txt`) and a file with the most likely tree (file name ends in `_phyml_tree.txt`) in Newick format. While it is running, PhyML prints the settings it was called with, the value of the log likelihood function for some trees and the total time taken to STDOUT.

5.4.2 Implementation Details

The pipeline includes a wrapper, called `runPhyML.pl`, that formats the data for PhyML, calls PhyML and parses the results. [Note that a patched version of PhyML is required. Appendix 8.6 contains the instructions for obtaining this patched version of PhyML.](#) The main script, `identify_founders.pl`, calls the `runPhyML.pl` script with the back tick notation and supplies two command line arguments which specify the input file and the location for the intermediate files and results. `runPhyML.pl` first changes the line ends of the input file to Unix style line endings (`\n` instead of `\r\n`) and then converts the file to the sequential Phylip format (hereafter referred to as the Phylip format). Phylip format specifies that the first line of the file must contain two integers separated by a space, the first number is the number of sequences in the file and the second is the length of the alignment (all sequences must be of equal length). The rest of the Phylip file contains the sequences, one per line. Each line contains the header and the sequence itself separated by a tab. A maximum of 4000 sequences can be passed to PhyML. If there are more than 4000 sequences, then only the first 4000 will be processed.

The call to PhyML is executed using the `system` function and several options, described in Table 35, are passed to PhyML as command line arguments. The output produced on STDOUT and STDERR by this call are captured into files whose names are formed by stripping the extension from the input

fasta file and appending `.phylip_phyml.out` and `.phylip_phyml.err` respectively. If any output was generated on STDERR, execution is halted and that output is printed to screen. If no errors were encountered, then the distance matrix is parsed out of the output that was printed to STDOUT. All the pairwise distances are stored in an array called `@diversity`. Summary statistics are computed on this array and stored in a file whose name is constructed by stripping the extension from the input fasta file and appending `'phylip_phyml_pwdiversity.txt'` to it. The summary statistics recorded in the pairwise diversity file includes the number of sequences, average, standard error, minimum, median, maximum and the 1st and 3rd quartiles.

Table 35: The parameters specified in the call to PhyML by the `runPhyML.pl` script.

Option Flag	Value	Description
<code>-i</code>	Name and path of the input file	
<code>-d</code>	nt	The sequence data type, nt indicates nucleotides.
<code>-q</code>	NA	When the <code>-q</code> flag is specified, then PhyML will treat the input data as if it is in sequential Phylip format.
<code>-b</code>	0	Do not use bootstrapping.
<code>-m</code>	GTR	Use the general time reversible substitution model.
<code>-v</code>	e	The proportion of invariable sites, where an invariable site is a site that does not evolve. The value 'e' means that this proportional should be estimated from the data.
<code>-c</code>	4	The number of categories of substitution rates. Rates of evolution vary from site to site. The value 4 indicates that the model should divide the rates of evolution into 4 categories and estimate rates for each of these four categories.
<code>-o</code>	tlr	Instruct PhyML to compute the optimal tree topology, branch lengths and rate parameters. PhyML can be used to compute the likelihood of a given tree for a given dataset, so setting this parameter to 'n' will prevent PhyML from changing the topology, branch lengths or rate parameters and it will instead just compute the likelihood of the tree.
<code>-a</code>	e	The parameter (alpha) of the gamma distribution that models the substitution rates should be estimated from the data.
<code>-f</code>	m	The frequencies of the bases under equilibrium should be estimated using maximum likelihood.
<code>-t</code>	e	The ratio of transitions to transversions should be estimated from the data.
<code>-s</code>	NNI	Which process should be used to optimize the tree topology? NNI specifies a hill-climbing algorithm that simultaneously adjusts the topology and branch lengths to maximize the likelihood.
<code>--no_memory_check</code>	Suppresses an interactive warning from PhyML if the analysis will use a large amount of memory	

5.4.3 Test Procedure

Testing the integration of PhyML into the pipeline focused on two aspects: The presence of all sequences in the trees produced and relative comparisons on the pairwise distance metrics. Since the

trees will be reviewed manually and setting up algorithmic tests for tree topology is a complex task, we did not include any tests for tree shape. Ensuring that all sequences are present in the tree is required since manually checking for the presence of 100s of sequences is not practical. Computing the true solution for the pairwise distances is also a complex task since it is dependent on the tree topology, hence we primarily employed tests that compare these metrics on two datasets when we know that one dataset is more diverse than the other. Some manual inspection of the absolute values of PhyML's pairwise distance estimates were performed and discussed where the unexpected was observed.

A further complication in testing phylogenetic calculations is that the simplest datasets (from a simulation point of view) present major obstacles to the algorithms that estimate the tree shape and parameter values. For example, the most basic dataset is one in which all sequences at all positions are exactly the same nucleotide. In this case there is no information for estimating the parameter values associated with the other three nucleotides and a very large set of shapes are equally likely. This presents challenges to any algorithm that needs to converge to some optimal solution.

The testing is divided into three sections. Unless stated otherwise, datasets were simulated to mimic a bifurcating tree in which each offspring sequence has a chance to have a number of mutations away from its parent. Data simulation is described in detail later in the next section. The first set of tests evaluates edge cases exploring extreme homogeneity (1.1), extreme heterogeneity (1.2), order of the sequences (1.3), presence of gaps (1.4) and presence of ambiguity characters (1.5). The second set of tests focuses on relative pairwise distances. In the first test in this section, the number of mutations differentiating an offspring sequence from its parent were systematically increased (2.1). Relative pairwise distances were also compared on datasets simulated to mimic single, dual and 'triple' infection (2.2). A dataset that does not have a tree shaped phylogeny was simulated by taking an initial sequence and deriving sequences by randomly mutating random bases in the sequence (2.3). The last tests in the relative pairwise distances based section examined MiSEQ-like (2.4) and SGQ-like (2.5) datasets. The final group of tests use datasets designed to mimic either a real low-diversity sample (3.1) or a real high diversity sample (3.2). More details about the tests that were performed are listed in Table 40.

Each test involves calling `runPhyML.pl` to process three closely related datasets, a target dataset, a dataset designed to mimic the target dataset but with less variability and a dataset designed to mimic the target dataset but with more variability. The target dataset is also referred to as the moderately diverse dataset. The type of comparison to perform between the target and low and high diversity datasets must also be specified. For most cases, the low (high) variability dataset will be expected to

have strictly smaller (larger) average pairwise distances than the target dataset. However in some cases, such as testing the ordering of the sequences, the goal is to test for equality of average pairwise distances, since having the result of the phylogenetic analysis dependent upon the input order of the sequences is sub-optimal.

Six checks are performed when a test is run. First, the `runPhyML.pl` script should execute successfully on all three datasets. All the sequences from the original dataset should be present in the input file prepared for PhyML and all the sequences present in the input file prepared for PhyML should be in the original dataset. Since the output from PhyML does not include the sequences themselves, the only way simplistic way to check that PhyML was run on the correct sequences is to ensure that the number of leaf nodes in the tree match the number of sequences in the original dataset. Lastly, the average pairwise distances from the low (high) diversity dataset should be smaller (larger) (or in special cases, equal) to the average pairwise distances. Additionally, the summary metrics of the pairwise distances were also recorded and reported. The results of all tests are presented in Table 41 and are further discussed in section 5.4.5.

5.4.4 Data simulation

Four distinct types of datasets were simulated for the testing of PhyML and Poisson Fitter. [Poisson Fitter is an algorithm for computing how much time has elapsed between initial infection of a patient with HIV and when a sample was taken based on sequence data for the viral quasispecies. Poisson Fitter is discussed in detail in section 5.5.](#) The first approach generated sequences by simulating a tree and producing a dataset from the leaf nodes. This approach should generate datasets that are easy for phylogenetic algorithms to model accurately. The second approach generates data that is not based on a tree structure and is equivalent to selecting an ancestral sequence and then generating sequences by randomly introducing mutations into the ancestral sequence. This so-called “star-like phylogeny model” has been proposed as sensible for acute HIV infection and is the model assumed by the Poisson Fitter algorithm (Elena E Giorgi et al., 2010). Unless otherwise stated, all simulations were initiated with a sequence that is just ACGT repeated until a sequence of adequate length was obtained. Next the preparation of a custom dataset, referred to as the balanced dataset, designed to allow modification while preserving the nucleotide and mutation ratios. The final portion of this section describes the simulation of datasets based on real world samples.

5.4.4.1 Simulation of tree-like datasets

A simple recursive algorithm was used to generate the tree-like datasets. Given a single sequence, two sequences are derived from it by introducing between zero and x mutations into the initial sequence. The location of the mutations is determined uniformly without replacement, and the replaced

nucleotide is sampled uniformly from the other three nucleotides when a mutation occurs. No gap characters are introduced by this process and only non-gap characters are eligible for mutation. The parameter x can be changed to control the amount of diversity in the dataset. Each of the two derived sequences are each then recursively treated as the single given sequence and two sequences are derived from each of them by introducing between zero and x mutations per new sequence. Thus after two steps, four sequences were generated. This process is repeated n times until the desired number of leaf sequences are obtained. This simulation approach is referred to in the rest of this section as the recursive algorithm.

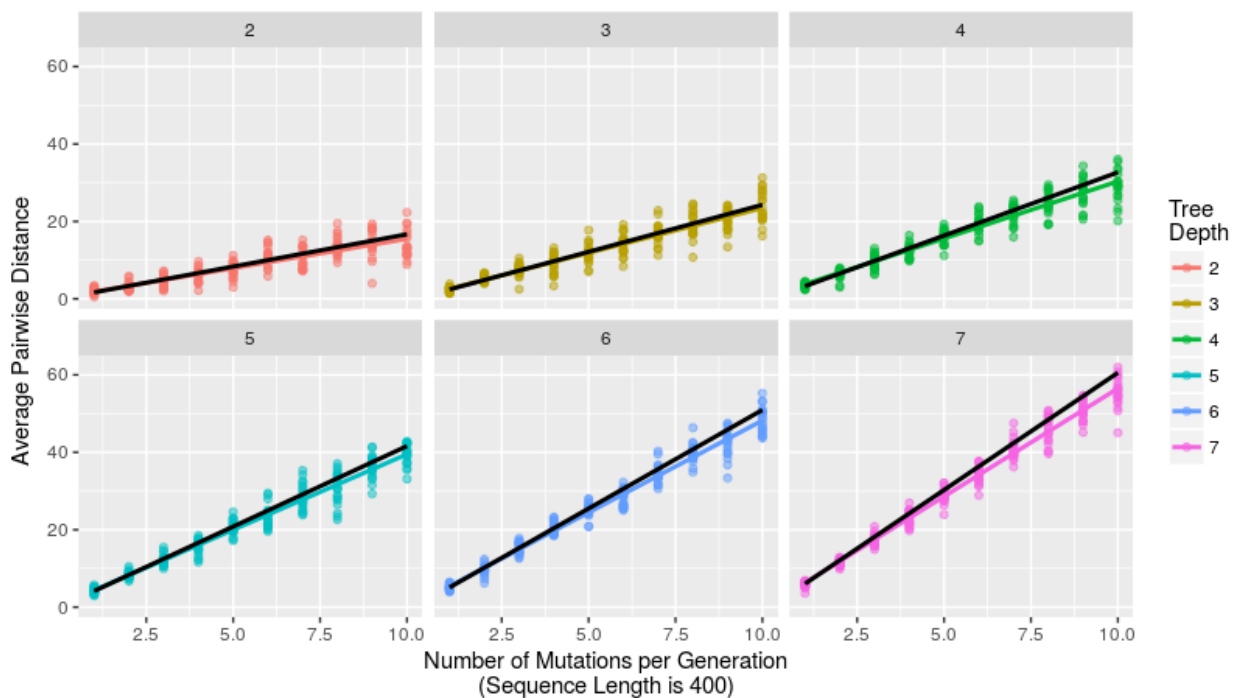


Figure 69: The average pairwise Hamming distance between two sequences in a dataset simulated using the tree-based algorithm based on the depth of the tree and the maximum number of mutations per generation. The black lines shows the predicted pairwise distance under the assumption that any position can only mutate once in the simulation process. Sequences are of length 400, so that 1 mutation will result in a 0.25% difference between two sequences. A tree depth of 2 results in a dataset with $2^2 = 4$ sequences while a tree depth of 7 yields a dataset with $2^7 = 128$ sequences.

A dataset produced with this recursive simulation approach contains all the n 'th generational offspring. Each offspring sequence differs from the ancestral sequence by $\frac{nx}{2}$ mutations on average since each of the n generations will introduce $\frac{x}{2}$ mutations on average under the assumption that the probability of two mutations affecting the same nucleotide is negligible. Additionally, the relatedness between the leaf nodes implies that each sequence has one (2^0) sequence that it differs from by $1 \cdot x$ mutations on average, two (2^1) sequences that differs from it by $2 \cdot x$ mutations, four (2^2) sequences that differs from it by $3 \cdot x$ mutations and so forth. Thus, on average two sequences in the dataset differ from each other by

$$\frac{\sum_{i=0}^{n-1} [2^i (i + 1)x]}{2^n - 1}$$

Equation 1

mutations on average assuming that no position in the sequence can mutate more than once. Figure 69 shows the measured (colored lines) and predicted (black lines) pairwise distances in datasets simulated using the recursive algorithm. At higher mutation rates and for larger generation numbers, the actual average pairwise sequence distances will diverge from the distances predicted by the previously mentioned equation as mutations will start to override each other as they will increasingly occur at the same positions.

5.4.4.2 Simulation of star-like datasets

To generate datasets that do not have a tree-like structure, a single frequency matrix containing the target prevalence of each nucleotide at each position is constructed using a simulation approach. In a second step, this frequency matrix will be used to generate the sequences of the dataset. The frequency matrix has four rows each representing a nucleotide. The number of columns in the frequency matrix correspond to the length of the sequences that will be simulated. A sequence is simulated from such a frequency matrix by drawing nucleotides for each position with the probabilities given by the frequency matrix. A dataset is simulated by simulating many sequences from the same frequency matrix.

To construct a frequency matrix, a parameter, called the dominance parameter (dom), and a target sequence is required. Each position is treated completely independently from the other positions. The dominant nucleotide (the most frequently occurring nucleotide) for the position is read from the target sequence. The prevalence for the dominant nucleotide at each position is sampled uniformly from the interval between the value of the dominance parameter and one. This produces a vector of frequencies, dom_i , containing the frequencies of the most prevalent nucleotide at each position indicated by i in the subscript. Note that each sampled dom_i is larger than dom parameter. The prevalences of the remaining three nucleotides at a position, i , is determined by uniformly drawing an integer, z_i , between zero and two which specifies how many of the remaining three parameters will be non-zero. From the remaining three nucleotides (the non-dominant nucleotides at the position), z_i nucleotides are randomly drawn and their prevalences are set to zero. The remaining nucleotide(s) will then be assigned prevalences equal to $\frac{1-dom_i}{3-z_i}$, where dom_i is the prevalence of the most prevalent nucleotide for position i . Hence, at each position, a single nucleotide will occur the majority of the time, some (possibly none) nucleotides will never occur and the remaining mass will be equally distributed between the remaining nucleotides. This process is repeated for each position

in the dataset until the entire frequency matrix is populated. The prevalences for each position forms a discrete uniform distribution whose support is the four nucleotides, A, C, G and T. Simulating sequences with draws weighed by the prevalences contained in this frequency matrix yields a dataset with a star-like phylogeny. This simulation procedure will be referred to as the star-like simulation approach in the rest of this document. Unless specified otherwise, the target sequence is taken to be ACGT repeated until a sequence of the desired length is obtained.

As an example of the simulation process of the star-like approach, consider the simulation of a dataset with sequences of length two and a dominance parameter of 0.9 and target sequence TA. For the first position, four random draws are performed:

- 1) A number between 0.9 and 1 is drawn, say 0.92;
- 2) The dominant nucleotide is read from the target sequence, T;
- 3) Another number between 0 and 2 is drawn, say 1; and
- 4) Another letter (that does not match the first) is drawn, say a G.

The discrete uniform distribution for the first position will then be: zero chance of drawing a G, 0.92 chance of drawing a , 0.04 chance of drawing a G and an 0.04 chance of drawing a C. Independently from the first position, another four random draws are performed for the second position:

- 1) A number between 0.9 and 1 is drawn, say 0.99;
- 2) The dominant nucleotide is read from the target sequence, A;
- 3) Another number between 0 and 2 is drawn, say 2; and
- 4) Two letters (that does not match the first) is drawn, say G and T.

The discrete uniform distribution for the first position will then be: 0% chance of drawing a G or a T, 99% chance of drawing an A and a 1% chance of drawing a T. Using these two discrete distributions, a dataset of sequences can easily be generated.

In a dataset that was simulated using the star-like approach, each sequence will deviate from the consensus sequence by $\left(\frac{1-dom}{2}\right) \cdot 100\%$ on average and any two sequences will differ from each other by approximately

$$(1 - dom) \cdot 100\% \qquad \text{Equation 2}$$

on average. To convert this number to an average pairwise Hamming distance, multiply it by the length of the sequence. We saw above that in the recursive simulation approach, two parameters, the tree depth and the number of mutations per generation together strongly influences the eventual average

pairwise distances in the dataset (Figure 69). In the star-like simulation approach, the average pairwise distance is only determined by the dominance parameter (Figure 70).

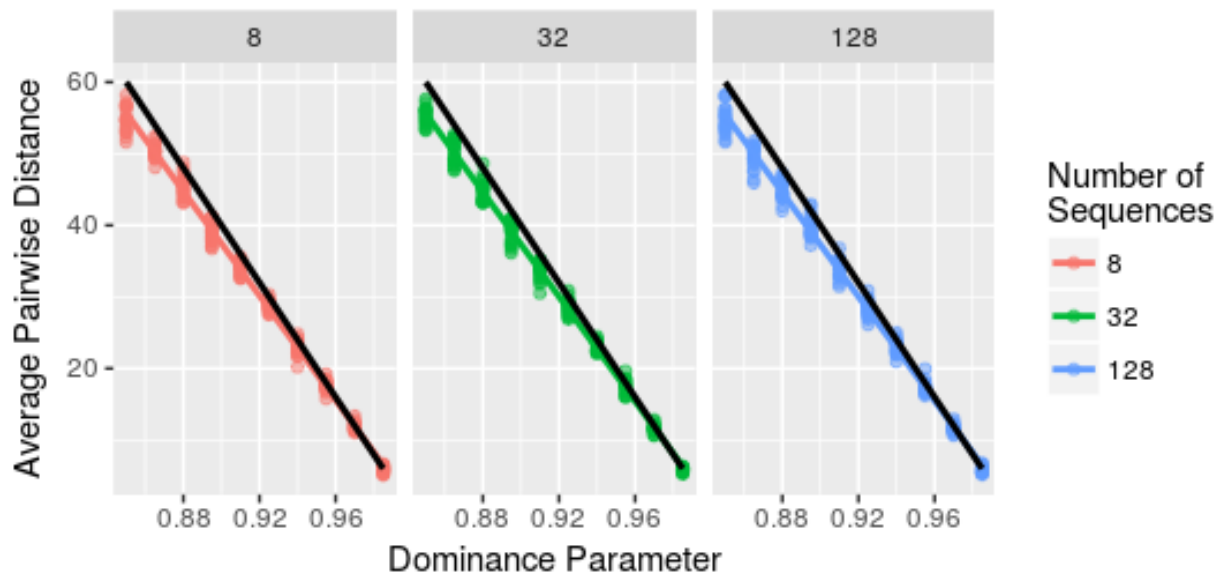


Figure 70: The average pairwise Hamming distance between two sequences in a dataset simulated using the star-like algorithm based on the number of sequences and the dominance parameter. The black lines shows the predicted pairwise distance under the assumption that any position can only mutate once in the simulation process. Sequences are of length 400. Parameters were chosen to be comparable to Figure 69 which presents the same data for the recursive algorithm. The number of sequences, 8, 32 and 128, correspond with tree depths of 3, 5 and 7 respectively. The dominance parameters were chosen to correspond to 1, 2,..., 10 mutations per generation at a tree depth of 7.

5.4.4.3 Simulation of dataset used to test effects of non-standard characters

For the tests that explore the effects of gaps and ambiguity characters, a dataset is required that contains some relationship between the data and which can be easily modified while preserving the ratios of the different nucleotides and mutations. To accomplish this, a base dataset was constructed to have 25% of each base and to have an equal amount of mutations between all nucleotides. The dataset consists of 128 sequences of length 400 formed by repeating the pattern ACGT 100 times. To add some phylogenetic relationships to the dataset, 60 positions were modified as documented in Table 36. Since each block of 4 positions contain an equal amount of each nucleotide, a side-effect of the repeating ACGT pattern, the modifications are built around blocks of 4 positions and maintains the equal amounts of each nucleotide in each block of four positions. The modifications were designed to divide the alignment into either two or three groups of the sequences that match each other. Those modifications that chunk the alignment into two groups can be further categorized by the relative sizes of the two groups:

- Each containing 50% of the alignment as at positions 9-12, 109-112, and 209-212;
- One group with 25% of the alignment and the other with 75% as at positions 1-4, 101-104, and 201-204;

- One group with 12.5% of the alignment and the other with 87.5% as at positions 5-8, 105-108, and 205-208.

The modifications that chunk the alignment into three groups always constructs groups with relative sizes of 25%, 25% and 50% of the alignment. This dataset is referred to as the balanced dataset.

Table 36: Description of positions modified in the dataset which forms the basis for the tests that explore gaps and ambiguous letters. Modifications were designed to maintain an equal distribution of the four nucleotides and to keep the number of mutations between nucleotides the same. Since the dataset contains 128 sequences, patterns of length 2, 4, or 8 can be repeated a number of times and affect all sequences without having any remainder thus preserving the nucleotide ratios.

Positions	Seq 1	Seq 2	Seq 3	Seq 4	Seq 5	Seq 6	Seq 7	Seq 8	Seq 9
1, 101, 201	A	C	C	C					Pattern Repeats
2, 102, 202	C	A	A	A					Pattern Repeats
3, 103, 203	G	T	T	T					Pattern Repeats
4, 104, 204	T	G	G	G					Pattern Repeats
5, 105, 205	A	C	C	C	C	C	C	C	Pattern Repeats
6, 106, 206	C	A	A	A	A	A	A	A	Pattern Repeats
7, 107, 207	G	T	T	T	T	T	T	T	Pattern Repeats
8, 108, 208	T	G	G	G	G	G	G	G	Pattern Repeats
9, 109, 209	A	C							Pattern Repeats
10, 110, 210	C	A							Pattern Repeats
11, 111, 211	G	T							Pattern Repeats
12, 112, 212	T	G							Pattern Repeats
13, 113, 213	A	C	C	G					Pattern Repeats
14, 114, 214	C	A	A	T					Pattern Repeats
15, 115, 215	G	T	T	C					Pattern Repeats
16, 116, 216	T	G	G	A					Pattern Repeats
17, 117, 217	C	A	C	G					Pattern Repeats
18, 118, 218	A	C	A	T					Pattern Repeats
19, 119, 219	T	G	T	C					Pattern Repeats
20, 120, 220	G	T	G	A					Pattern Repeats

5.4.4.4 Simulation of datasets based on real world samples

The HVTN 503/Phambili study followed HIV negative subjects monitoring for HIV-1 infection to evaluate an HIV-1 vaccine (Gray et al., 2011). For testing purposes, we took the PID Illumina MiSeq sequence data from two time points (HVTN503-162400146-1011, referred to as low diversity or LD dataset, and HVTN503-162450071-1056, referred to as the high diversity or HD dataset) and built phylogenetic trees with RAxML. The setting specified for RAxML together with their explanations are listed in Table 37. Using the trees produced by RAxML, a random subtype-C sequence was selected (referred to as the seed sequence) from LANL (C.ZA.08.707PKE34F2.HM623575), restricted to the same amplicon as the real dataset and mutated according to these trees.

To simulate test data, the trees were loaded into R in a `data.frame` in which each row represents an edge. The `data.frame` contain three columns, the first one listing the ancestor, the second one listing the descendant and the last one the length of the edge. The simulation is initiated by assigning the seed sequence to the descendant in the first row of the dataset. The ancestor is then constructed by randomly mutating the seed sequence until it diverged by the edge length. The newly simulated ancestor sequence is the used to generate the other sequences that are directly related to it. This process is continued until all the sequences in the entire tree (including the internal nodes) are generated. To introduce extra variability into the datasets, a `mutation_booster` variable was used. This variable was set to 0.5, 1 or 2 and the branch lengths were multiplied by this variable enabling the generation datasets with differing levels of diversity while keeping the underlying phylogeny unchanged. These simulated datasets are referred to by appending `_bx` to their source dataset where `x` is the factor by which the branch lengths were multiplied. For example, the dataset constructed by multiplying the branch lengths of the LD dataset by 2 is called the LD_b2 dataset.

To evaluate the simulated datasets, RAxML was used to draw trees from the simulated datasets which was then visually compared to the tree of the real datasets (Figure 71). While minor changes to the trees occurred, the datasets based on the LD dataset maintains a star like phylogeny and the datasets based on the HD dataset exhibits more complex behavior. The datasets used most frequently are the LD_b1 and HD_b2 datasets and they are referred to as the LD_SEQs and HD_SEQs datasets respectively.

Table 37: RAxML settings used to draw trees from which the testing datasets were simulated.

Setting	Description
-f a	Perform rapid bootstrap analysis and search for the best-scoring maximum likelihood tree in one program run.
-x 12345	Seed for the random number generator used by the rapid bootstrap analysis.
-p 12345	Seed for the random number generator used in the parsimony inferences.
-# 100	The number of bootstrap analyses to run on distinct starting trees.
-m GTRGAMMA	The model used for the nucleotide substitutions. The general time reversible model with optimization of the substitution rates and the GAMMA model of rate heterogeneity.

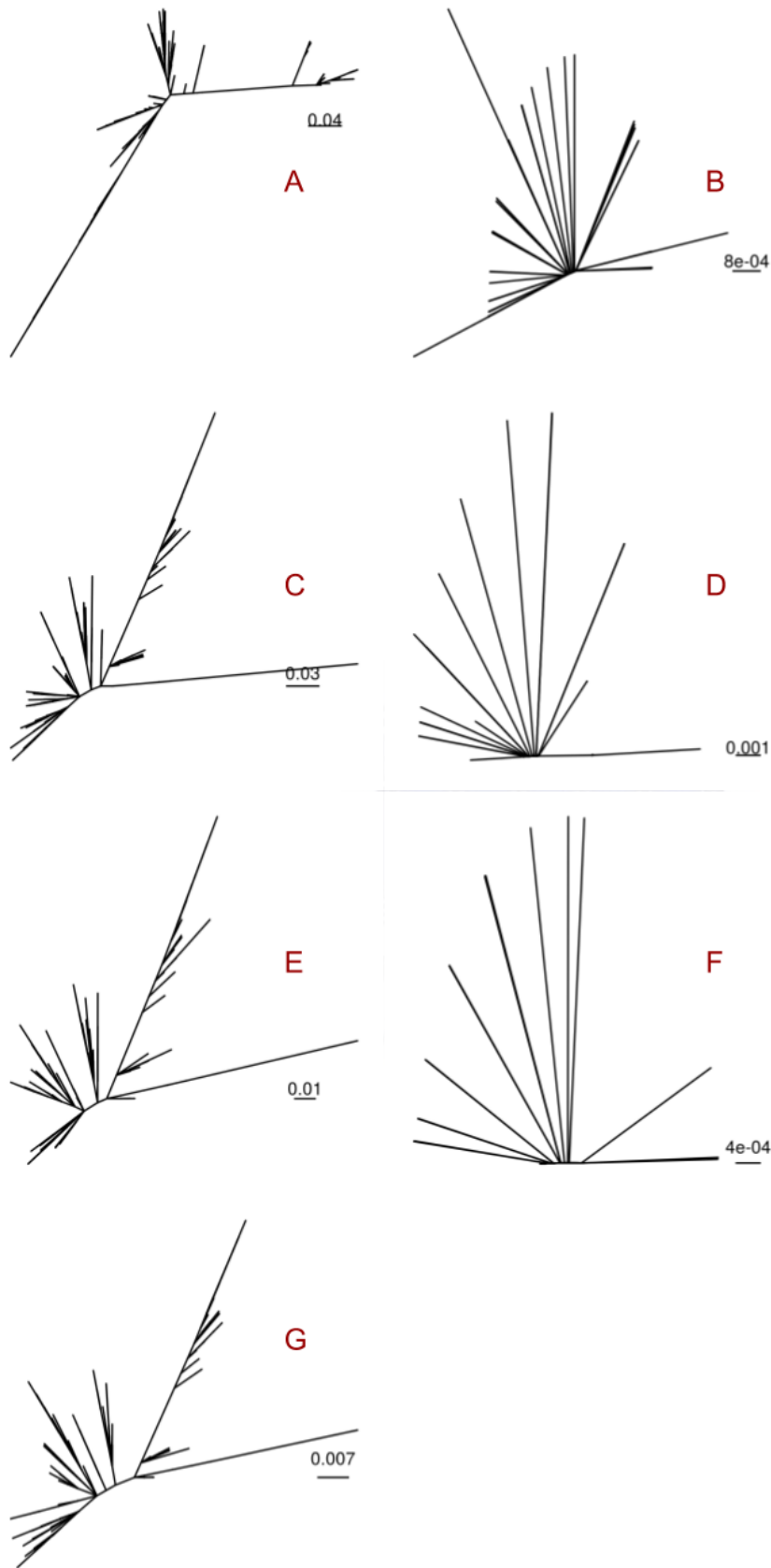


Figure 71: The two trees constructed from the datasets (A and B) and the five trees (C-G) built from the simulated datasets. The left column (A, C, E, G) are based on the HD dataset and the right column (B, D, F) is based on the LD dataset. The first row (A,B) is the trees constructed from the datasets, the second row (C,D) is constructed with the branch lengths doubled, the third row (E,F) is the trees constructed with the branch lengths unmodified and the last row (G) has the branch lengths halved. Due to the very low levels of diversity in the datasets based on the LD dataset, the branch lengths were not halved for this dataset. These datasets are referred to by appending *_bx* to their source dataset where *x* is the factor by which the branch lengths were multiplied. For example, the dataset from which tree D was drawn is called LD_b2 and HD_b0.5 for tree G.

5.4.5 Test results and discussion

In total 14 tests were performed, each comparing three datasets to each other. Summarized descriptions of the tests and datasets are listed in Table 40 and the results and brief comments about the results are presented in Table 41. In this section, each test will be described in detail and the results will be interpreted. The tests verify that PhyML ran without error on each of the datasets, that all input sequences are present in the output and that all the output sequences were also present in the input datasets. The results of these checks are summarized in the “Checks 1 to 4” column of Table 41. The second part of the test evaluates PhyML’s diversity metrics on three datasets designed to have a specific ranking of diversity. The columns “Low PD stat.,” “Target PD stat.” and “High PD stat.” of Table 41 report the average pairwise distances as computed by PhyML. The columns named “Low over Target” and “Target over High” reports the ratios of the average pairwise distances computed by PhyML which are used to decide whether or not the test was passed or failed.

In cases where unexpected results occurred, comparisons are made between the average pairwise distances computed by PhyML and metrics computed on the test datasets themselves. Two such metrics used are a predicted pairwise diversity and a computed pairwise Hamming distance. Equation 1 and Equation 2 are used to predict the Hamming distance for a simulated dataset. Since the simulation process is stochastic, the true average pairwise Hamming distance of the dataset can vary from the predicted amount as demonstrated in Figure 72 and Figure 73.

5.4.5.1 Test 1.1.1: Extreme homogeneity

Using very small datasets with only four sequences each of length 10, the ability of PhyML to process and compute pairwise distances on extremely homogenous datasets were tested. The low diversity dataset contained only As at all positions. A single mutation, changing the first A in the first sequence to a C, was introduced to construct the target (or moderately diverse) dataset. The last dataset for this test was constructed by introducing two mutations relative to the low diversity dataset. These two mutations were changing the first position in the first sequence to an A and changing the second position in the first sequence to a T. PhyML correctly reported all pairwise distances as zero for the low diversity dataset and each added mutation was accompanied by a corresponding increase in the average pairwise distance indicating that PhyML can process small extremely homogenous datasets correctly. However, the increases in the estimated pairwise distances were large, with a change to 1 of the 40 nucleotides increasing PhyML’s estimate by 0.05 per nucleotide where this same change increased the observed Hamming distances between the sequences by only 0.0125. While this

discrepancy seems large, this is an extreme case and overall the distances computed by PhyML correlates well with the Hamming distance as will be shown later in Figure 76. The datasets used in this test are summarized in the first row of Table 40 and the results are shown in the first row of Table 41.

5.4.5.2 *Test 1.2.1: Extreme heterogeneity*

Extreme diversity in the datasets was simulated by using four sequences of length 10, the first sequence containing only As, the second sequence only Cs and so forth. The implication is that for the most diverse dataset, at no position do any sequence match any other sequence. To reduce the diversity for the target and dataset, the last letter of the second sequence was changed to an A so that the first and second sequences matched each other at a single position. In the low diversity dataset the last position contained only As, so that there is no variation at the last position. The average pairwise distances were reported as equal for all three these datasets even though there clearly are less mismatches in the low diversity dataset than in the high diversity dataset. Inspection of the parameters estimated by PhyML yields large changes in the parameters in the instantaneous rate matrix so that increase in the number of mismatches between the sequences are offset by the changes in the weightings of those mismatches due to the changed parameter values. Nonetheless, the fact that the pairwise distances are computed to be exactly equal is suspect and indicates that PhyML did not perform well in this case of extreme heterogeneity. This poor performance is not surprising given the extreme nature of this example, but does form part of our motivation for removing PhyML from the pipeline in the future. The datasets used in this test are summarized in the second row of Table 40 and the results are shown in the second row of Table 41.

5.4.5.3 *Test 1.3.1: Sequence order for tree-like datasets*

Using the recursive algorithm described in the previous section, a dataset with 128 sequences each of length 400 was simulated allowing up to two mutations between parent and offspring. The predicted average pairwise Hamming distance, using Equation 1, between the sequences in this datasets is 12.11, or, stated differently, mismatches at approximately 3% of the positions. The sequence order was shuffled three times to form three different datasets with identical sequences in different orders. Finding an exact solution to the question of which tree topology and set of mutation parameters best explains the dataset is computationally impractical due to the size of the search space. Hence phylogenetic software use heuristic algorithms which will be affected by initial conditions. The average pairwise distances estimated by PhyML were 0.02589, 0.02614, and 0.02621. The largest of the three values is 1.2% larger than the smallest value. This indicates that the effect of sequence order on the

PhyML estimates is non-trivial. The datasets used in this test are summarized in the third row of Table 40 and the results are shown in the third row of Table 41.

To obtain better insight into the effect of sequence order on the estimated Hamming distances, six datasets were generated, using the same parameters (128 sequences of length 400 with up to two mutations between generations and the recursive algorithm), each of the six datasets was shuffled 30 times, and the average pairwise distances were estimated with PhyML (Figure 72 and Table 38). The average of the average pairwise distance as computed by PhyML ranged from 0.02435 mutations per nucleotide to 0.03034 mutations per nucleotide across the 6 simulated datasets indicating the stochastic nature of the simulation approach. The standard deviations/interquartile ranges for the reshuffled versions of the 6 datasets ranged from 0.0001297 to 0.0001648 mutations per nucleotide, approximately 0.5% the value of the average. Taking the most variable of the six cases considered, this implies that in 95% of cases, the variation introduced into the dataset due to the instability of the algorithm w.r.t. to the sequence order is less than 0.0003 which amounts to 1% of the value of the average. This is a small amount and unlikely to significantly affect downstream analyses.

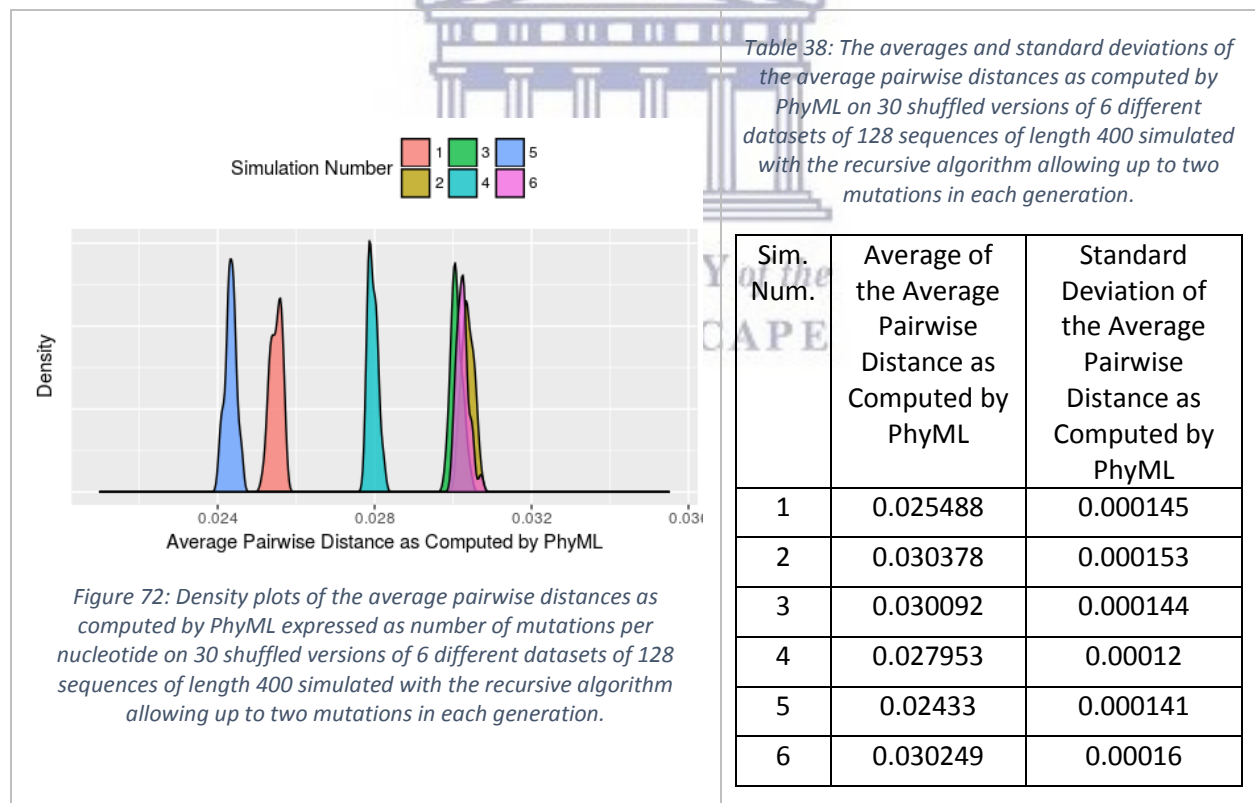


Table 38: The averages and standard deviations of the average pairwise distances as computed by PhyML on 30 shuffled versions of 6 different datasets of 128 sequences of length 400 simulated with the recursive algorithm allowing up to two mutations in each generation.

Sim. Num.	Average of the Average Pairwise Distance as Computed by PhyML	Standard Deviation of the Average Pairwise Distance as Computed by PhyML
1	0.025488	0.000145
2	0.030378	0.000153
3	0.030092	0.000144
4	0.027953	0.00012
5	0.02433	0.000141
6	0.030249	0.00016

5.4.5.4 Test 1.3.2: Sequence order for non-tree-like datasets

When the assumption that the data has an underlying tree-like structure is violated, we expect the estimation procedures to be less reliable and that modifications to the sequence order will have a

larger effect. A dataset was generated using the star-like simulation approach described in the previous section. The dataset had 128 sequences each of length 400. To keep the datasets comparable to those generated using the tree-based approach, a dominance parameter of 0.9697244 was chosen, so that on average two sequences will mismatch at 3.02756% of their positions leading to an average pairwise Hamming distance of 12.11. The order of the sequences was shuffled three times and PhyML was run on each of the three shuffled variations. The average pairwise distances estimated by PhyML were 0.02583, 0.02606, and 0.02629. The largest of the three values is 1.8% larger than the smallest value. This indicates that the effect of sequence order on the PhyML estimates for non-tree-like data is only slightly larger than when the data is tree-like. However, due to the stochastic nature of the simulation approach, this result may be misleading. Thus, a more comprehensive analysis was performed. The datasets used in this test are summarized in the fourth row of Table 40 and the results are shown in the fourth row of Table 41.

As in Test 1.3.1, six datasets were generated, using the same parameters (128 sequences of length 400 with a dominance parameter of 0.9697 and the star-like simulation approach), each of the six datasets was shuffled 30 times, and the average pairwise distances were estimated with PhyML (Figure 73 and Table 39). The average of the average pairwise distance as computed by PhyML ranged from 0.0255 mutations per nucleotide to 0.0269 mutations per nucleotide across the 6 simulated datasets indicating the stochastic nature of the simulation approach. While this range is much smaller than in Test 1.3.1, this does not reflect on PhyML's accuracy, it is primarily a product of the differences between the recursive algorithm and the star-like simulation approach. The interquartile ranges standard deviations for the reshuffled versions of the 6 datasets ranged from 0.000415 to 0.000741 mutations per nucleotide 0.00035 to 0.00064. Taking the most variable of the six cases considered, this implies that in 95% of cases, the variation introduced into the dataset due to the instability of the algorithm w.r.t. to the sequence order is less than 0.001 which amounts to 4.8% of the value of the average. This is 4.8 times larger than the 1% in the case of tree-like datasets. This implies that for 1 in 20 cases the variability introduced by sequence order will be equal to 4.8% of the expected value. While the initial tests based on only three reshuffled versions of a single dataset revealed minor differences between PhyML's ability to estimate pairwise distances based on the underlying structures in the dataset, these additional simulation show that violations of the assumption that the dataset follows a tree-like structure has a large effect on the variability of the estimates. Since this pipeline is aimed at samples soon after infection, we should expect the datasets we operate on to be subject these higher levels of variability due to the order of the sequences in the datasets. This additional variability adds to our decision to remove PhyML from future versions of the pipeline.

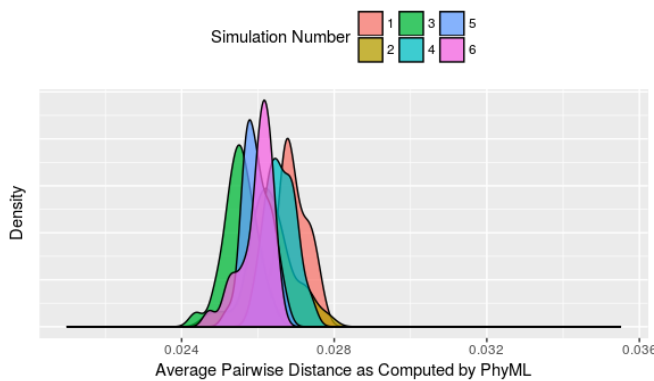


Figure 73: Density plots of the average pairwise distances as computed by PhyML expressed as number of mutations per nucleotide on 30 shuffled versions of 6 different datasets of 128 sequences of length 400 simulated with the star-like algorithm with a dominance parameter of 0.9697.

Table 39: The averages and standard deviations of the average pairwise distances as computed by PhyML on 30 shuffled versions of 6 different datasets of 128 sequences of length 400 simulated with the star-like algorithm with a dominance parameter of 0.9697.

Sim. Num.	Average of the Average Pairwise Distance as Computed by PhyML	Standard deviation of the Average Pairwise Distance as Computed by PhyML.
1	0.02690	0.00042
2	0.02633	0.00064
3	0.02553	0.00042
4	0.02660	0.00038
5	0.02598	0.00035
6	0.02597	0.00043

5.4.5.5 Test 1.4.1: Handling of gaps

To test the handling of gaps, three variations of the balanced dataset described in the data simulation section were made. The low diversity dataset contains no gaps, the target dataset has four gaps inserted at positions 41-44 in the first sequence and the high diversity dataset has those same gaps in all the sequences. The introduction of gaps in a single sequence had no effect on the pairwise distances estimated by PhyML. Adding gaps to all the sequences resulted in a 1.1% increase in the average pairwise distance as estimated by PhyML. The positions that were replaced by all gaps were perfectly conserved. Since replacing them with gaps causes them to be excluded from the analysis because gaps are treated as missing data, the overall diversity of the dataset increased, consistent with the 1.1% increase PhyML reported. Treating a position that consists only of gaps as missing data is a reasonable strategy. The datasets used in this test are summarized in the fifth row of Table 40 and the results are shown in the fifth row of Table 41.

5.4.5.6 Test 1.5.1: Handling of ambiguity letters

The three datasets used in Test 1.4.1 were modified to explore the effects of ambiguity letters that include the non-ambiguity letters present at the same position on the pairwise distances computed by PhyML. The low diversity dataset was left unchanged. Instead of inserting gaps at positions 41-44, the ambiguity letters RYSW were inserted in either only the 1st sequence (target dataset) or the 1st

eight sequences (high diversity dataset). The pattern RYSW was chosen since they *include* the letters that occur in the rest of the sequences at those positions. For example, all unmodified sequences has an A at position 41 and an R can be either an A or a G. Since the ambiguity letters represent the characters found in the other sequences at those positions, we expect these changes to have no, or very minor, effects on the estimated differences. Adding the ambiguity characters to positions 41-44 in only one sequence had no effect on the pairwise distances estimated by PhyML. Adding the ambiguity characters to eight sequences had a tiny effect, increasing the average pairwise distance estimated by PhyML by 0.1%. This effect is only 10% as strong as the effect seen in Test 1.4.1 when gaps were introduced. However, given that these numbers exactly match those observed in Test 1.5.2, it seems apparent that the match between the ambiguity characters and the rest of the letters at the given position has no effect on the estimates. While the prevalence of ambiguity characters is low, ignoring the available data in them is not the optimal approach. The datasets used in this test are summarized in the sixth row of Table 40 and the results are shown in the sixth row of Table 41.

5.4.5.7 Test 1.5.2: Handling of ambiguity letters

The three datasets used in Test 1.5.1 were modified to explore the effects of ambiguity letters that excludes the non-ambiguity letters present at the same position on the pairwise distances computed by PhyML. The low diversity dataset was left unchanged. Instead of inserting the ambiguity letters RYSW at positions 41-44, the ambiguity letters YRWS were inserted in either only the 1st sequence (target dataset) or the 1st eight sequences (high diversity dataset). The pattern YRWS was chosen since they *exclude* the letters that occur in the rest of the sequences at those positions. For example, all unmodified sequences has an A at position 41 and a Y can be either a C or a T. Since the ambiguity letters *does not* represent the characters found in the other sequences at those positions, we expect these changes to have a non-trivial effect on the estimated differences. Adding the ambiguity characters to positions 41-44 in only one sequence had no effect on the pairwise distances estimated by PhyML. Adding the ambiguity characters to eight sequences had a tiny effect, increasing the average pairwise distance estimated by PhyML by 0.1%. This effect was identical to the effect observed when ambiguity letters matching the non-ambiguity letters at the position were introduced, suggesting that ambiguity letter are not handled correctly. The datasets used in this test are summarized in the seventh row of Table 40 and the results are shown in the seventh row of Table 41.

5.4.5.8 Test 1.5.3: Handling of ambiguity letters

To investigate why the results from Test 1.5.1 and Test 1.5.2 were identical, but different from the results of Test 1.4.1, a similar test was run, but using the replacement pattern YYYY. The purpose of this test is to investigate whether or not all non-nucleotide letters are treated the same by PhyML

explaining the different results between the gap-based test (Test 1.4.1) and the ambiguity character based tests (Test 1.5.1, Test 1.5.2). We speculate that PhyML's approach to handling non-standard characters (non-ACGT characters), is to expand the alphabet over which mutation can occur. This expansion of the alphabet hypothesis would be disproved if the replacement patterns YYY and YRWS results in the same average pairwise distance estimates from PhyML, since it would be very unlikely for the addition of a single character to the alphabet to have the same effect as the addition of four letters to the alphabet. The results of this test (replacement pattern YYY) were identical to those of Test 1.5.1 (replacement pattern RYSW) and Test 1.5.2 (replacement pattern YRWS), disproving this hypothesis. Since the ultimate conclusion of this section is that PhyML should be removed from the pipeline, no further investigation was performed. The datasets used in this test are summarized in the eighth row of Table 40 and the results are shown in the eighth row of Table 41.

5.4.5.9 Test 2.1.1: Comparison of different tree-like datasets

Three datasets each with a tree-like structure but with differing levels of diversity were constructed. Sequences were of length 400 and each dataset included 128 sequences. To generate differing levels of diversity, offspring sequences were allowed to differ from their parents with between zero and 1 (for the low diversity dataset), 2 (for the moderately diverse datasets) or 3 (for the high diversity dataset). The ancestral sequence is the pattern ACGT repeated 100 times. The expected average pairwise distances as predicted by Equation 1 for the three dataset were 0.015, 0.030 and 0.045 respectively. The average pairwise distance estimated by PhyML on the moderately diverse sequences was approximately 88% larger than that of the low diversity dataset. Additionally, the average pairwise distance estimated by PhyML on the high diversity dataset was approximately 22% higher than that of the moderately diverse dataset. We expected these numbers to be closer to 100% and 50% respectively, motivating the more detailed analysis described in the next paragraph. The datasets used in this test are summarized in the ninth row of Table 40 and the results are shown in the ninth row of Table 41.

Since simulation with the recursive algorithm introduces stochasticity, measurements based on a single simulation may be misleading. Thirty replicates were simulated for each of the three datasets and the distributions of the averages pairwise distances estimated by PhyML is presented in Figure 74. The average of the average pairwise distance computed by PhyML was 0.0147, 0.0285 and 0.422 for the low, moderate and high diversity datasets respectively. Hence, the moderately diverse datasets' average pairwise distances as estimated by PhyML were double those that were computed on the low diversity datasets, mimicking the doubling of the number of mutations allowed each generation. Likewise, a 50% increase was observed when comparing the high and moderate diversity datasets, clearly illustrating the ability of PhyML's pairwise distance calculations to track the changes

in mutation rates. The average pairwise distance estimates produced by PhyML were very similar to the average pairwise Hamming distances (Figure 76), with PhyML's estimates being 96.6% of the Hamming distances on average. This high correlation simultaneously gives us confidence that PhyML is correctly quantifying the diversity in the datasets and that using PhyML add almost no benefit over using a much simpler metric such as the Hamming distance.

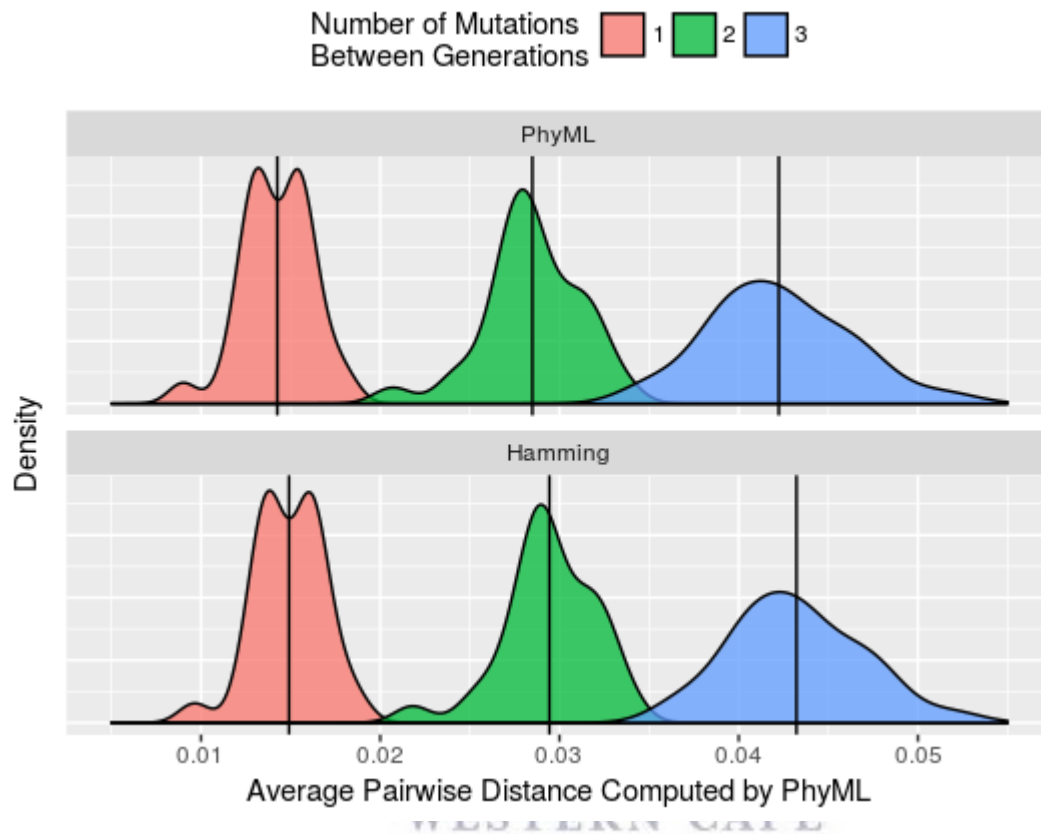


Figure 74: Distributions of average pairwise distances as estimated by PhyML and the average pairwise Hamming distances on datasets of 128 sequences of length 400 simulated by the recursive algorithm allowing 1, 2 or 3 mutations per generation.

5.4.5.10 Test 2.2.1: Handling of distinct subpopulations

The previous test, Test 2.1.1, is designed to test if PhyML's pairwise distance estimates are sensitive to changes to the rate of mutation. Another type of diversity that is of interest is if the patient was infected multiple times. In this case there will be distinct subpopulations within the quasispecies infecting the patient. The three datasets for this test were simulated to have one, two or three different subpopulations corresponding to the low, moderate and high diversity datasets used in the test. For the low diversity dataset, the recursive algorithm was used to simulate 128 sequences of length 400 allowing up to two mutations from parent to offspring using the standard ancestral sequence formed by repeating the pattern ACGT 100 times. The moderate diversity dataset was constructed by combining two datasets each having 64 sequences of length 400 simulated using the recursive algorithm with different ancestral sequences. One ancestral sequence matching that used

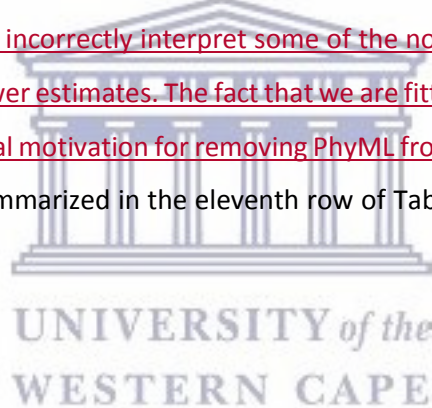
in the low diversity dataset and the other formed by repeating the pattern AAGTCCGT 50 times. Three datasets, two with 32 sequences and one with 64 sequences of length 400 each simulated using the recursive algorithm with different ancestral sequences were combined to form the high diversity datasets. The two datasets with 32 sequences utilized the same ancestral sequences as the moderately diverse dataset and the ancestral sequence for the dataset with 64 sequences was formed by repeating the pattern ACGGACTT 50 times.

For the low diversity dataset, the only source of variation is introduced via the up to two mutations between parent and offspring introduced by the recursive algorithm. Using Equation 1, the predicted average Hamming distance is 12.11 for this dataset. For the target dataset, the average expected pairwise Hamming distance between two sequences from different ancestral sequences is 100, completely dominating the variation introduced by the recursive algorithm. Since the dataset is split 50/50 between the two ancestral sequences, for each sequence half of the sequences will be expected to have a Hamming distance in excess of 100, implying an average pairwise Hamming distance in excess of 50 (0.125 per nucleotide) for the dataset. The expected Hamming distance between one sequence generated with AAGTCCGT pattern and a sequence generated using the ACGGACTT pattern is approximately 200, implying that the expected average pairwise Hamming distance for the high diversity dataset will be approximately 90 (0.225 per nucleotide). PhyML's computed average pairwise distance estimates were 0.0329, 0.1851, and 0.2239 for the low, moderate and high diversity datasets respectively. These observations align well with expectation, demonstrating that the distances computed by PhyML reflect the added diversity resulting from multi-founder populations. The datasets used in this test are summarized in the tenth row of Table 40 and the results are shown in the tenth row of Table 41.

5.4.5.11 *Test 2.3.1: Comparison of different non-tree-like datasets.*

Test 2.1.1 illustrated that the pairwise distances as estimated by PhyML correctly reflects the increased mutation rates in datasets with an underlying tree-like structure. To test if this will hold on datasets that do not have a tree-like structure, three datasets were simulated using the star-like simulation approach described in the previous section. The dominance parameter was set to 0.9849 in the low diversity dataset, 0.9697 in the target dataset and to 0.9545 in the high diversity dataset. These values were chosen so that the expected average pairwise Hamming distances would be comparable to Test 2.1.1. Each dataset had 128 sequences of length 400. The average pairwise distance estimated by PhyML on the moderately diverse were approximately 127% larger than that of the low diversity dataset. Additionally, the average pairwise distance estimated by PhyML on the high diversity dataset was approximately 47% higher than those in the moderately diverse dataset.

Since the simulation process introduces stochasticity, measurements based on a single simulation may be misleading. Thirty replicates were simulated for each of the three datasets and the distributions of the averages pairwise distances estimated by PhyML is presented in Figure 75. The average of the average pairwise distance computed by PhyML was 0.0125, 0.0249, and 0.0375 for the low, moderate and high diversity datasets respectively. Hence, the moderately diverse datasets' average pairwise distances as estimated by PhyML were double that which was computed on the low diversity datasets, mimicking the doubling of the number of mutations allowed each generation. Likewise, a 50% increase as observed when comparing the high and moderate diversity datasets, clearly illustrating the ability of PhyML's pairwise distance calculations to track the changes in mutation rates even when the data does not have an underlying tree-like structure. As in Test 2.1.1, the average pairwise distance estimates produced by PhyML were strongly correlated with the average pairwise Hamming distances (Figure 76), but on the star-like data, PhyML's estimates were noticeably smaller being only 83.8% of the Hamming distances on average. The smaller estimates produced by PhyML is explained by over fitting, i.e. PhyML is imposing a complex tree-like structure on the dataset when there is not such structure. This means that it will incorrectly interpret some of the noise in the data as part of a tree-like structure, resulting in the lower estimates. The fact that we are fitting a complex model to a simple data structure provides additional motivation for removing PhyML from the pipeline in the future. The datasets used in this test are summarized in the eleventh row of Table 40 and the results are shown in the eleventh row of Table 41.



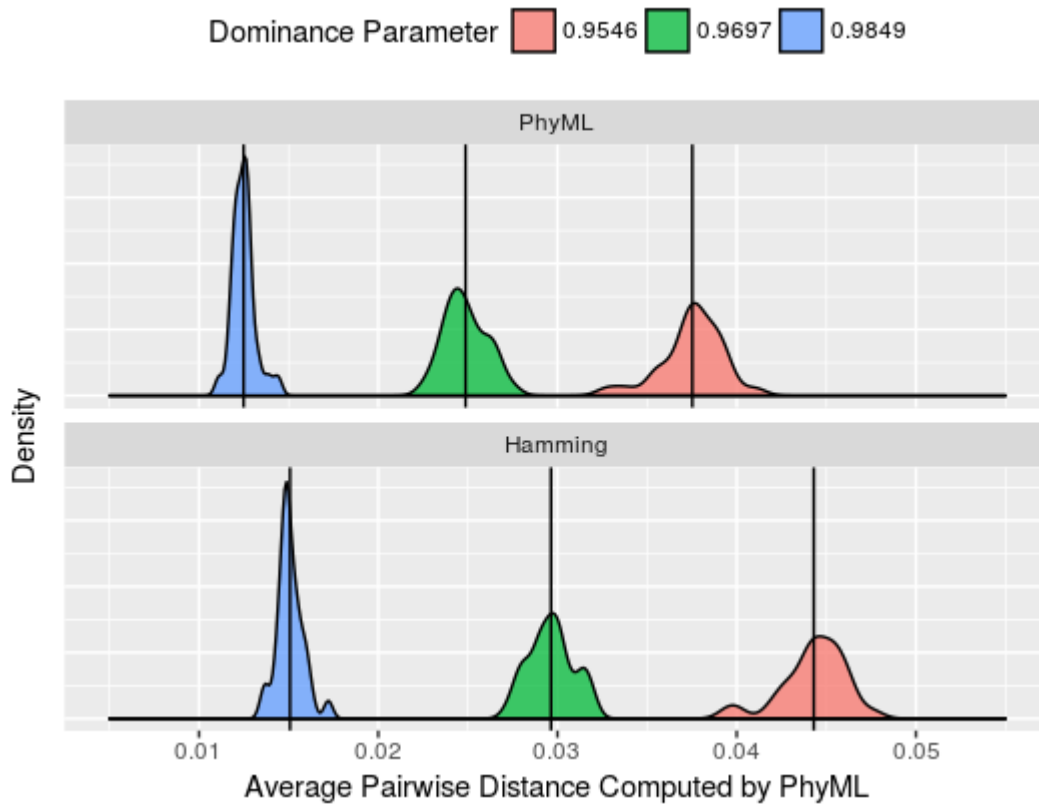


Figure 75: Distributions of average pairwise distances as estimated by PhyML and the average pairwise Hamming distances on datasets of 128 sequences of length 400 simulated by the star-like approach with dominance parameters of 0.98486, 0.96972, or 0.95459.

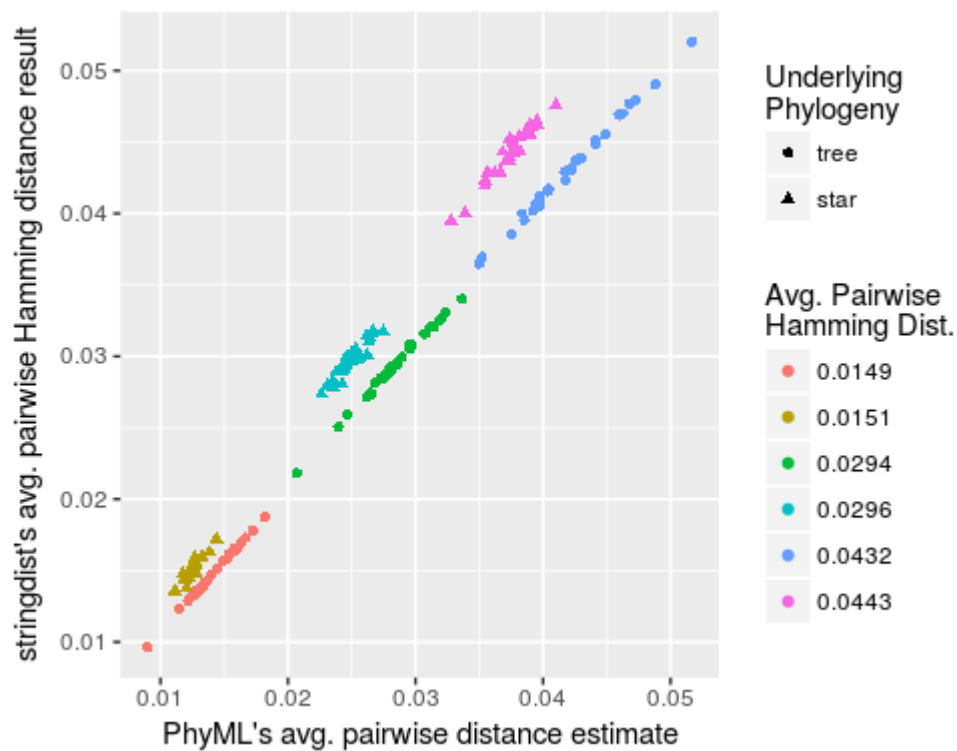


Figure 76: Scatterplot with the average pairwise distance estimated by PhyML plotted against the average pairwise Hamming distance computed with the stringdist R package on the datasets described in tests 2.1.1 and 2.3.1.

5.4.5.12 **Test 2.4.1:** Handling of large datasets of short sequences.

Phylogenetic calculations are often computationally intensive and applying them to large datasets can take prohibitively long or require expensive computer hardware. To check that PhyML will perform adequately on datasets like those generated by Illumina's MiSEQ sequencers, datasets with 1024 sequences each of length 400 were constructed using the recursive algorithm that generated datasets with an underlying tree-structure. Offspring differed from their parents by up to 1, 2 or 3 mutations in the low, moderate and high diversity datasets respectively. Due to the extra generations that are needed to generate the additional 896 sequences, the average predicted Hamming distances, as per Equation 1, were larger than those of the datasets used for Test 2.1.1. The 10 generations required to generate the 1024 sequences means that the 1, 2 and 3 mutations per generation translates Hamming distances of 9.01, 18.02 and 27.03 as predicted by Equation 1. When normalized for sequence length, the Hamming distances are 0.022, 0.045, and 0.068 respectively. The computation took less than 2 hours on each dataset on a high performance laptop (Intel® Core™ i7-4700MQ CPU @ 2.40GHz). Since this pipeline will be used to analyze a small number of dataset 2 hours is an acceptable amount of time. The average pairwise distances as estimated by PhyML were 0.0231, 0.0455 and 0.0635 closely mimicking the Hamming distances predicted by Equation 1. The datasets used in this test are summarized in the twelfth row of Table 40 and the results are shown in the twelfth row of Table 41.

5.4.5.13 **Test 2.5.1:** Handling of large datasets: Long sequences

The computational challenges involved in handling a few long sequences are different from handling many short sequences. This test explores PhyML's capability of processing longer sequences such as those expected from SGA sequencing. Three datasets with underlying tree-like structures were simulated using the recursive algorithm. The datasets were simulated with sequence lengths of 10000, six generations (producing 64 sequences), and up to 44, 88, and 132 mutations per generation. The choices for the number of mutations yields predicted average pairwise Hamming distances of 0.022, 0.045 and 0.067 per base as predicted by Equation 1, which is similar to Test 2.4.1. The computation took less than 4 minutes on each dataset on a high performance laptop (Intel® Core™ i7-4700MQ CPU @ 2.40GHz). The average pairwise distances also reflected the changes in mutation rates correctly. The average pairwise distances as estimated by PhyML were 0.0187, 0.0386 and 0.0601 similar the Hamming distances predicted by Equation 1. The datasets used in this test are summarized in the thirteenth row of Table 40 and the results are shown in the thirteenth row of Table 41.

5.4.5.14 **Test 3.1.1:** Low diversity realistic dataset

The simulations performed in the earlier tests are simplistic and ignore most of the factors that influence the evolution of a quasispecies. The LD_SEQs dataset was simulated based on a low diversity

real world sample as described in section 5.4.4.4. This dataset was used as the moderately diverse dataset. An accompanying low diversity dataset was constructed by generating 872 sequences of length 471 composed of only As. The high diversity dataset was simulated using the star-like simulation approach with a dominance parameter of 0.9 and the consensus sequence of LD_SEQs as the ancestral sequence. The measured average Hamming distances, by tallying mismatches between the sequences, on these datasets are 0, 0.0001, and 0.0912 respectively. PhyML processed these three datasets without incident estimating the average pairwise distances as 0, 0.00002 and 0.07939 respectively, capturing the differences in the diversity of the datasets well. The datasets used in this test are summarized in the fourteenth row of Table 40 and the results are shown in the fourteenth row of Table 41.

5.4.5.15 *Test 3.2.1: High diversity realistic dataset*

Analogous to test 3.1.1, this test is based on a higher diversity dataset designed to mimic a real patient sample more closely. Using the higher diversity HD_SEQs dataset, described in section 5.4.4.4, low and high diversity variants were produced as described in Test 3.1.1, with a low diversity dataset consisting of 691 sequences of length 471 of only As and the high diversity dataset generated using the star-like simulation approach with a dominance parameter of 0.9 and the consensus sequence of HD_SEQs as the ancestral sequence. The measured average pairwise Hamming distances, by tallying mismatches between the sequences, on these datasets are 0, 0.0678, and 0.0964 respectively. PhyML processed these datasets estimating the average pairwise distances as 0, 0.07036, and 0.08476 respectively, reflecting the differing levels of diversity in the datasets well. The datasets used in this test are summarized in the fifteenth row of Table 40 and the results are shown in the fifteenth row of Table 41.

Table 40: Description and rationale of each test for the PhyML and Poisson Fitter sections.

Test Number	Datasets: Low; Target; High	Description and Rationale
T1.1.1	all_a_4_10; all_a_one_pvt_4_10; all_a_two_pvt_4_10	Based on small datasets with only 4 sequences each of length 10. The low diversity dataset contained only As at all positions. A single mutation, changing the first A in the first sequence to a C, was introduced to construct the target (or moderately diverse) dataset. The last dataset for this test was constructed by introducing two mutations, changing the first two As of the first sequence to CT, in the low diversity dataset. Tests if changes in extremely homogenous datasets can be detected.
T1.2.1	same_base_one_pos_nonv_4_10; same_base_one_mut_4_10; same_base_4_10	The datasets with 4 sequences, each of length 10, ranged from a maximally divergent dataset where not one sequence matches another sequence at any position (high), to a dataset where two sequences match each other at one position (target) and a dataset where there is no variation at one position (low). The maximally divergent dataset contains one sequence of only As, one sequence of only Cs, and so forth. Tests if small changes in extremely heterogeneous datasets can be detected.
T1.3.1	tree_2_400_128_o1; tree_2_400_128_o2; tree_2_400_128_o3	Tree-like datasets with seven generations in which each offspring sequence diverges from its parent by up to two mutations with the sequences arranged in three different orders. The datasets contained 128 sequences each of length 400 and the parameters were chosen to produce an expected average pairwise Hamming distance of 12.11, as predicted by Equation 1. The only difference between the datasets is the order of the sequences.
T1.3.2	nont_9697_400_128_o1; nont_9697_400_128_o2; nont_9697_400_128_o3	A dataset was constructed using the star-like approach. The dominance parameter was set to 0.9697244, so that the average expected pairwise Hamming distances between the sequences would be the same as in T1.3.1. The only difference between the datasets is the order of the sequences.
T1.4.1	custom_gaps_none_400_128; custom_gaps_four_400_128; custom_gaps_thr2_400_128	The balanced dataset, as described in the Data Simulation section, was used for this test. The low diversity dataset contains no gaps, the target dataset has four gaps inserted at positions 41-44 in the first sequence and the high diversity dataset has those same gaps in the first 8 sequences. Tests the handling of gaps.
T1.5.1	custom_mambig_none_400_128; custom_mambig_four_400_128; custom_mambig_thr2_400_128	The same as the three datasets constructed for T1.4.1, but instead of inserting gaps at positions 41-44, the ambiguity letters RYSW were inserted in either only the 1 st sequence (target dataset) or the 1 st eight sequences (high diversity dataset). The pattern RYSW was chosen since they <i>include</i> the letters

		that occur in the rest of the sequences at those positions. For example, all unmodified sequences has an A at position 41 and an R can be either an A or a G. Tests the handling of ambiguity characters.
T1.5.2	custom_nambig_none_400_128; custom_nambig_four_400_128; custom_nambig_thr2_400_128	The same as the three datasets constructed for T1.4.1, but instead of inserting gaps at positions 41-44, the ambiguity letters YRWS were inserted in either only the 1 st sequence (target dataset) or the 1 st eight sequences (high diversity dataset). The pattern YRWS was chosen since they <i>exclude</i> the letters that occur in the rest of the sequences at those positions. For example, all unmodified sequences has an A at position 41 and a Y can be either an C or a T. Tests the handling of ambiguity characters.
T1.5.3	custom_sambig_none_400_128; custom_sambig_four_400_128; custom_sambig_thr2_400_128	The same as the three datasets constructed for T1.4.1, but instead of inserting gaps at positions 41-44, the ambiguity letters YYYY were inserted in either only the 1 st sequence (target dataset) or the 1 st eight sequences (high diversity dataset). The pattern YYYY was chosen to investigate if all non-nucleotide letters are handled the same.
T2.1.1	tree_400_128_m1; tree_400_128_m2; tree_400_128_m3	Tree-like datasets with 128 sequences each of length 400 in which the offspring sequences differ from their parents by up to one (low), two (target) or three (high) mutations. The ancestral sequence is just the pattern ACGT repeated 100 times. Tests detection of changes in diversity in tree-like datasets.
T2.2.1	single_400_128; dual_400_128; sdual_400_192	Tree-like datasets in which the offspring sequences differ from their parents by up to two mutation. The datasets were seeded by a single (low), two (target) or three (high) different ancestral sequences. The ancestral sequences are formed by repeating one of the patterns ACGT, AAGTCCGT, or ACGGACTT until the sequence is of length 400. Tests detection of diversity changes in datasets with distinct subpopulations.
T2.3.1	star_data_var_low; star_data_var_target; star_data_var_high	Non tree-like datasets produced by the star-like simulation approach with dominance parameters of 0.9849 (low), 0.9697 (target), or 0.9546 (high). These dominance parameters were chosen so that the expected average pairwise Hamming distances would be similar between tests T2.1.1 and this test (T2.3.1). Tests detection of diversity changes in datasets with star-like phylogeny.
T2.4.1	large_1_600_1024; large_2_600_1024; large_3_600_1024	Large tree-like datasets with 1024 sequences of length 600 in which the offspring sequences differ from their parents by up to one (low), two (target) or three (high) mutations. Designed to mimic Illumina MiSEQ data. The additional generations needed to produce the extra sequences added more diversity to the datasets, so that the predicted average pairwise Hamming distances for the three datasets were 9.01 (low), 18.02 (target) and 27.03 (high). Tests processing of many short sequences.

T2.5.1	<p>large_44_10000_64; large_88_10000_64; large_132_10000_64</p>	<p>Large tree-like datasets with 64 sequences of length 10000 in which the offspring sequences differ from their parents by up to 44 (low), 88 (target) or 132 (high) mutations. Designed to mimic SGA data. The number of mutations per generation was chosen so that the predicted pairwise Hamming distances, 0.0224 (low), 0.0448 (target), and 0.0673 (high), will be similar to that of test T2.4.1 when normalized for sequence length. Tests processing of a few long sequences.</p>
T3.1.1	<p>homo_ld_seqs; ld_seqs; ld_seqs_10perc</p>	<p>Datasets derive from the LD_SEQs dataset which was simulated to mimic data from a patient sample with low diversity. These datasets have 872 sequences and are of length 471. The low diversity dataset has no variation ("A" is the only letter that occur in the dataset). The target dataset is the LD_SEQs dataset described in section 5.4.4.4. The high diversity dataset was simulated using the star-like simulation approach with a dominance parameter of 0.9 and the consensus sequence of LD_SEQs as the ancestral sequence. The measured average Hamming distances, by tallying mismatches between the sequences, on these datasets are 0, 0.0001, and 0.0912 respectively. Tests processing of a low diversity datasets that mimics a real sample.</p>
T3.2.1	<p>homo_hd_seqs; hd_seqs; hd_seqs_10perc</p>	<p>Datasets derive from the HD_SEQs dataset which was simulated to mimic data from a patient sample with low diversity. These datasets have 691 sequences of length 471. The low diversity dataset has no variation ("A" is the only letter that occur in the dataset). The target dataset is the HD_SEQs dataset described in section 5.4.4.4. The high diversity dataset was simulated using the star-like simulation approach with a dominance parameter of 0.9 and the consensus sequence of HD_SEQs as the ancestral sequence. The measured average pairwise Hamming distances, by tallying mismatches between the sequences, on these datasets are 0, 0.0678, and 0.0964. Tests processing of a high diversity dataset that mimics a real sample.</p>

Table 41: Results and brief discussion of the results for each test. Detailed description of the test can be found in Table 40. The description column contains a brief indicator of what the aim of the test is. The Checks 1 to 4 column indicates whether or not the first four checks performed for the test passed or failed. The Low PD stat., Target PD stat. and High PD stat. columns report the average pairwise distance as estimated by PhyML. The Low over Target and Target over High columns reports the ratios of the relevant stats to each other.

Test Number	Description	Checks 1 to 4	Low PD stat.	Target PD stat.	High PD stat.	Low over Target	Target over High	Comment
T1.1.1	Extreme homogeneity	PASS	0.00000	0.05423	0.11964	∞	0.45331	Pairwise distances correctly computed as zero in the case of no diversity. Adding more mutations increases mean pairwise distances.
T1.2.1	Extreme heterogeneity	PASS	0.09167	0.09167	0.09167	1.00000	1.00000	The pairwise distances did not change even though the datasets were designed to have increasing diversity.
T1.3.1	Sequence Order (tree-like)	PASS	0.02614	0.02621	0.02589	1.00261	0.98809	The exact same tree-like dataset presented to PhyML with the sequences in a different order resulted in minor changes in the average pairwise distance metrics.
T1.3.2	Sequence Order (not tree-like)	PASS	0.02606	0.02629	0.02583	1.00906	0.9824	The exact same alignment presented to PhyML with the sequences in a different order resulted in different average pairwise distance metrics.
T1.4.1	Gaps	PASS	0.07572	0.07572	0.07656	1	1.011	We expect gaps to be ignored. This means we expect the introduction of gaps to have a negligible effect. Introduction of gaps in eight sequences increased the PhyML estimated distances by 1%.
T1.5.1	Matching Ambiguous Bases	PASS	0.07572	0.07572	0.0758	1	1.00101	Introducing ambiguous bases in a non-variant position should increase the pairwise distances. Introduction of four ambiguous bases in eight sequences yielded a minor increase.
T1.5.2	Non-Matching Ambiguous Bases	PASS	0.07572	0.07572	0.0758	1	1.00101	Introducing ambiguous bases in a non-variant position should increase the pairwise distances. Introduction of four ambiguous bases in eight sequences yielded a minor increase.

T1.5.3	The Same Ambiguous Base	PASS	0.07572	0.07572	0.0758	1	1.00101	Introducing the same ambiguous base in a non-variant position should increase the pairwise distances. Introduction of four identical ambiguous bases in eight sequences yielded the same increase as in T1.5.1 and T1.5.2, but smaller than the increase in T1.4.1.
T2.1.1	Increasing Diversity (tree-like)	PASS	0.01634	0.03065	0.03729	1.87606	1.21691	Increasing the amount of mutations between parents and offspring in a tree should, as was observed, increase the pairwise distances.
T2.2.1	Multiple Infection	PASS	0.03293	0.18507	0.2239	5.62044	1.20982	Constructing datasets with more founders should, as was observed, increase the pairwise distances.
T2.3.1	Increasing Diversity (not tree-like)	PASS	0.01138	0.02587	0.03805	2.27343	1.47076	Decreasing the dominance parameter results in more mutations away from the ancestral sequence. PhyML's pairwise distances reflected this.
T2.4.1	Mimics Illumina MiSEQ data	PASS	0.02312	0.04549	0.0635	1.96805	1.39573	Increases in diversity can be detected on a large Illumina MiSEQ-like datasets. Large Illumina-like datasets can be processed on available computational platforms.
T2.5.1	Mimics SGA data	PASS	0.01872	0.03858	0.06009	2.06127	1.55746	Increases in diversity can be detected on a large SGA-like dataset. Large SGA-like datasets can be processed on available computational platforms.
T3.1.1	Mimics Patient Sample	PASS	0	0.00002	0.07939	∞	4887.283	Changes in pairwise distances reflects changes in diversity relative to a dataset constructed to mimic a real patient sample with low diversity.
T3.2.1	Mimics Patient Sample	PASS	0	0.07036	0.08476	∞	1.20458	Changes in pairwise distances reflects changes in diversity relative to a dataset constructed to mimic a real patient sample with high diversity.

5.4.6 Future work and Conclusions

Testing phylogenetic software is a complex task because of the large number of parameters they estimate and the complexity of searching for an optimally shaped tree. Furthermore, phylogenetic software is designed to capture complex patterns in the data which requires careful consideration to produce with simulation. Datasets that are simplistic to simulate present unusual challenges to phylogenetic software so that even the most basic tests can result in complex behavior. For example, the most simplistic dataset is a dataset in which all sequences at all positions is just the same letter. While trivial to simulate, this dataset contains no information from which the rates of the other three parameters can be estimated.

PhyML performed poorly in tests for various edge cases (tests T1.1.1 through T1.5.2). Estimates on the highly conserved and diverse datasets were unreliable, resulting in either oversized or no detectable changes in estimated pairwise distances. Shuffling the sequence order of the datasets perturbed PhyML's estimates, but when the underlying data structure was tree-like the effect was limited. The handling of gaps and ambiguity letters were not in line with expectations. The effect of the introduction of gaps was larger than expected since phylogenetic software typically treat gaps as missing data, but inserting gaps in eight out of 128 sequences perturbed PhyML's measurements by 1%. Different IUPAC ambiguity characters should match different nucleotides. Inserting matching or non-matching ambiguity letters had no effect on PhyML's estimates. Further investigation did not reveal additional insight into PhyML's handling these ambiguous characters. If the sequences are non-negligibly ambiguous, it could have substantial impacts and should be evaluated further.

PhyML performed well in the test cases where the datasets were more realistic (tests T2.1.1 through T3.2.1). Phylogenetic software is sensitive to the initial conditions of the fitting algorithm, so that changing the sequence order in a dataset changes PhyML's estimates. The sensitivity of the estimates were 4 times higher in datasets with a star-like structure than in datasets with a tree-like structure. Nonetheless, PhyML's average pairwise distance estimates reflected the diversity patterns that were simulated. In cases where the underlying data structure was tree-like, PhyML's estimates were very similar, smaller by only 3.5% on average, to the observed Hamming distances on the datasets. In star-like datasets, the different types of distances were still strongly correlated, but the actual values differed more, with the PhyML estimates being 16% smaller than the observed Hamming distances on average. PhyML was able to process large datasets within acceptable timeframes and produced realistic estimates on the datasets based on real world samples.

PhyML is only utilized to obtain an average of the pairwise distances between the sequences as part of deciding whether or not the patient was infected multiple times. Other measures of diversity which are much simpler to compute may be used to accomplish this task as shown by the strong correlation between the Hamming distances and PhyML's distance estimates. Furthermore, instability of the estimates under the star like model are problematic given that this model is likely the better descriptor of HIV evolution than the branching phylogeny, especially in the acute phase of infection. Future versions of the pipeline should remove PhyML and base the multi-founder classification on a simpler metric such as the average pairwise Hamming distance. Since the results of PhyML is tightly correlated with the average pairwise Hamming distance, swapping PhyML with the average pairwise Hamming distance will have a negligible effect on the results produced by the pipeline.



5.5 Poisson Fitter

There is a genetic bottleneck during HIV-1 transmission, implying that an infection is often started by a single genetic strain. Additionally, during the initial phase of infection, there is little immune pressure and the virus grows exponentially. In this scenario, it can be shown that the pairwise Hamming distances follows a Poisson distribution. By comparing the pairwise Hamming distances to a Poisson distribution Poisson Fitter determines whether or not the dataset follows a star-like phylogeny. Using a generation time of 2 days and a mutation rate of 0.0000216, Poisson Fitter estimates the time since infection. In those cases where the dataset does follow a star-like phylogeny, this is a reasonable estimate of the time since infection (Elena E Giorgi et al., 2010).

5.5.1 Implementation Details

Poisson Fitter is integrated into the pipeline by a script called `runPoissonFitter.R`. This script generates a distance matrix from the alignment and runs Poisson Fitter by calling the `PFitter.R` script. This `PFitter.R` script is available from github (<https://github.com/pedlefsen/hiv-founder-id>) and contains minor modifications from the one distributed by LANL. The call to `PFitter.R` provides the name of the file that contains the distance matrix, the mutation rate and the number of unique sequences as inputs. The `PFitter.R` script requires a distance matrix as input. Since there are many different ways to compute distances between sequences, this adds a layer of uncertainty to the process. The `PFitter.R` script checks if the dataset is consistent with a star-like phylogeny by comparing the observed data with what is expected under a star-like phylogeny. Two comparisons are performed, one uses a standard Chi-square goodness of fit test while the other is a custom comparison that is described later in this section. It also estimates the time since most recent common ancestor and outputs the results and additional metrics to an output file. The pipeline reads these results and incorporates them into the `identify_founders.tab` file which is the file that collects all the pipeline's results.

The main pipeline script, `identify_founders.pl`, calls the `runPoissonFitter.R` script using the backtick notation and passes information to the `runPoissonFitter.R` script by setting environment variables. For the purposes of this thesis, only two of these variables are relevant:

- 1) `runPoissonFitter_inputFilename` which specifies the name and path to the fasta file that contains the input alignment, and
- 2) `runPoissonFitter_outputDir` which specifies the folder in which `runPoissonFitter.R` is to store its output.

Additional options and variations of Poisson Fitter are also incorporated into the pipeline, these include, but are not limited to, a variation that excludes non synonymous mutations from the analysis

and a variation that, if the sample is found to be multi-founder, splits the dataset into clusters each associated with a single founder and then runs Poisson Fitter independently on each of these clusters. These variations are beyond the scope of this thesis, and will not be described here.

The `runPoissonFitter.R` script generates a consensus for the input file using the consensus function from the `seqinr` package. It also removes all duplicated sequences from the dataset, keeping track of the number of each for use by Poisson Fitter. Using a dataset that consists of the consensus sequence together with all the unique sequences, a distance matrix is computed using the `dist.dna` function from the `ape` package. The distance calculation ignores gaps and ambiguity characters, treating them as matches (Table 42). This distance matrix is converted into a `data.frame` with three columns, the first two naming the two sequences and third recording the distance between those two sequences. This file includes only unique sequences with the number of sequences identical to the current sequences reported with an integer appended to the names of the sequences.

Table 42: Pairwise Hamming distances computed between 5 short sample sequences using `dist.dna` from the `ape` package.

Sequence	S1	S2	S3	S4	S5
S1 ACGTACGTACGT	0				
S2 ACGTACGTACGT	0	0			
S3 ACGTAATTACGT	2	2	0		
S4 ACGTA--TACGT	0	0	0	0	
S5 ACGTANNACGT	0	0	0	0	0

The calculation performed by Poisson Fitter is based on two assumptions. The first is that most HIV-1 infections are founded by a single virus. The second is that during the early stages of infection, the immune system is ineffective at inhibiting the growth of the virus, so that the viral population grows exponentially. Under these assumptions the distribution of the frequencies of the pairwise Hamming distances in the dataset is Poisson with parameter λ . The λ parameter can be estimated using the equation

$\lambda = \frac{\sum_{i=0}^n iY_i}{\sum_{i=0}^n Y_i} = E(Y)$	<i>Equation 3</i>
---	-------------------

where $Y = (Y_0, \dots, Y_n)$ are the pairwise Hamming distance frequency counts in which the subscript gives the Hamming distance between the sequences. Using a model for the growth of the quasispecies in a patient, this λ can be converted into a time since infection with the equation derived in (Lee et al., 2009).

The first step in the `Pfitter.R` script is to compute the frequencies of Hamming distances between the sequences and the consensus, denoted by $X = (X_1, \dots, X_m)$ where i is the distance between the

sequence and the consensus, as well as all the pairwise distances between the sequences, denoted by $Y = (Y_0, \dots, Y_n)$ as above with $n = 2m$. Using Equation 3, the number of days since infection can then be estimated. Under a star-like phylogeny, the frequencies of the pairwise Hamming distances are related to the frequencies of the Hamming distances between the sequences and the consensus sequence by Equation 4

$\bar{Y}_k = \frac{1}{2} \sum_{i=0}^k (X_{k-i}X_i - \delta_{i,k-i}X_k)$	Equation 4
---	------------

for $k = 1, \dots, n$ and $\delta_{ij} = 1$ if $i = j$ and 0 otherwise. By comparing the \bar{Y}_k values with the Y_i values using a Chi-square goodness of fit test, a p-value can be obtained for the hypothesis that the dataset has a star-like phylogeny. In addition to the Chi-square goodness of fit test, Poisson Fitter performs a check that is similar in nature to the Chi-square goodness of fit test. It compares the value Equation 5

$\frac{\sum_{i=0}^n \bar{Y}_i - Y_i }{\sum_{i=0}^n Y_i}$	Equation 5
---	------------

to 0.1. If the value computed with Equation 5 is less than or equal to 0.1, then the dataset is determined to have a star-like phylogeny. The motivation for this alternative check is explored later in this section.

Table 43: Observed (OBS) and expected (CONV) frequencies of Hamming Distances (HD) for the dataset with star-like phylogeny used in test 1.3.2.

HD	OBS	CONV	HD	OBS	CONV	HD	OBS	CONV
0	0	0	10	713	681	19	200	211
1	0	0	11	898	871	20	131	150
2	4	3	12	878	872	21	74	89
3	8	6	13	874	875	22	31	45
4	38	37	14	811	818	23	19	26
5	69	63	15	637	658	24	10	12
6	158	146	16	558	579	25	5	8
7	287	261	17	387	405	26	NA	1
8	416	382	18	288	331	27	NA	0
9	634	598						

The Pfinder.R script produces six output files, two containing results and four with plots comparing the distributions to each other. The result files are titled CONVOLUTION.results.txt and LOG_LIKELIHOOD.results.txt. These files are produced in the output folder passed with runPoissonFitter_outputDir environment variable. The CONVOLUTION.results.txt file lists the observed and expected Hamming distance frequencies (Table 43) as well as a determination about whether or not the sample follows a star-like phylogeny based on the check mechanism that uses Equation 5. The LOG_LIKELIHOOD.results.txt file lists the various statistics the Pfinder.R script

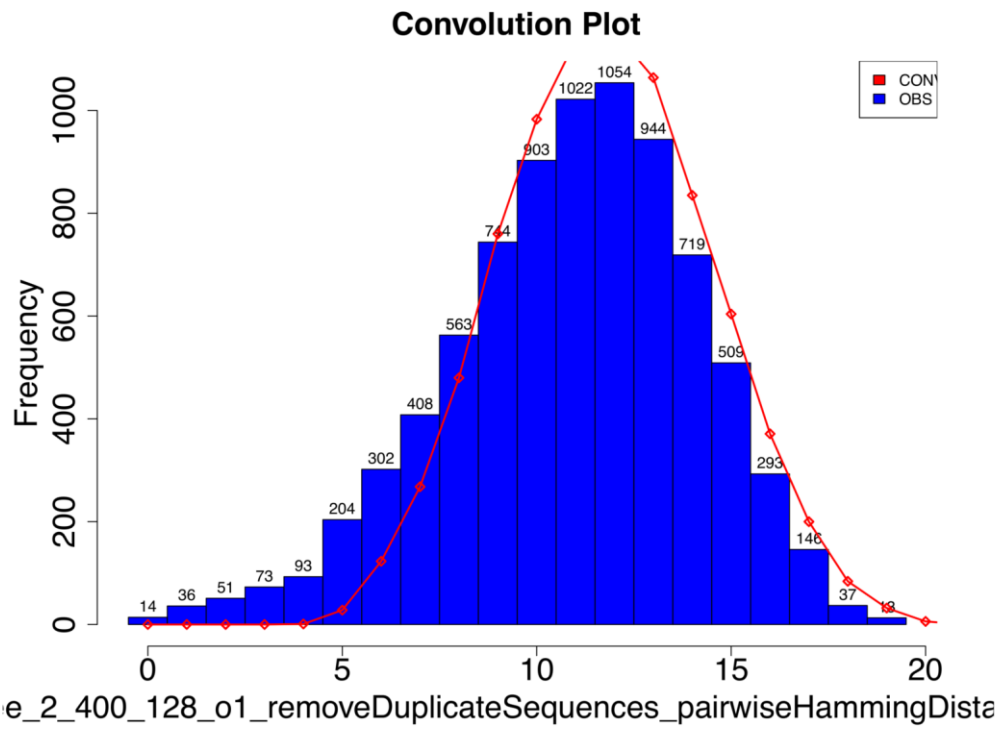
estimates (Table 44). The plots show the information of the CONVOLUTION.results.txt file in graphical format (Figure 77).

When considering all pairwise distances between a set of sequences, the number of individual observations (the pairwise distances) become large very quickly. In a dataset of 128 sequences, there are 8128 pairwise distances. With large sample sizes, statistical tests become extremely sensitive flagging minute variations as statistically significant. The dataset from which part A of Figure 77 was constructed was simulated with an underlying tree-like structure. The deviation of the observed and expected frequencies are clear and systematic with more than expected sequences having very small deviations from each other. However the dataset from which part B of Figure 77 was constructed was specifically based on a dataset with a star-like structure. For part B, there is a strong concordance between the observed and expect values, but with some fluctuations. The p-value for the Chi-square goodness of fit test for this dataset is less than $2 \cdot 10^{-16}$, leading to the rejection of the null hypothesis that the dataset follows a star-like phylogeny. Since the Chi-square goodness of fit test is so overpowered in this situation, using a less sensitive criterion for classifying the datasets into star-like or non-star-like is warranted. Whether or not the use of this alternative check is optimal remains an open question that should be formally investigated.

Table 44: Outputs produced by the PFitter.R script.

Variable Name	Description
Sample	The path to and the name of the input file, but excluding the extension of the file.
Lambda	The parameter of the Poisson distribution estimated from the observed Hamming distance frequencies.
St.Dev	The standard deviation of Lambda estimated using the approach described in (E E Giorgi & Bhattacharya, 2012)
NSeq	The number of sequences in the dataset.
NBases	The length of the alignment.
MeanHD	The average pairwise Hamming distance.
MaxHD	The maximum pairwise Hamming distance.
Days(CI)	The estimated time since infection computed using the model described in (Lee et al., 2009)
Chi2	The test statistic of the Chi-square goodness of fit test.
DF	The degrees of freedom used in the Chi-square goodness of fit test.
Goodness_of_pval	The p-value returned by the Chi-square goodness of fit test.

A



B

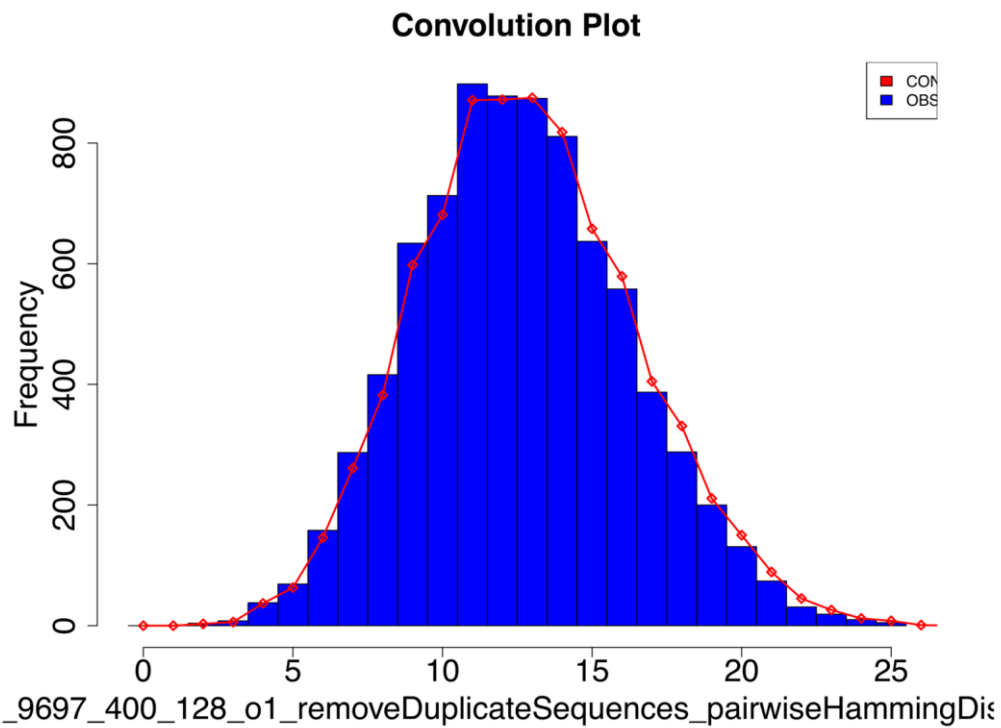


Figure 77: Plots comparing the observed Hamming distance frequencies (blue bars) to the expected frequencies (red line) for a dataset with a tree-like phylogeny (A on the top) and a dataset with a star-like phylogeny (B on the bottom).

5.5.2 Tests and Examples

Testing the integration of Poisson Fitter into the pipeline focused on three aspects: Certainty that the correct datasets were analyzed, basic correctness of the results (such as the estimate must be within the confidence interval that surrounds it) and relative comparisons on the time since infection estimates on datasets that were specifically constructed to have known relationships to each other. The same datasets that were used to test PhyML were used to test Poisson Fitter.

5.5.2.1 Overview of testing procedure

The general organization of the tests is the same as that used for PhyML, for more details, see section 5.4.3 and Table 40. As for PhyML, the tests each contain three datasets, one with low diversity, one with moderate diversity (also referred to as target), and one with high diversity. The checks that are performed for each test is different from what was used to evaluate PhyML due to the differences between PhyML and Poisson Fitter.

First Poisson Fitter should have run without errors on all three datasets. The two files that contain the results and the comparison of the observed and expected distributions of pairwise Hamming distances must both exist. The determination of whether or not the dataset exhibits star like phylogeny is expected to be consistent with the p-value that was observed. Due to the over sensitivity of the Chi-square goodness of fit test, Poisson Fitter also provides an alternative check. The use of this alternative check is not formally motivated thus any deviations between it and the decision based on the p-value from the Chi-square goodness of fit test is considered as a failed test. The estimated time since infection must lie within the confidence interval for the estimated time since infection. The number of sequences analyzed by Poisson Fitter must equal the number of sequences in the input file and the length of the sequences must match. Finally, both the time since infection and the mean pairwise distance between the sequences must be larger (or in some cases equal) on the moderate diversity dataset than on the low diversity dataset. Likewise, both these quantities must be larger (or in some cases equal) on the high diversity datasets than on the moderate diversity datasets.

5.5.3 Test results and discussion

In total 15 tests were performed which each compare three datasets to each other. Summarized description of the tests and datasets are listed in Table 40. The results and brief comments about the results are presented in Table 45. In this section, each test will be described in detail and the results will be interpreted. The mean Hamming distance and the estimated time since infection reported by Poisson Fitter will be evaluated against the mean Hamming distance and estimated time since infection obtained on the other datasets that comprise the test.

In cases where unexpected results occurred, comparisons are made between the average pairwise distances computed by Poisson Fitter and metrics computed on the test datasets themselves. Two such metrics are used a predicted pairwise diversity and a computed pairwise Hamming distance. Equation 1 and Equation 2 is used to predict the Hamming distance for a simulated dataset. Since the simulation process is stochastic, the true average pairwise Hamming distance of the dataset can vary from the predicted amount as demonstrated in Figure 72 and Figure 73.

5.5.3.1 *Test 1.1.1: Extreme homogeneity*

Using very small datasets with only four sequences each of length 10, the ability of Poisson Fitter to process and estimate infection times on extremely homogenous datasets were tested. The low diversity dataset contained only As at all positions. A single mutation, changing the first A in the first sequence to a C, was introduced to construct the target (or moderately diverse) dataset. The last dataset for this test was constructed by introducing two mutations relative to the low diversity dataset. These two mutations were changing the first position in the first sequence to an A and changing the second position in the first sequence to a T. Poisson Fitter cannot be run on a dataset in which all pairwise distances are zero, so the checks for the low diversity dataset failed. The estimated infection times increased as one (moderate diversity dataset) or two mutations (high diversity dataset) were added. These increases were large, adding approximately 2100 days to the time since infection each. This is because when the sequences are so short, the presence of any mutation is highly unlikely unless large amounts of time have elapsed. The datasets were identified as star-like, which was expected since there are not enough mutations for any kind of structure to be discerned. This test is designed to be an edge case which pushed the algorithm to its limits and as such extreme results are expected. These extreme results does not affect our confidence in the algorithm since this case is not designed to test the algorithm, rather it is used to stress test the implementation of the algorithm. Since the software ran without error and produced results that can be explained sensibly, we consider this a successful test. The datasets used in this test are summarized in the first row of Table 40 and the results are shown in the first row of Table 45.

5.5.3.2 *Test 1.2.1: Extreme heterogeneity*

Extreme diversity in the datasets was simulated by using four sequences of length 10, the first sequence containing only As, the second sequence only Cs and so forth. The implication is that for the most diverse dataset, at no position does any sequence match any other sequence. To reduce the diversity for the target dataset, the last letter of the second sequence was changed to an A so that the first and second sequences matched each other at a single position. In the low diversity dataset the last position contained only As, so that there is no variation at the last position. Poisson Fitter was able

to differentiate the three datasets from each other correctly ranking the datasets by time since infection. As expected, the estimates for time since infection are extremely large since a very large amount of time would be required for all sequences to mismatch all other sequences at all positions. As in test 1.1.1, these extreme results are not concerning. The datasets used in this test are summarized in the second row of Table 40 and the results are shown in the second row of Table 45.

5.5.3.3 *Test 1.3.1: Sequence order for tree-like datasets*

Using the recursive algorithm described in the previous section, a dataset with 128 sequences each of length 400 was simulated, allowing up to two mutations between parent and offspring. The predicted average pairwise Hamming distance, using Equation 1, between the sequences in this dataset is 12.11, or, stated differently, mismatches at approximately 3% of the positions. The sequence order was shuffled three times to form three different datasets with identical sequences in different orders. The Poisson Fitter algorithm is exact and obtained identical results, proving itself to be invariant with respect to changes in sequence order. The datasets were correctly flagged as non-star-like. The datasets used in this test are summarized in the third row of Table 40 and the results are shown in the third row of Table 45.

5.5.3.4 *Test 1.3.2: Sequence order for non-tree-like datasets*

When the assumption that the data has an underlying tree-like structure is violated, we expect the estimation procedures to be less reliable and that modifications to the sequence order will have a larger effect. A dataset was generated using the star-like simulation approach described in the previous section. The dataset had 128 sequences each of length 400. To keep the datasets comparable to those generated using the tree-based approach, a dominance parameter of 0.9697244 was chosen, so that on average two sequences will mismatch at 3.02756% of their positions leading to an average pairwise Hamming distance of 12.11. The order of the sequences was shuffled three times and Poisson Fitter was run on each of the three shuffled variations. Again, the Poisson Fitter results were invariant with respect to sequence order and all three datasets were correctly identified as star-like. The checks failed since the p-values from the goodness of fit test was not consistent with the check identifying the datasets as star-like. The determination based on the check (Equation 5) is in line with our expectation while the goodness of fit test incorrectly flags the dataset as non-star-like. The datasets used in this test are summarized in the fourth row of Table 40 and the results are shown in the fourth row of Table 45.

5.5.3.5 *Test 1.4.1: Handling of gaps*

To test the handling of gaps, three variations of the balanced dataset described in the data simulation section were made. The low diversity dataset contains no gaps, the target dataset has four gaps

inserted at positions 41-44 in the first sequence and the high diversity dataset has those same gaps in all the sequences. The introduction of gaps in a single sequence had no effect on Poisson Fitter, as expected since gaps are treated as missing data (Table 42). Adding gaps to all the sequences resulted in a 1% increase in the estimated time since infection, primarily driven by the removal of the four perfectly conserved positions from the dataset, so that the average diversity in the dataset was increased a little. [This approach of treating positions consisting of only gaps is a reasonable approach.](#) The datasets used in this test are summarized in the fifth row of Table 40 and the results are shown in the fifth row of Table 45.

5.5.3.6 Test 1.5.1: Handling of ambiguity letters

The three datasets used in Test 1.4.1 were modified to explore the effects of ambiguity letters that *include* the non-ambiguity letters present at the same position. The low diversity dataset was left unchanged. Instead of inserting gaps at positions 41-44, the ambiguity letters RYSW were inserted in either only the 1st sequence (target dataset) or the 1st eight sequences (high diversity dataset). The pattern RYSW was chosen since they *include* the letters that occur in the rest of the sequences at those positions. For example, all unmodified sequences has an A at position 41 and an R can be either an A or a G. Since the ambiguity letters are ignored by the distance matrix computation step, we expect the results on these three datasets to be identical. The estimated times since infection was identical for all three datasets in this test. [A future improvement to the pipeline should add more sophisticated handling of ambiguity letters to the distance matrix computation step.](#) The datasets used in this test are summarized in the sixth row of Table 40 and the results are shown in the sixth row of Table 45.

5.5.3.7 Test 1.5.2: Handling of ambiguity letters

The three datasets used in Test 1.5.1 were modified to explore the effects of ambiguity letters that *excludes* the non-ambiguity letters present at the same position. The low diversity dataset was left unchanged. Instead of inserting the ambiguity letters RYSW at positions 41-44, the ambiguity letters YRWS were inserted in either only the 1st sequence (target dataset) or the 1st eight sequences (high diversity dataset). The pattern YRWS was chosen since they *exclude* the letters that occur in the rest of the sequences at those positions. For example, all unmodified sequences has an A at position 41 and a Y can be either a C or a T. Since the ambiguity letters *does not* represent the characters found in the other sequences at those positions, the ideal case would be for these changes to have a non-trivial effect on the estimated differences. Since ambiguity letters are ignored by the distance matrix computation step, the estimated times since infection were identical for all three datasets. [A future improvement to the pipeline should add more sophisticated handling of ambiguity letters to the](#)

[distance matrix computation step](#). The datasets used in this test are summarized in the seventh row of Table 40 and the results are shown in the seventh row of Table 45.

5.5.3.8 *Test 1.5.3: Handling of ambiguity letters*

The three datasets used in Test 1.5.1 were modified to explore the effects of ambiguity letters that excludes the non-ambiguity letters present at the same position. The low diversity dataset was left unchanged. Instead of inserting the ambiguity letters RYSW at positions 41-44, the ambiguity letters YYY were inserted in either only the 1st sequence (target dataset) or the 1st eight sequences (high diversity dataset). This test was designed to test a specific proposition about how PhyML handles ambiguity characters and is only included here for completeness. As in Test 1.5.1 and Test 1.5.2, the estimated times since infection was identical for all three datasets. [A future improvement to the pipeline should add more sophisticated handling of ambiguity letters to the distance matrix computation step](#). The datasets used in this test are summarized in the eighth row of Table 40 and the results are shown in the eighth row of Table 45.

5.5.3.9 *Test 2.1.1: Comparison of different tree-like datasets*

Three datasets each with a tree-like structure but with differing levels of diversity were constructed. Sequences were of length 400 and each dataset included 128 sequences. To generate differing levels of diversity, offspring sequences were allowed to differ from their parents with between zero and 1 (for the low diversity dataset), 2 (for the moderately diverse datasets) or 3 (for the high diversity dataset). The ancestral sequence is the pattern ACGT repeated 100 times. The expected average pairwise distances as predicted by Equation 1 for the three dataset were 0.015, 0.030 and 0.045 respectively. The estimated time since infection on the moderately diverse was approximately double that of the low diversity dataset. Additionally, the estimated time since infection of the high diversity dataset was approximately 22% higher than that of the moderately diverse dataset. While deviating from the predicted ratios, this is within the expected ranges produced by the stochastic nature of the simulations as demonstrated in section 5.4.5.9. The datasets used in this test are summarized in the ninth row of Table 40 and the results are shown in the ninth row of Table 45.

5.5.3.10 *Test 2.2.1: Handling of distinct subpopulations*

The previous test, Test 2.1.1, is designed to test if Poisson Fitter is sensitive to changes to the rate of mutation. Another type of diversity that is of interest is if the patient was infected multiple times. In this case there will be distinct subpopulations within the quasispecies infecting the patient. The three datasets for this test were simulated to have one, two or three different subpopulations corresponding to the low, moderate and high diversity datasets used in the test. For the low diversity dataset, the recursive algorithm was used to simulate 128 sequences of length 400 allowing up to two

mutations from parent to offspring using the standard ancestral sequence formed by repeating the pattern ACGT 100 times. The moderate diversity dataset was constructed by combining two datasets each having 64 sequences of length 400 simulated using the recursive algorithm with different ancestral sequences. One ancestral sequence matching that used in the low diversity dataset and the other formed by repeating the pattern AAGTCCGT 50 times. Three datasets, two with 32 sequences and one with 64 sequences of length 400 each simulated using the recursive algorithm with different ancestral sequences were combined to form the high diversity datasets. The two datasets with 32 sequences utilized the same ancestral sequences as the moderately diverse dataset and the ancestral sequence for the dataset with 64 sequences was formed by repeating the pattern ACGGACTT 50 times.

For the low diversity dataset, the only source of variation is introduced via the up to two mutations between parent and offspring introduced by the recursive algorithm. Using Equation 1, the predicted average Hamming distance is 12.11 for this dataset. For the target dataset, the average expected pairwise Hamming distance between two sequences from different ancestral sequences is 100, completely dominating the variation introduced by the recursive algorithm. Since the dataset is split 50/50 between the two ancestral sequences, for each sequence half of the sequences will be expected have a Hamming distance in excess of 100, implying an average pairwise Hamming distance in excess of 50 (0.125 per nucleotide) for the dataset. The expected Hamming distance between one sequence generated with AAGTCCGT pattern and a sequence generated using the ACGGACTT pattern is approximately 200, implying that the expected average pairwise Hamming distance for the high diversity dataset will be approximately 90 (0.225 per nucleotide). Poisson Fitter's estimated times since infection was 1420, 6222 and 10125 for the low, moderate and high diversity datasets respectively. These numbers correlate well with the average pairwise distances expected on the datasets, but reveals Poisson Fitter's inaccuracy in the case of multiple founders. Features are included in the pipeline that correct the estimates in the case of infection by multiple variants, but these features are beyond the scope of this thesis. The datasets used in this test are summarized in the tenth row of Table 40 and the results are shown in the tenth row of Table 45.

5.5.3.11 *Test 2.3.1: Comparison of different non-tree-like datasets.*

Test 2.1.1 illustrated that the estimated time since infection correctly reflects the increased mutation rates in datasets with an underlying tree-like structure. To test if this will hold on datasets that do not have a tree-like structure, three datasets were simulated using the star-like simulation approach described in the previous section. The dominance parameter was set to 0.9849 in the low diversity dataset, 0.9697 in the target dataset and to 0.9545 in the high diversity dataset. These values were chosen so that the expected average pairwise Hamming distances would be comparable to Test 2.1.1. Each dataset had 128 sequences of length 400. The estimated time since infection on the moderately

diverse were approximately 125% larger than that of the low diversity dataset. Additionally, the estimated time since infection on the high diversity dataset was 49% higher than those in the moderately diverse dataset which align well with what is expected given the choice of dominance parameters. All datasets were correctly identified as star-like datasets. Again, the checks failed since the p-values from the goodness of fit test was not consistent with identifying the datasets as star-like, but the software still made the determination correctly, pointing to a flaw in the reporting process. The datasets used in this test are summarized in the eleventh row of Table 40 and the results are shown in the eleventh row of Table 45.

5.5.3.12 *Test 2.4.1: Handling of large datasets of short sequences.*

To check that Poisson Fitter will process datasets like those generated by Illumina's MiSEQ sequencers quickly and accurately, datasets with 1024 sequences each of length 400 were constructed using the recursive algorithm that generated datasets with an underlying tree-structure. Offspring differed from their parents by up to 1, 2 or 3 mutations in the low, moderate and high diversity datasets respectively. Due to the extra generations that are needed to generate the additional 896 sequences, the average predicted Hamming distances, as per Equation 1, were larger than those of the datasets used for Test 2.1.1. The 10 generations required to generate the 1024 sequences means that the 1, 2 and 3 mutations per generation translates Hamming distances of 9.01, 18.02 and 27.03 as predicted by Equation 1. When normalized for sequence length, the Hamming distances are 0.022, 0.045, and 0.068 respectively. The computation took less than one hour on each dataset on a high performance laptop (Intel® Core™ i7-4700MQ CPU @ 2.40GHz). Since this pipeline will be used to analyze a small number of dataset an hour is an acceptable amount of time. The estimated times since infection were 987, 1916 and 2643. These estimates maintains the ratios between the diversity of the datasets as predicted by Equation 1. The datasets used in this test are summarized in the twelfth row of Table 40 and the results are shown in the twelfth row of Table 45.

5.5.3.13 *Test 2.5.1: Handling of large datasets: Long sequences*

The computational challenges involved in handling a few long sequences are different from handling many short sequences. This test explores Poisson Fitter's capability of processing longer sequences such as those expected from SGA sequencing. Three datasets with underlying tree-like structures were simulated using the recursive algorithm. The datasets were simulated with sequence lengths of 10000, six generations (producing 64 sequences), and up to 44, 88, and 132 mutations per generation. The choices for the number of mutations yields predicted average pairwise Hamming distances of 0.022, 0.045 and 0.067 per base as predicted by Equation 1, which is similar to Test 2.4.1. The computation took less than 4 minutes on each dataset on a high performance laptop (Intel® Core™ i7-4700MQ CPU

@ 2.40GHz). The ratios between the estimated times since infection reflected the ratios expected to result from the changes in mutation rates correctly. The estimated times since infection were 827, 1688 and 2593. The datasets used in this test are summarized in the thirteenth row of Table 40 and the results are shown in the thirteenth row of Table 45.

5.5.3.14 *Test 3.1.1: Low diversity realistic dataset*

The simulations performed in the earlier tests are simplistic and ignore most of the factors that influence the evolution of a quasispecies. The LD_SEQs dataset was simulated based on a low diversity real world sample as described in section 5.4.4.4. This dataset was used as the moderately diverse dataset. An accompanying low diversity dataset was constructed by generating 872 sequences of length 471 composed of only As. The high diversity dataset was simulated using the star-like simulation approach with a dominance parameter of 0.9 and the consensus sequence of LD_SEQs as the ancestral sequence. The measured average Hamming distances, by tallying mismatches between the sequences, on these datasets are 0, 0.0001, and 0.0912 respectively. Poisson Fitter was unable to process the low diversity dataset since all the sequences were identical. The estimated time since infection on the moderate and high diversity datasets were 4 and 3830. These numbers accurately reflect the diversity of the datasets. Furthermore, the moderately diverse dataset was flagged as a star-like dataset which is expected since the extremely low diversity in the dataset will make it difficult to detect any structure and the dataset is based on a real world dataset which was sampled very soon after infection. The datasets used in this test are summarized in the fourteenth row of Table 40 and the results are shown in the fourteenth row of Table 45.

5.5.3.15 *Test 3.2.1: High diversity realistic dataset*

Analogous to Test 3.1.1, this test is based on a higher diversity dataset designed to mimic a real patient sample more closely. Using the higher diversity HD_SEQs dataset, described in section 5.4.4.4, low and high diversity variants were produced as described in Test 3.1.1, with a low diversity dataset consisting of 691 sequences of length 471 of only As and the high diversity dataset generated using the star-like simulation approach with a dominance parameter of 0.9 and the consensus sequence of HD_SEQs as the ancestral sequence. The measured average pairwise Hamming distances, by tallying mismatches between the sequences, on these datasets are 0, 0.0678, and 0.0964 respectively. Poisson Fitter was unable to process the low diversity dataset since all the sequences were identical. The estimated time since infection on the moderate and high diversity datasets were 2844 and 4048. These numbers accurately reflect the diversity of the datasets. Furthermore, the moderately diverse dataset was flagged as non-star-like dataset which is expected since the dataset is based on a real world dataset which was sampled long after infection. The datasets used in this test are

summarized in the fifteenth row of Table 40 and the results are shown in the fifteenth row of Table 45.



Table 45: Results and brief discussion of the results for each test. Detailed description of the test can be found in Table 40. The description column contains a brief indicator of what the aim of the test is. The Checks 1 to 7 column indicates whether or not the first seven checks performed for the test passed or failed. The Low ETSI; Avg. HD, Target ETSI; Avg. HD and High ETSI; Avg. HD columns report the estimated time since infection (ETSI) and the average pairwise Hamming Distance that was computed by *dist.dna*. The Low over Target ETSI; Avg. HD and Target over High ETSI; Avg. HD columns reports the ratios of the relevant stats to each other.

Test Number	Description	Checks 1 to 7	Low ETSI; Avg. HD	Target ETSI; Avg. HD	High ETSI; Avg. HD	Low over Target ETSI; Avg HD	Target over High ETSI; Avg HD	Star (low; target; high)	Comment
T1.1.1	Extreme homogeneity	Fail	0; 0	2099; 0.5	4197; 1	0; 0	0.5; 0.5	True; True; True	If all sequences are identical, Poisson Fitter can't process the dataset and then returns a dummy result and issues a warning, causing some of the checks to fail on the low diversity dataset.
T1.2.1	Extreme heterogeneity	Pass	37772; 9	41270; 9.8	41969; 10	0.92; 0.92	0.98; 0.98	False; False; False	Small differences in diversity can be successfully detected in extremely diverse datasets.
T1.3.1	Sequence Order (tree-like)	Pass	1144; 11	1144; 11	1144; 11	1; 1	1; 1	False; False; False	Sequence order has no effect.
T1.3.2	Sequence Order (not tree-like)	Fail	1323; 13	1323; 13	1323; 13	1; 1	1; 1	True; True; True	The metric used to check if the sample is star-like is not the p-value from the goodness of fit test. Sequence order has no effect.
T1.4.1	Gaps	Fail	2974; 28	2974; 28	3004; 28	1; 1	0.99; 1	False; False; False	Computation of the confidence intervals produced NaNs. The columns that consists of only gaps were removed so Poisson Fitter reported operating on fewer positions than were present in the input dataset.
T1.5.1	Matching Ambiguous Bases	Fail	2974; 28	2974; 28	2974; 28	1; 1	1; 1	False; False; False	Computation of the confidence intervals produced NaNs. Adding ambiguous characters has no effect on the calculations.

T1.5.2	Non-Matching Ambiguous Bases	Fail	2974; 28	2974; 28	2974; 28	1; 1	1; 1	False; False; False	Computation of the confidence intervals produced NaNs. Adding ambiguous characters has no effect on the calculations.
T1.5.3	The Same Ambiguous Base	Fail	2974; 28	2974; 28	2974; 28	1; 1	1; 1	False; False; False	Computation of the confidence intervals produced NaNs. Adding ambiguous characters has no effect on the calculations.
T2.1.1	Increasing Diversity (tree-like)	Pass	710; 6.8	1334; 13	1621; 15	0.53; 0.52	0.82; 0.87	False; False; False	Estimated time since infection increases as diversity increases. Samples correctly identified as non-star-like.
T2.2.1	Multiple Infection	Fail	1420; 14	6222; 59	10125; 96	0.23; 0.24	0.61; 0.61	False; False; False	Computation of the confidence intervals produced NaNs on the target dataset. Estimated time since infection increases dramatically when distinct subpopulations are added.
T2.3.1	Increasing Diversity (not tree-like)	Fail	568; 5.4	1279; 12	1911; 18	0.44; 0.45	0.67; 0.67	True; True; True	The metric used to check if the sample is star-like is not the p-value from the goodness of fit test.
T2.4.1	Mimics Illumina MiSEQ data	Pass	987; 9.4	1916; 18	2643; 25	0.52; 0.52	0.72; 0.72	False; False; False	Poisson Fitter is able to process large MiSeq-like datasets in a timely manner.
T2.5.1	Mimics SGA data	Fail	827; 197	1688; 402	2593; 618	0.49; 0.49	0.65; 0.65	False; False; False	The metric used to check if the sample is star-like is not the p-value from the goodness of fit test.
T3.1.1	Mimics Patient Sample	Fail	0; 0	4; 0.048	3830; 43	0; 0	0.001; 0.001	True; True; False	If all sequences are identical, Poisson Fitter can't process the dataset and then returns a dummy result and issues a warning, causing some of the checks to fail on the low diversity dataset.

T3.2.1	Mimics Patient Sample	Fail	0; 0	2844; 32	4048; 45	0; 0	0.70; 0.71	True; False; False	If all sequences are identical, Poisson Fitter can't process the dataset and then returns a dummy result and issues a warning, causing some of the checks to fail on the low diversity dataset.
--------	-----------------------	------	---------	-------------	-------------	---------	---------------	--------------------------	---



5.5.4 Future work and Conclusions (Poisson Fitter)

Poisson Fitter is a well-known tool for timing HIV infections and it has the attractive property of side-stepping the ongoing debate regarding the validity of phylogenetic trees for HIV-1 given the high rates of recombination. Here we designed a suite of tests for our incorporation of Poisson Fitter into a computational pipeline. Currently, development of more sophisticated implementations are in progress that will allow us to remove some of Poisson Fitter's unrealistic assumptions. The goal is to incorporate that implementation into the pipeline before the analysis of the AMP trials. This suite of tests will form the basis of demonstrating that the new implementation is correct and that it matches the results of Poisson Fitter when Poisson Fitter's assumptions are not violated.

The tests revealed a number of problems in the current implementation. The edge case in which all sequences are identical is not properly handled. While this has heretofore been incredibly unlikely to happen in real world data, however it might have to be considered as possible in the context of the AMP trial. If a subject gets infected while there is a high concentration of VRC01 in the patient's bloodstream, replication may be severely inhibited possibly suppressing viral load to the point where only one unique template gets recovered from the patient.

There is a mismatch between the Poisson Fitter software's assignments of datasets as star-like or non-star-like and the p-values reported. This is probably a strategy to compensate for the over sensitivity of the Chi-square goodness of fit test for large samples. Since each individual pairwise distance is treated as an observation, the effective sample size is large even for a moderate number of sequences. In the presence of gaps or ambiguity letters, the computation of the confidence intervals for the estimated time since infection fails. This is a limitation that must be corrected. Gaps and ambiguity characters have no effect on the point estimates produced by Poisson Fitter, this is short-coming of the technique used to prepare the distance matrices for Poisson Fitter and it should be corrected in the `runPoissonFitter.R` script.

The estimates produced by Poisson Fitter are highly sensitive to the presence of distinct sub-populations in the datasets. This is expected in cases where the infection was founded by multiple distinct founders. The pipeline already includes a step that clusters the dataset into sub-clusters and then runs Poisson Fitter independently on each of the sub-clusters of the dataset. The separate estimated times since infection are obtained for each sub-cluster is then averaged to obtain the overall estimate for the sample. Future effort should extend the test procedure to cover this case. Careful attention must be paid to cases in which the distinct founders are closely related enough to make it hard to accurately cluster the sample into sub-clusters.

5.6 Future work and Conclusions (Infection Timing)

Accurately timing infections and reconstructing the founding viruses will be important aspects of a major ongoing clinical trial. This chapter presented a set of tests and investigations of a portion of a pipeline that aims to solve the timing aspect of this problem. hypermutR performed well in the testing scenarios with only a single discrepancy arising. This discrepancy involves an obscure case in which a gap is inserted into the potentially hypermutated position and reveals an internal inconsistency in the implementation of the algorithm on the LANL website. The subsequent three sections of this chapter all involve computing a diversity measure, whether that is an aggregate of positional entropy, an average pairwise phylogenetic distance based on the assumption of an underlying tree-like structure in the dataset or a complex remapping of a pairwise Hamming distance matrix into an estimate of the time since infection. While the entropy calculation presented is well implemented with handling for IUPAC ambiguity characters and gaps, it assumes that positions are independent of each other. PhyML produces a measure that is tightly correlated with the much simpler average pairwise Hamming distance while being dependent on the order of the sequences in the dataset – especially in the case when the data is not based on a tree-like data structure. This tight correlation with such a simple measure combined with the shortcoming of the estimates produced by PhyML motivates our suggestion of removing PhyML from the pipeline. The Poisson Fitter implementation's inability to handle gaps or ambiguity characters result from the chosen method to produce the distance matrix on which it operates and can be easily addressed. The Poisson Fitter algorithm was shown to be highly sensitive to datasets that have multiple subpopulations, such as those found in patients who were infected multiple times with distinct variants of HIV. PhyML fits a complex model that specifies the shape of a tree and the length of the branches in the tree to a dataset. This involves the estimation of a large set of parameters. When generating datasets to test a software implementation, an emphasis must be placed on small simple datasets that will produce predictable results. However, such datasets do not contain much information that can be used to estimate the parameters of the model, leading to unstable results. In the case of the test of extremely heterogeneous datasets, PhyML was unable to detect any differences between three datasets that were specifically constructed to have different underlying structures. The order of the sequences in the dataset was shown to have an effect of the estimates produced by PhyML with the variability introduced by sequence order being 4 times higher on datasets that have a star-like phylogeny than on datasets with a tree-like phylogeny. PhyML treats gaps as missing data and does not account for the additional information present in ambiguity letters. The Poisson Fitter algorithm operates on a distance matrix and not directly on a sequence dataset. Thus the approach to computing the distance matrix has a large effect on the results produced by Poisson Fitter. The approach used in this pipeline does not handle gaps or ambiguity letters and this

should be corrected. In cases where all sequences are identical, the algorithm cannot be run and sensible default results needs to be returned. Poisson Fitter compares the distribution of the frequencies of pairwise Hamming distances between the observed data and what it expects under a star-like phylogeny. The software includes a Chi-square goodness of fit test, but the determination of model fit reported by the software is based on an informal check. The results of the informal check aligns with the expectations from our test datasets while the results from the Chi-square goodness of fit test does not. This is potentially due to the large numbers of observations causing the goodness of fit test to be over sensitive. These discrepancies should be formally investigated.

The presence of two (three) distinct subpopulations within a dataset increased the estimated time since infection by a factor of four (seven). Datasets should be carefully screened for multiple infection and such cases should be handled separately. Functionality, which was not reviewed in this thesis, exists within the pipeline that clusters the sequences in a dataset into distinct sets and then applies the Poisson Fitter algorithm on each such set independently. The estimates from each set is then averaged to form a global estimated time since infection. In cases where multiple infection occurred and the variants were closely related to each other, detection of the multiple infection event can be hard. Simulation studies must be performed to explore the limits where multiple infection cannot be detected and must quantify the bias expected in such cases. Extensive recombination between the subpopulations may also render multiple infection undetectable. Simulation studies should also explore which levels of recombination render the multiplicity of infection undetectable and the effect this has on the timing estimates.

While the overall test results for the software were promising, with the algorithms being able to detect increases and decreases in diversity, the AMP trial introduces a major unknown into the equation. The subjects on the active arm will have the VRC01 antibody circulating in their bodies. Since VRC01 is a broadly neutralizing antibody, it is expected to target the envelope gene of many variants and to cause these variants to be removed from the population. This violates the key assumption of the Poisson Fitter algorithm that requires the population to grow exponentially at a constant rate. Furthermore, if a substantial portion of the viral population dies off in each generation, the parameters of the equation that transform the distance matrix into an estimated time since infection may no longer be valid.

6 Future work and conclusions

Deep sequencing of diverse populations is a useful technique, allowing one to investigate complex phenomena like the presence and frequency of drug resistant variants (Jabara et al., 2011), as well as measuring the diversity of microbial species in various biological samples (Lundberg et al., 2013). However, when sequencing a population with a high level of diversity, differentiating between sequencing error and real variation becomes problematic. The result of this is that the cutoff frequency for detecting low abundance polymorphisms is limited by the level of sequencing error rate. Additionally, skewed amplification of certain templates during PCR confounds the measurement of the frequency distribution of different variants.

The PID approach (Jabara et al., 2011) has been proposed to address these problems. By tagging each input template with a unique nucleotide sequence (PID), all amplicons derived from a specific template should share the same PID tag. The resampling of the initial template can be used to correct for non-systematic sequencing errors, as taking the consensus base at each position should correct for these errors. At the same time, the PCR skewing problem is also addressed, since collapsing all sequences with the same PID into one sequence, corrects for the skewed amplification of certain templates.

We developed MotifBinner to deal with the complications associated with PID data. MotifBinner identifies the PIDs added to the sequences and uses them to group the reads into bins. It then constructs a consensus sequence for each such bin and outputs these sequences which are high quality representations of the original input templates. During this process, MotifBinner uses fuzzy matching to ensure that minor mutation in the primers do not prevent the identification of the PID, discards bins that are likely to be the result of sequencing error in the PID itself, flags bins as chimeric if there are positions with too many mismatches in the sequences of the bin, inspects bins to find sequences that were erroneously assigned to the bin and remove them and then aligns the sequences in the bins. MotifBinner constructs extensive reports that aid in troubleshooting the sample preparation process in the case that poor quality data was produced.

An extensive optimization study was conducted to test the effect of various parameters on the data quality. We explored using droplet PCR, longer elongation times, and different numbers of PCR cycles. All approaches drastically reduced the sequence error rate in the consensus sequences. The error rates observed in the consensus sequences were similar to the error rates of the reverse transcriptase enzyme implying that all other sources of error were reduced to near zero. We found that estimates for prevalences of different variants in a sample were much lower variance when using estimators based on the consensus sequences than when using unprocessed sequences. A key concern was the

rate at which chimeric sequences were produced as a result of recombination during PCR. With the exception of the protocols that utilized droplet PCR during the second round of PCR, virtually no chimeric reads were found. Out of a total of 172 544 consensus sequences, only 25 were flagged as chimeric in the samples that did not use droplet PCR in the second round. In the unprocessed sequences, we clearly demonstrated that sample composition (the relative prevalence of the different variants) strongly influence the rate of chimerism and that recombination occurs at higher rates in the regions where the variants are very similar to each other.

While the relative performance of the different protocols were similar, we were able to use them to produce guidelines that will ensure that an optimal dataset is produced given the requirements of the study. If the goal is to detect as many minority variants as possible, then two rounds of normal PCR should be used. The number of cycles of PCR should be carefully optimized to ensure that the reaction is stopped prior to saturation or depletion of reagents. If the goal is to recover sequences with the explicit goal of minimizing the probability that any sequence is the result of PCR induced recombination, then the mxPCR protocol may be considered which uses dPCR for the first round and normal PCR for the second round. Again, optimizing the PCR reaction to ensure that they stop before saturation or depletion of the reagents will decrease error and recombination rates. This approach reduces PCR recombination, but yield is suppressed and the rate of non-recombination errors are increased. Lastly, if the goal is to search for extremely rare (1% or below) single nucleotide polymorphisms (SNP) at known positions, then a re-analysis of the data, scanning for bins in which that specific SNP is present at levels higher than what is expected due to sequencing error is a method to improve detection. However, careful exploration of the false positive rate will be required when using this technique. Lastly, increasing the elongation times to 10min yields little improvement, suggesting that for most cases using 2.5min elongation times will yield adequate results. [A potential approach that can decrease recombination even further is to explore the use of less stable emulsions during the dPCR step to increase the efficiency with which the PCR product can be extracted.](#)

Accurately timing infections and reconstructing the founding viruses will be important aspects of a major ongoing clinical trial. The final chapter of this thesis presented a set of tests and investigations of a portion of a pipeline that aims to solve the timing aspect of this problem. The first step in the pipeline is the removal hypermutated sequences. We re-implemented the hypermut 2.0 algorithm. We demonstrated that our implementation faithfully reproduces the results of the implementation available from LANL with the only deviation resulting from carefully investigated edge case. We were able to produce a compelling argument for retaining the deviation from the LANL implementation since it ensures that the algorithm handles hypermutation and control positions consistently.

All the subsequent steps in the timing pipeline involve computing a diversity measure. The first diversity measure considered is the Shannon entropy computed for each position and then summed across positions. This entropy calculation is well implemented and is the only diversity measure that correctly handles IUPAC ambiguity characters, but it assumes that positions are independent of each other. The other two diversity measures do not rely on the assumption that the positions are independent. The second diversity measure is based on an average phylogenetic distance computed from fitting a tree-like model to the dataset and the last one relies on the average pairwise Hamming distance between the sequences.

PhyML produces a measure that is tightly correlated with the much simpler average pairwise Hamming distance while being dependent on the order of the sequences in the dataset – especially in the case when the data is not based on a tree-like data structure. This tight correlation with such a simple measure combined with the shortcoming of the estimates produced by PhyML motivates our suggestion of removing PhyML from the pipeline. Additionally, PhyML fits a complex model that specifies the shape of a tree and the length of the branches in the tree to a dataset. Since the goal of the pipeline is to process data from samples taken soon after infection the data is unlikely to have such a complex structure. As a result, the complex PhyML model will fit the noise in the dataset and produce underestimates of the diversity. While this is a minor effect, misspecification of a model should be avoided when possible. The next step is to remove PhyML from the pipeline and to conduct some tests to confirm that the conclusions based on a simpler distance metric is comparable to those based on PhyML's distance metric.

The Poisson Fitter algorithm operates on a Hamming distance matrix and not directly on a sequence dataset. Thus the approach to computing the distance matrix has a large effect on the results produced by Poisson Fitter. The approach used in this pipeline does not handle gaps or ambiguity letters and this should be corrected. Poisson Fitter compares the distribution of the frequencies of pairwise Hamming distances between the observed data and what it expects under a star-like phylogeny. The software includes a Chi-square goodness of fit test, but the determination of model fit reported by the software is based on an informal check. The results of the informal check aligns with the expectations from our test datasets and is reported by the pipeline in the final dataset.

The presence of two (three) distinct subpopulations within a dataset erroneously increased the estimated time since infection by a factor of four (seven). Datasets should be carefully screened for multiple infection and such cases should be handled separately. Functionality, which was not reviewed in this thesis, exists within the pipeline that clusters the sequences in a dataset into distinct sets and then applies the Poisson Fitter algorithm on each such set independently. The estimates from each

set is then averaged to form a global estimated time since infection. In cases where multiple infection occurred and the variants were closely related to each other, detection of the multiple infection event can be hard. Simulation studies must be performed to explore the limits where multiple infection cannot be detected and must quantify the bias expected in such cases. Extensive recombination between the subpopulations may also render multiple infection undetectable. Simulation studies should also explore which levels of recombination render the multiplicity of infection undetectable and the effect this has on the timing estimates.

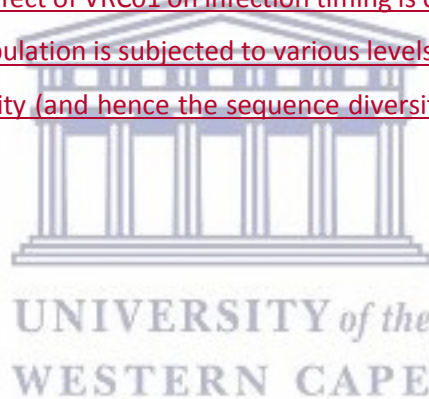
The work to formalize and test the pipeline will continue. Large portions of the pipeline remain to be evaluated such as the modifications to PFitter that hides the nonsynonymous mutations and the clusters the sequences for multifounder samples as well as those parts that characterize the samples as multifounder. Work done in parallel to this thesis, but not described here, fitted the estimates produced by the pipeline to existing datasets with well characterized infection times (RV217 and CAPRISA002). This fitting procedure produces a linear regression model (logistic regression in the case of the multifounder prediction) that calibrates the pipeline's outputs to existing datasets resulting in a more accurate predictor. Variables such as the diagnosis date, viral load and sequence diversity are added into these models as covariates. A key observation from this work is that the accuracy of the predictions of infection timing do not decrease when the sequence data is removed from the model.

The current hypothesis to explain this unexpected finding is that the conduct of the trials (RV217 in particular) was so rigid that the statistical models model the trial design and not the actual growth and evolution of the quasispecies. For example, in RV217, patients were screened every 3 to 7 days for infection and the sequencing occurred 6 month later. Thus, by modelling infection time as a random normal variable with a mean of 6 months and 5 days and a standard deviation proportional to the standard deviation of the timing of the sequencing event, a model with good predictive power can be constructed. To address this problem we plan to conduct simulations to test the hypothesis that the calibration procedure just modelled the trial design and to learn the conditions under which such a calibration approach might be generalizable. We also plan to gather extra datasets so that the calibration models are trained on larger and more diverse datasets with different follow-up and screening protocols. Such an increase in the diversity of the trial designs from which the training dataset was generated should reduce the predictive power of models that capture the trial design instead of the growth and evolution characteristics of the quasispecies.

While the overall test results for the software were promising, with the algorithms being able to detect increases and decreases in diversity, the AMP trial introduces a major unknown into the equation. The subjects on the active arm will have the VRC01 antibody circulating in their bodies. Since VRC01 is a

broadly neutralizing antibody, it is expected to target the envelope gene of many variants and to cause these variants to be removed from the population. This violates the key assumption of the Poisson Fitter algorithm that requires the population to grow exponentially at a constant rate. Furthermore, if a substantial portion of the viral population dies off in each generation, the parameters of the equation that transform the distance matrix into an estimated time since infection may no longer be valid.

To explore the effect of VRC01 on the accuracy of HIV infection timing requires significant effort and financial resources. Ideally, primate studies can be conducted in which the growth and evolution can be observed in a controlled environment. Another possible avenue is to find data where people who were on pre-exposure prophylaxis (PrEP) who still became infected. Such data may shed light on the effect of agents that inhibit the replication of HIV on the growth and evolution of the quasispecies. However, given the paucity of data of adequate quality to calibrate infection timing methodologies in individuals who were on PrEP, it is highly unlikely that a suitable dataset can be constructed. Hence, the next step for exploring the effect of VRC01 on infection timing is currently restricted to simulation studies where the simulated population is subjected to various levels of selective pressure to explore the result this has on the diversity (and hence the sequence diversity based infection timing) in the resulting quasispecies.



7 Bibliography

- Abrahams, M.-R., Anderson, J. A., Giorgi, E. E., Seoghe, C., Mlisana, K., Ping, L.-H., ... Williamson, C. (2009). Quantitating the multiplicity of infection with human immunodeficiency virus type 1 subtype C reveals a non-poisson distribution of transmitted variants. *Journal of Virology*, 83(8), 3556–67. <https://doi.org/10.1128/JVI.02132-08>
- Andrake, M. D., & Skalka, A. M. (1996). Retroviral integrase, putting the pieces together. *The Journal of Biological Chemistry*, 271(33), 19633–6. Retrieved from <http://www.ncbi.nlm.nih.gov/pubmed/8702660>
- Bebenek, K., Abbotts, J., Wilson, S. H., & Kunkel, T. A. (1993). Error-prone polymerization by HIV-1 reverse transcriptase. Contribution of template-primer misalignment, miscoding, and termination probability to mutational hot spots. *The Journal of Biological Chemistry*, 268(14), 10324–34. Retrieved from <http://www.ncbi.nlm.nih.gov/pubmed/7683675>
- Behrens, A. (2017). Fine structure of the HIV-1 glycan shield. Retrieved from <https://ora.ox.ac.uk/objects/uuid:3cec0ef7-c305-411e-a76b-125d5e7e9954>
- Bhiman, J. N., Anthony, C., Doria-Rose, N. A., Karimanzira, O., Schramm, C. A., Khoza, T., ... Moore, P. L. (2015). Viral variants that initiate and drive maturation of V1V2-directed HIV-1 broadly neutralizing antibodies. *Nature Medicine, advance on*. <https://doi.org/10.1038/nm.3963>
- Boltz, V. F., Rausch, J., Shao, W., Hattori, J., Luke, B., Maldarelli, F., ... Coffin, J. M. (2016). Ultrasensitive single-genome sequencing: accurate, targeted, next generation sequencing of HIV-1 RNA. *Retrovirology*, 13(1), 87. <https://doi.org/10.1186/s12977-016-0321-6>
- Brodin, J., Hedskog, C., Heddini, A., Benard, E., Neher, R. A., Mild, M., & Albert, J. (2015). Challenges with using primer IDs to improve accuracy of next generation sequencing. *PloS One*, 10(3), e0119123. <https://doi.org/10.1371/journal.pone.0119123>
- Butler, D. M., Pacold, M. E., Jordan, P. S., Richman, D. D., & Smith, D. M. (2009). The efficiency of single genome amplification and sequencing is improved by quantitation and use of a bioinformatics tool. *Journal of Virological Methods*, 162(1–2), 280–283. <https://doi.org/10.1016/j.jviromet.2009.08.002>
- Casbon, J. A., Osborne, R. J., Brenner, S., & Lichtenstein, C. P. (2011). A method for counting PCR template molecules with application to next-generation sequencing. *Nucleic Acids Research*, 39(12), e81–e81. <https://doi.org/10.1093/nar/gkr217>
- Charif, D., & Lobry, J. R. (2007). SeqinR 1.0-2: A Contributed Package to the R Project for Statistical Computing Devoted to Biological Sequences Retrieval and Analysis. In U. Bastolla, M. Porto, E. Roman, & M. Vendruscolo (Eds.), *Structural Approaches to*

Sequence Evolution (pp. 207–232). Springer-Verlag Berlin Heidelberg.
https://doi.org/10.1007/978-3-540-35306-5_10

- Chun, T. W., Stuyver, L., Mizell, S. B., Ehler, L. A., Mican, J. A., Baseler, M., ... Fauci, A. S. (1997). Presence of an inducible HIV-1 latent reservoir during highly active antiretroviral therapy. *Proceedings of the National Academy of Sciences of the United States of America*, 94(24), 13193–7. Retrieved from <http://www.ncbi.nlm.nih.gov/pubmed/9371822>
- Crick, F. (1970). Central dogma of molecular biology. *Nature*, 227(5258), 561–3. Retrieved from <http://www.ncbi.nlm.nih.gov/pubmed/4913914>
- Crick, F. H. (1958). On protein synthesis. *Symposia of the Society for Experimental Biology*, 12, 138–63. Retrieved from <http://www.ncbi.nlm.nih.gov/pubmed/13580867>
- Davis, T. L. (2017). optparse: Command Line Option Parser. Retrieved from <https://cran.r-project.org/package=optparse>
- Doms, R. W., & Trono, D. (2000). The plasma membrane as a combat zone in the HIV battlefield. *Genes & Development*, 14(21), 2677–88. Retrieved from <http://www.ncbi.nlm.nih.gov/pubmed/11069884>
- Dovichi, N. J., & Zhang, J. (2000). How capillary electrophoresis sequenced the human genome. *Angewandte Chemie International Edition*, 39(24), 4463–4468.
- Edgar, R. C. (2004). MUSCLE: multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Research*, 32(5), 1792–7. <https://doi.org/10.1093/nar/gkh340>
- Fairbanks, D. J., & Andersen, W. R. (1999). *Genetics*. Brooks/Cole Publishing Company.
- Fu, G. K., Hu, J., Wang, P.-H., & Fodor, S. P. A. (2011). Counting individual DNA molecules by the stochastic attachment of diverse labels. *Proceedings of the National Academy of Sciences of the United States of America*, 108(22), 9026–31. <https://doi.org/10.1073/pnas.1017621108>
- Fu, G. K., Xu, W., Wilhelmy, J., Mindrinos, M. N., Davis, R. W., Xiao, W., & Fodor, S. P. A. (2014). Molecular indexing enables quantitative targeted RNA sequencing and reveals poor efficiencies in standard library preparations. *Proceedings of the National Academy of Sciences of the United States of America*, 111(5), 1891–6. <https://doi.org/10.1073/pnas.1323732111>
- Gao, F., Chen, Y., Levy, D. N., Conway, J. A., Kepler, T. B., & Hui, H. (2004). Unselected mutations in the human immunodeficiency virus type 1 genome are mostly nonsynonymous and often deleterious. *Journal of Virology*, 78(5), 2426–33. Retrieved from <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=369203&tool=pmcentrez&rendertype=abstract>
- Gilbert, P. B., Juraska, M., deCamp, A. C., Karuna, S., Edupuganti, S., Mgodhi, N., ... Corey, L. (2017). Basis and Statistical Design of the Passive HIV-1 Antibody Mediated

- Prevention (AMP) Test-of-Concept Efficacy Trials. *Statistical Communications in Infectious Diseases*, 9(1). <https://doi.org/10.1515/scid-2016-0001>
- Giorgi, E. E., & Bhattacharya, T. (2012). A note on two-sample tests for comparing intra-individual genetic sequence diversity between populations. *Biometrics*, 68(4), 1323–6; author reply 1326. <https://doi.org/10.1111/j.1541-0420.2012.01775.x>
- Giorgi, E. E., Funkhouser, B., Athreya, G., Perelson, A. S., Korber, B. T., & Bhattacharya, T. (2010). Estimating time since infection in early homogeneous HIV-1 samples using a poisson model. *BMC Bioinformatics*, 11(1), 532. <https://doi.org/10.1186/1471-2105-11-532>
- Girard, M. P., Osmanov, S., Assossou, O. M., & Kieny, M. P. (2011). Human immunodeficiency virus (HIV) immunopathogenesis and vaccine development: A review. *Vaccine*, 29(37), 6191–6218. <https://doi.org/10.1016/j.vaccine.2011.06.085>
- Gottlieb, M. S., Schroff, R., Schanker, H. M., Weisman, J. D., Fan, P. T., Wolf, R. A., & Saxon, A. (1981). Pneumocystis carinii pneumonia and mucosal candidiasis in previously healthy homosexual men: evidence of a new acquired cellular immunodeficiency. *The New England Journal of Medicine*, 305(24), 1425–31. <https://doi.org/10.1056/NEJM198112103052401>
- Gray, G. E., Allen, M., Moodie, Z., Churchyard, G., Bekker, L.-G., Nchabeleng, M., ... HVTN 503/Phambili study team. (2011). Safety and efficacy of the HVTN 503/Phambili Study of a clade-B-based HIV-1 vaccine in South Africa: a double-blind, randomised, placebo-controlled test-of-concept phase 2b study. *The Lancet Infectious Diseases*, 11(7), 507–515. [https://doi.org/10.1016/S1473-3099\(11\)70098-6](https://doi.org/10.1016/S1473-3099(11)70098-6)
- Gregori, J., Perales, C., Rodriguez-Frias, F., Esteban, J. I., Quer, J., & Domingo, E. (2016). Viral quasispecies complexity measures. *Virology*. <https://doi.org/10.1016/j.virol.2016.03.017>
- Grothendieck, G. (2013). nls2: Non-linear regression with brute force. Retrieved from <https://cran.r-project.org/package=nls2>
- Guindon, S., Dufayard, J. F., Lefort, V., Anisimova, M., Hordijk, W., & Gascuel, O. (2010). New algorithms and methods to estimate maximum-likelihood phylogenies: Assessing the performance of PhyML 3.0. *Systematic Biology*, 59(3), 307–321. <https://doi.org/10.1093/sysbio/syq010>
- Holm, S. (1979). A Simple Sequentially Rejective Multiple Test Procedure. *Scandinavian Journal of Statistics*, 6(2), 65–70. Retrieved from <http://www.jstor.org/stable/4615733>
- Hong, L. Z., Hong, S., Wong, H., Aw, P., Cheng, Y., Wilm, A., ... Burkholder, W. F. (2014). BAsE-Seq: a method for obtaining long viral haplotypes from short sequence reads. *Genome Biology*, 15(11), 517. <https://doi.org/10.1186/s13059-014-0517-9>
- Illumina. (2015). Illumina MiSeq Specifications. Retrieved from

http://www.illumina.com/content/illumina-marketing/amr/en_US/systems/miseq/performance_specifications.html

- Jabara, C. B., Jones, C. D., Roach, J., Anderson, J. A., & Swanstrom, R. (2011). Accurate sampling and deep sequencing of the HIV-1 protease gene using a Primer ID. *Proceedings of the National Academy of Sciences of the United States of America*, *108*(50), 20166–71. <https://doi.org/10.1073/pnas.1110064108>
- Kanagawa, T. (2003). Bias and Artifacts in Multitemplate Polymerase Chain Reactions(PCR). *Journal of Bioscience and Bioengineering*, *96*(4), 317–323. <https://doi.org/10.1263/jbb.96.317>
- Katoh, K., Misawa, K., Kuma, K., & Miyata, T. (2002). MAFFT: a novel method for rapid multiple sequence alignment based on fast Fourier transform. *Nucleic Acids Research*, *30*(14), 3059–66. Retrieved from <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=135756&tool=pmcentrez&rendertype=abstract>
- Keele, B. F., Giorgi, E. E., Salazar-Gonzalez, J. F., Decker, J. M., Pham, K. T., Salazar, M. G., ... Shaw, G. M. (2008). Identification and characterization of transmitted and early founder virus envelopes in primary HIV-1 infection. *Proceedings of the National Academy of Sciences*, *105*(21), 7552–7557. <https://doi.org/10.1073/pnas.0802203105>
- Kinde, I., Wu, J., Papadopoulos, N., Kinzler, K. W., & Vogelstein, B. (2011). Detection and quantification of rare mutations with massively parallel sequencing. *Proceedings of the National Academy of Sciences of the United States of America*, *108*(23), 9530–5. <https://doi.org/10.1073/pnas.1105422108>
- Kivioja, T., Vähärautio, A., Karlsson, K., Bonke, M., Enge, M., Linnarsson, S., & Taipale, J. (2012). Counting absolute numbers of molecules using unique molecular identifiers. *Nature Methods*, *9*(1), 72–4. <https://doi.org/10.1038/nmeth.1778>
- Kohl, N. E., Emini, E. A., Schleif, W. A., Davis, L. J., Heimbach, J. C., Dixon, R. A., ... Sigal, I. S. (1988). Active human immunodeficiency virus protease is required for viral infectivity. *Proceedings of the National Academy of Sciences of the United States of America*, *85*(13), 4686–90. Retrieved from <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=280500&tool=pmcentrez&rendertype=abstract>
- Kou, R., Lam, H., Duan, H., Ye, L., Jongkam, N., Chen, W., ... Li, S. (2016). Benefits and Challenges with Applying Unique Molecular Identifiers in Next Generation Sequencing to Detect Low Frequency Mutations. *PloS One*, *11*(1), e0146638. <https://doi.org/10.1371/journal.pone.0146638>
- LaFemina, R. L., Schneider, C. L., Robbins, H. L., Callahan, P. L., LeGrow, K., Roth, E., ... Emini, E. A. (1992). Requirement of active human immunodeficiency virus type 1 integrase enzyme for productive infection of human T-lymphoid cells. *Journal of*

Virology, 66(12), 7414–9. Retrieved from <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=240448&tool=pmcentrez&rendertype=abstract>

- Langmead, B., Trapnell, C., Pop, M., & Salzberg, S. L. (2009). Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biology*, 10(3), R25. <https://doi.org/10.1186/gb-2009-10-3-r25>
- Lee, H. Y., Giorgi, E. E., Keele, B. F., Gaschen, B., Athreya, G. S., Salazar-Gonzalez, J. F., ... Perelson, A. S. (2009). Modeling sequence evolution in acute HIV-1 infection. *Journal of Theoretical Biology*, 261(2), 341–60. <https://doi.org/10.1016/j.jtbi.2009.07.038>
- Li, H., & Durbin, R. (2009). Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics (Oxford, England)*, 25(14), 1754–60. <https://doi.org/10.1093/bioinformatics/btp324>
- Liu, J., Song, H., Liu, D., Zuo, T., Lu, F., Zhuang, H., & Gao, F. (2014). Extensive recombination due to heteroduplexes generates large amounts of artificial gene fragments during pcr. *PLoS ONE*, 9(9), e106658. <https://doi.org/10.1371/journal.pone.0106658>
- Liu, L., Li, Y., Li, S., Hu, N., He, Y., Pong, R., ... Law, M. (2012). Comparison of next-generation sequencing systems. *Journal of Biomedicine & Biotechnology*, 2012, 251364. <https://doi.org/10.1155/2012/251364>
- Lundberg, D. S., Yourstone, S., Mieczkowski, P., Jones, C. D., & Dangl, J. L. (2013). Practical innovations for high-throughput amplicon sequencing. *Nature Methods*, 10(10), 999–1002. <https://doi.org/10.1038/nmeth.2634>
- Mansky, L. M., & Temin, H. M. (1995). Lower in vivo mutation rate of human immunodeficiency virus type 1 than that predicted from the fidelity of purified reverse transcriptase. *Journal of Virology*, 69(8), 5087–94. Retrieved from <http://www.ncbi.nlm.nih.gov/pubmed/7541846>
- Masur, H., Michelis, M. A., Greene, J. B., Onorato, I., Stouwe, R. A., Holzman, R. S., ... Cunningham-Rundles, S. (1981). An outbreak of community-acquired Pneumocystis carinii pneumonia: initial manifestation of cellular immune dysfunction. *The New England Journal of Medicine*, 305(24), 1431–8. <https://doi.org/10.1056/NEJM198112103052402>
- McMichael, A. J., & Koff, W. C. (2014). Vaccines that stimulate T cell immunity to HIV-1: the next step. *Nature Immunology*, 15(4), 319–322. <https://doi.org/10.1038/ni.2844>
- Miller, M. D., Farnet, C. M., & Bushman, F. D. (1997). Human immunodeficiency virus type 1 preintegration complexes: studies of organization and composition. *Journal of Virology*, 71(7), 5382–90. Retrieved from <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=191777&tool=pmcentrez&rendertype=abstract>

- Miller, R. G. J. (2012). *Simultaneous Statistical Inference*. Springer Science & Business Media. Retrieved from <https://books.google.com/books?hl=en&lr=&id=4C7VBwAAQBAJ&pgis=1>
- Morgan, M., Anders, S., Lawrence, M., Aboyoun, P., Pagès, H., & Gentleman, R. (2009). {ShortRead}: a {B}ioconductor package for input, quality assessment and exploration of high-throughput sequence data. *Bioinformatics*, 25, 2607–2608. <https://doi.org/10.1093/bioinformatics/btp450>
- Nachega, J. B., Marconi, V. C., van Zyl, G. U., Gardner, E. M., Preiser, W., Hong, S. Y., ... Gross, R. (2011). HIV treatment adherence, drug resistance, virologic failure: evolving concepts. *Infectious Disorders Drug Targets*, 11(2), 167–74. Retrieved from <http://www.ncbi.nlm.nih.gov/pubmed/21406048>
- Nowak, M. A. (1992). What is a quasispecies? *Trends in Ecology & Evolution*, 7(4), 118–21. [https://doi.org/10.1016/0169-5347\(92\)90145-2](https://doi.org/10.1016/0169-5347(92)90145-2)
- Ochsenbauer, C., Edmonds, T. G., Ding, H., Keele, B. F., Decker, J., Salazar, M. G., ... Kappes, J. C. (2012). Generation of Transmitted/Founder HIV-1 Infectious Molecular Clones and Characterization of Their Replication Capacity in CD4 T Lymphocytes and Monocyte-Derived Macrophages. *Journal of Virology*, 86(5), 2715–2728. <https://doi.org/10.1128/JVI.06157-11>
- OpenStax-College. (2015). *Anatomy & Physiology*. OpenStax CNX. Retrieved from <http://cnx.org/contents/14fb4ad7-39a1-4eee-ab6e-3ef2482e3e22@7.30>
- Ouzounis, C. A., & Valencia, A. (2003). Early bioinformatics: the birth of a discipline--a personal view. *Bioinformatics (Oxford, England)*, 19(17), 2176–90. Retrieved from <http://www.ncbi.nlm.nih.gov/pubmed/14630646>
- Pagès, H., Aboyoun, P., Gentleman, R., & DebRoy, S. (2017). Biostrings: String objects representing biological sequences, and matching algorithms. Retrieved from <http://bioconductor.org/packages/release/bioc/html/Biostrings.html>
- Pau, A. K., & George, J. M. (2014). Antiretroviral therapy: current drugs. *Infectious Disease Clinics of North America*, 28(3), 371–402. <https://doi.org/10.1016/j.idc.2014.06.001>
- Perelson, A. S., Neumann, A. U., Markowitz, M., Leonard, J. M., & Ho, D. D. (1996). HIV-1 dynamics in vivo: virion clearance rate, infected cell life-span, and viral generation time. *Science (New York, N.Y.)*, 271(5255), 1582–6. Retrieved from <http://www.ncbi.nlm.nih.gov/pubmed/8599114>
- Polz, M. F., & Cavanaugh, C. M. (1998). Bias in template-to-product ratios in multitemplate PCR. *Applied and Environmental Microbiology*, 64(10), 3724–30. Retrieved from <http://www.ncbi.nlm.nih.gov/pubmed/9758791>
- Price, M. N., Dehal, P. S., & Arkin, A. P. (2009). FastTree: computing large minimum evolution trees with profiles instead of a distance matrix. *Molecular Biology and*

Evolution, 26(7), 1641–50. <https://doi.org/10.1093/molbev/msp077>

- Rambaut, A., Posada, D., Crandall, K. A., & Holmes, E. C. (2004). The causes and consequences of HIV evolution. *Nature Reviews. Genetics*, 5(1), 52–61. <https://doi.org/10.1038/nrg1246>
- Ranwez, V., Harispe, S., Delsuc, F., & Douzery, E. J. P. (2011). MACSE: Multiple Alignment of Coding SEquences Accounting for Frameshifts and Stop Codons. *PLoS One*, 6(9), e22594. <https://doi.org/10.1371/journal.pone.0022594>
- Revolution Analytics, & Weston, S. (2015). foreach: Provides Foreach Looping Construct for R. Retrieved from <https://cran.r-project.org/package=foreach>
- Rigby, R. A., & Stasinopoulos, D. M. (2005). Generalized additive models for location, scale and shape (with discussion). *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 54(3), 507–554. <https://doi.org/10.1111/j.1467-9876.2005.00510.x>
- Rose, P. P., & Korber, B. T. (2000). Detecting hypermutations in viral sequences with an emphasis on G-to-A hypermutation. *Bioinformatics (Oxford, England)*, 16(4), 400–1. [https://doi.org/Doi 10.1093/Bioinformatics/16.4.400](https://doi.org/Doi%2010.1093/Bioinformatics/16.4.400)
- Saiki, R., Gelfand, D., Stoffel, S., Scharf, S., Higuchi, R., Horn, G., ... Erlich, H. (1988). Primer-directed enzymatic amplification of DNA with a thermostable DNA polymerase. *Science*, 239(4839), 487–491. <https://doi.org/10.1126/science.2448875>
- Saiki, R. K., Scharf, S., Faloona, F., Mullis, K. B., Horn, G. T., Erlich, H. A., & Arnheim, N. (1985). Enzymatic amplification of beta-globin genomic sequences and restriction site analysis for diagnosis of sickle cell anemia. *Science (New York, N.Y.)*, 230(4732), 1350–4. Retrieved from <http://www.ncbi.nlm.nih.gov/pubmed/2999980>
- Sanger, F., Nicklen, S., & Coulson, A. R. (1977). DNA sequencing with chain-terminating inhibitors. *Proceedings of the National Academy of Sciences of the United States of America*, 74(12), 5463–7. Retrieved from <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=431765&tool=pmcentrez&endertype=abstract>
- Schierup, M. H., & Hein, J. (2000). Recombination and the Molecular Clock. *Molecular Biology and Evolution*, 17(10), 1578–1579. <https://doi.org/10.1093/oxfordjournals.molbev.a026256>
- Schirmer, M., Ijaz, U. Z., D'Amore, R., Hall, N., Sloan, W. T., & Quince, C. (2015). Insight into biases and sequencing errors for amplicon sequencing with the Illumina MiSeq platform. *Nucleic Acids Research*, gku1341-. <https://doi.org/10.1093/nar/gku1341>
- Shao, W., Boltz, V. F., Spindler, J. E., Kearney, M. F., Maldarelli, F., Mellors, J. W., ... Coffin, J. M. (2013). Analysis of 454 sequencing error rate, error sources, and artifact recombination for detection of Low-frequency drug resistance mutations in HIV-1 DNA. *Retrovirology*, 10, 18. <https://doi.org/10.1186/1742-4690-10-18>

- Sheward, D. J., Murrell, B., & Williamson, C. (2012). Degenerate Primer IDs and the birthday problem. *Proceedings of the National Academy of Sciences of the United States of America*, *109*(21), E1330; author reply E1331. <https://doi.org/10.1073/pnas.1203613109>
- Shiroguchi, K., Jia, T. Z., Sims, P. A., & Xie, X. S. (2012). Digital RNA sequencing minimizes sequence-dependent bias and amplification noise with optimized single-molecule barcodes. *Proceedings of the National Academy of Sciences of the United States of America*, *109*(4), 1347–52. <https://doi.org/10.1073/pnas.1118018109>
- Simon, V., & Ho, D. D. (2003). HIV-1 dynamics in vivo: implications for therapy. *Nature Reviews Microbiology*, *1*(3), 181–190. <https://doi.org/10.1038/nrmicro772>
- Thomas, R. K., Nickerson, E., Simons, J. F., Jänne, P. A., Tengs, T., Yuza, Y., ... Meyerson, M. (2006). Sensitive mutation detection in heterogeneous cancer specimens by massively parallel picoliter reactor sequencing. *Nature Medicine*, *12*(7), 852–5. <https://doi.org/10.1038/nm1437>
- Thompson, J. R., Marcelino, L. A., & Polz, M. F. (2002). Heteroduplexes in mixed-template amplifications: formation, consequence and elimination by “reconditioning PCR”. *Nucleic Acids Research*, *30*(9), 2083–2088. <https://doi.org/10.1093/nar/30.9.2083>
- van der Loo, M. (2014). {stringdist}: an {R} Package for Approximate String Matching. *The R Journal*, *6*(1), 111–122.
- Voelkerding, K. V, Dames, S. A., & Durtschi, J. D. (2009). Next-generation sequencing: from basic research to diagnostics. *Clinical Chemistry*, *55*(4), 641–58. <https://doi.org/10.1373/clinchem.2008.112789>
- Wagner, A., Blackstone, N., Cartwright, P., Dick, M., Misof, B., Snow, P., ... Pendleton, J. (1994). Surveys of Gene Families Using Polymerase Chain Reaction: PCR Selection and PCR Drift. *Systematic Biology*, *43*(2), 250. <https://doi.org/10.2307/2413465>
- Watson, J. D., & Crick, F. H. (1953). Molecular structure of nucleic acids; a structure for deoxyribose nucleic acid. *Nature*, *171*(4356), 737–8. Retrieved from <http://www.ncbi.nlm.nih.gov/pubmed/13054692>
- Wickham, H. (2011). testthat: Get Started with Testing. *The R Journal*, *3*, 5–10. Retrieved from https://journal.r-project.org/archive/2011-1/RJournal_2011-1_Wickham.pdf
- Yourstone, S. M., Lundberg, D. S., Dangl, J. L., & Jones, C. D. (2014). MT-Toolbox: improved amplicon sequencing using molecule tags. *BMC Bioinformatics*, *15*(1), 284. <https://doi.org/10.1186/1471-2105-15-284>
- Yu G, Smith D, Zhu H, G. Y. and L. T. (n.d.). (submitted) ggtree: an R package for visualization and annotation of phylogenetic tree with different types of meta-data. *Methods in Ecology and Evolution*.
- Zagordi, O., Bhattacharya, A., Eriksson, N., & Beerenwinkel, N. (2011). ShoRAH: estimating the genetic diversity of a mixed sample from next-generation sequencing data. *BMC*

Bioinformatics, 12(1), 119. <https://doi.org/10.1186/1471-2105-12-119>

- Zanini, F., Brodin, J., Thebo, L., Lanz, C., Bratt, G., Albert, J., & Neher, R. A. (2015). Population genomics of inpatient HIV-1 evolution. *Populations and Evolution; Genomics*.
- Zhang, J., Kobert, K., Flouri, T., & Stamatakis, A. (2014). PEAR: a fast and accurate Illumina Paired-End reAd mergeR. *Bioinformatics (Oxford, England)*, 30(5), 614–20. <https://doi.org/10.1093/bioinformatics/btt593>
- Zhang, L., Gilbert, P. B., Capparelli, E., & Huang, Y. (2018). Pharmacokinetics Simulations for Studying Correlates of Prevention Efficacy of Passive HIV-1 Antibody Prophylaxis in the Antibody Mediated Prevention (AMP) Study. Retrieved from <http://arxiv.org/abs/1801.08626>
- Zhou, Q., Chen, D., Pierstorff, E., & Luo, K. (1998). Transcription elongation factor P-TEFb mediates Tat activation of HIV-1 transcription at multiple stages. *The EMBO Journal*, 17(13), 3681–91. <https://doi.org/10.1093/emboj/17.13.3681>
- Zhou, Q., & Sharp, P. A. (1995). Novel mechanism and factor for regulation by HIV-1 Tat. *The EMBO Journal*, 14(2), 321–8. Retrieved from <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=398086&tool=pmcentrez&rendertype=abstract>
- Zhou, S., Jones, C., Mieczkowski, P., & Swanstrom, R. (2015). Primer ID Validates Template Sampling Depth and Greatly Reduces the Error Rate of Next Generation Sequencing of HIV-1 Genomic RNA Populations. *Journal of Virology*, 89(16), 8540–55. <https://doi.org/10.1128/JVI.00522-15>
- Zhou, T., Georgiev, I., Wu, X., Yang, Z.-Y., Dai, K., Finzi, A., ... Kwong, P. D. (2010). Structural basis for broad and potent neutralization of HIV-1 by antibody VRC01. *Science (New York, N.Y.)*, 329(5993), 811–7. <https://doi.org/10.1126/science.1192819>
- Abrahams, M.-R., Anderson, J. A., Giorgi, E. E., Seighe, C., Mlisana, K., Ping, L.-H., ... Williamson, C. (2009). Quantitating the multiplicity of infection with human immunodeficiency virus type 1 subtype C reveals a non-poisson distribution of transmitted variants. *Journal of Virology*, 83(8), 3556–67. <http://doi.org/10.1128/JVI.02132-08>
- Andrake, M. D., & Skalka, A. M. (1996). Retroviral integrase, putting the pieces together. *The Journal of Biological Chemistry*, 271(33), 19633–6. Retrieved from <http://www.ncbi.nlm.nih.gov/pubmed/8702660>
- Bebenek, K., Abbotts, J., Wilson, S. H., & Kunkel, T. A. (1993). Error-prone polymerization by HIV-1 reverse transcriptase. Contribution of template-primer misalignment, miscoding, and termination probability to mutational hot spots. *The Journal of Biological Chemistry*, 268(14), 10324–34. Retrieved from

<http://www.ncbi.nlm.nih.gov/pubmed/7683675>

- Behrens, A. (2017). Fine structure of the HIV-1 glycan shield. Retrieved from <https://ora.ox.ac.uk/objects/uuid:3cec0ef7-c305-411e-a76b-125d5e7e9954>
- Bhiman, J. N., Anthony, C., Doria-Rose, N. A., Karimanzira, O., Schramm, C. A., Khoza, T., ... Moore, P. L. (2015). Viral variants that initiate and drive maturation of V1V2-directed HIV-1 broadly neutralizing antibodies. *Nature Medicine, advance on*. <http://doi.org/10.1038/nm.3963>
- Boltz, V. F., Rausch, J., Shao, W., Hattori, J., Luke, B., Maldarelli, F., ... Coffin, J. M. (2016). Ultrasensitive single-genome sequencing: accurate, targeted, next generation sequencing of HIV-1 RNA. *Retrovirology, 13*(1), 87. <http://doi.org/10.1186/s12977-016-0321-6>
- Brodin, J., Hedskog, C., Heddini, A., Benard, E., Neher, R. A., Mild, M., & Albert, J. (2015). Challenges with using primer IDs to improve accuracy of next generation sequencing. *PLoS One, 10*(3), e0119123. <http://doi.org/10.1371/journal.pone.0119123>
- Butler, D. M., Pacold, M. E., Jordan, P. S., Richman, D. D., & Smith, D. M. (2009). The efficiency of single genome amplification and sequencing is improved by quantitation and use of a bioinformatics tool. *Journal of Virological Methods, 162*(1–2), 280–283. <http://doi.org/10.1016/j.jviromet.2009.08.002>
- Casbon, J. A., Osborne, R. J., Brenner, S., & Lichtenstein, C. P. (2011). A method for counting PCR template molecules with application to next-generation sequencing. *Nucleic Acids Research, 39*(12), e81–e81. <http://doi.org/10.1093/nar/gkr217>
- Charif, D., & Lobry, J. R. (2007). SeqinR 1.0-2: A Contributed Package to the R Project for Statistical Computing Devoted to Biological Sequences Retrieval and Analysis. In U. Bastolla, M. Porto, E. Roman, & M. Vendruscolo (Eds.), *Structural Approaches to Sequence Evolution* (pp. 207–232). Springer-Verlag Berlin Heidelberg. http://doi.org/10.1007/978-3-540-35306-5_10
- Chun, T. W., Stuyver, L., Mizell, S. B., Ehler, L. A., Mican, J. A., Baseler, M., ... Fauci, A. S. (1997). Presence of an inducible HIV-1 latent reservoir during highly active antiretroviral therapy. *Proceedings of the National Academy of Sciences of the United States of America, 94*(24), 13193–7. Retrieved from <http://www.ncbi.nlm.nih.gov/pubmed/9371822>
- Crick, F. (1970). Central dogma of molecular biology. *Nature, 227*(5258), 561–3. Retrieved from <http://www.ncbi.nlm.nih.gov/pubmed/4913914>
- Crick, F. H. (1958). On protein synthesis. *Symposia of the Society for Experimental Biology, 12*, 138–63. Retrieved from <http://www.ncbi.nlm.nih.gov/pubmed/13580867>
- Davis, T. L. (2017). optparse: Command Line Option Parser. Retrieved from <https://cran.r-project.org/package=optparse>
- Doms, R. W., & Trono, D. (2000). The plasma membrane as a combat zone in the HIV battlefield. *Genes & Development, 14*(21), 2677–88. Retrieved from

<http://www.ncbi.nlm.nih.gov/pubmed/11069884>

- Dovichi, N. J., & Zhang, J. (2000). How capillary electrophoresis sequenced the human genome. *Angewandte Chemie International Edition*, 39(24), 4463–4468.
- Edgar, R. C. (2004). MUSCLE: multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Research*, 32(5), 1792–7. <http://doi.org/10.1093/nar/gkh340>
- Fairbanks, D. J., & Andersen, W. R. (1999). *Genetics*. Brooks/Cole Publishing Company.
- Fu, G. K., Hu, J., Wang, P.-H., & Fodor, S. P. A. (2011). Counting individual DNA molecules by the stochastic attachment of diverse labels. *Proceedings of the National Academy of Sciences of the United States of America*, 108(22), 9026–31. <http://doi.org/10.1073/pnas.1017621108>
- Fu, G. K., Xu, W., Wilhelmy, J., Mindrinos, M. N., Davis, R. W., Xiao, W., & Fodor, S. P. A. (2014). Molecular indexing enables quantitative targeted RNA sequencing and reveals poor efficiencies in standard library preparations. *Proceedings of the National Academy of Sciences of the United States of America*, 111(5), 1891–6. <http://doi.org/10.1073/pnas.1323732111>
- Gao, F., Chen, Y., Levy, D. N., Conway, J. A., Kepler, T. B., & Hui, H. (2004). Unselected mutations in the human immunodeficiency virus type 1 genome are mostly nonsynonymous and often deleterious. *Journal of Virology*, 78(5), 2426–33. Retrieved from <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=369203&tool=pmcentrez&endertype=abstract>
- Gilbert, P. B., Juraska, M., deCamp, A. C., Karuna, S., Edupuganti, S., Mgodi, N., ... Corey, L. (2017). Basis and Statistical Design of the Passive HIV-1 Antibody Mediated Prevention (AMP) Test-of-Concept Efficacy Trials. *Statistical Communications in Infectious Diseases*, 9(1). <http://doi.org/10.1515/scid-2016-0001>
- Giorgi, E. E., & Bhattacharya, T. (2012). A note on two-sample tests for comparing intra-individual genetic sequence diversity between populations. *Biometrics*, 68(4), 1323–6; author reply 1326. <http://doi.org/10.1111/j.1541-0420.2012.01775.x>
- Giorgi, E. E., Funkhouser, B., Athreya, G., Perelson, A. S., Korber, B. T., & Bhattacharya, T. (2010). Estimating time since infection in early homogeneous HIV-1 samples using a poisson model. *BMC Bioinformatics*, 11(1), 532. <http://doi.org/10.1186/1471-2105-11-532>
- Girard, M. P., Osmanov, S., Assossou, O. M., & Kieny, M. P. (2011). Human immunodeficiency virus (HIV) immunopathogenesis and vaccine development: A review. *Vaccine*, 29(37), 6191–6218. <http://doi.org/10.1016/j.vaccine.2011.06.085>
- Gottlieb, M. S., Schroff, R., Schanker, H. M., Weisman, J. D., Fan, P. T., Wolf, R. A., & Saxon, A. (1981). Pneumocystis carinii pneumonia and mucosal candidiasis in previously healthy

- homosexual men: evidence of a new acquired cellular immunodeficiency. *The New England Journal of Medicine*, 305(24), 1425–31. <http://doi.org/10.1056/NEJM198112103052401>
- Gray, G. E., Allen, M., Moodie, Z., Churchyard, G., Bekker, L.-G., Nchabeleng, M., ... HVTN 503/Phambili study team. (2011). Safety and efficacy of the HVTN 503/Phambili Study of a clade-B-based HIV-1 vaccine in South Africa: a double-blind, randomised, placebo-controlled test-of-concept phase 2b study. *The Lancet Infectious Diseases*, 11(7), 507–515. [http://doi.org/10.1016/S1473-3099\(11\)70098-6](http://doi.org/10.1016/S1473-3099(11)70098-6)
- Gregori, J., Perales, C., Rodriguez-Frias, F., Esteban, J. I., Quer, J., & Domingo, E. (2016). Viral quasispecies complexity measures. *Virology*. <http://doi.org/10.1016/j.virol.2016.03.017>
- Grothendieck, G. (2013). nls2: Non-linear regression with brute force. Retrieved from <https://cran.r-project.org/package=nls2>
- Guindon, S., Dufayard, J. F., Lefort, V., Anisimova, M., Hordijk, W., & Gascuel, O. (2010). New algorithms and methods to estimate maximum-likelihood phylogenies: Assessing the performance of PhyML 3.0. *Systematic Biology*, 59(3), 307–321. <http://doi.org/10.1093/sysbio/syq010>
- Holm, S. (1979). A Simple Sequentially Rejective Multiple Test Procedure. *Scandinavian Journal of Statistics*, 6(2), 65–70. Retrieved from <http://www.jstor.org/stable/4615733>
- Hong, L. Z., Hong, S., Wong, H., Aw, P., Cheng, Y., Wilm, A., ... Burkholder, W. F. (2014). BAsE-Seq: a method for obtaining long viral haplotypes from short sequence reads. *Genome Biology*, 15(11), 517. <http://doi.org/10.1186/s13059-014-0517-9>
- Illumina. (2015). Illumina MiSeq Specifications. Retrieved from http://www.illumina.com/content/illumina-marketing/amr/en_US/systems/miseq/performance_specifications.html
- Jabara, C. B., Jones, C. D., Roach, J., Anderson, J. A., & Swanstrom, R. (2011). Accurate sampling and deep sequencing of the HIV-1 protease gene using a Primer ID. *Proceedings of the National Academy of Sciences of the United States of America*, 108(50), 20166–71. <http://doi.org/10.1073/pnas.1110064108>
- Kanagawa, T. (2003). Bias and Artifacts in Multitemplate Polymerase Chain Reactions(PCR). *Journal of Bioscience and Bioengineering*, 96(4), 317–323. <http://doi.org/10.1263/jbb.96.317>
- Katoh, K., Misawa, K., Kuma, K., & Miyata, T. (2002). MAFFT: a novel method for rapid multiple sequence alignment based on fast Fourier transform. *Nucleic Acids Research*, 30(14), 3059–66. Retrieved from <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=135756&tool=pmcentrez&rendertype=abstract>

- Keele, B. F., Giorgi, E. E., Salazar-Gonzalez, J. F., Decker, J. M., Pham, K. T., Salazar, M. G., ... Shaw, G. M. (2008). Identification and characterization of transmitted and early founder virus envelopes in primary HIV-1 infection. *Proceedings of the National Academy of Sciences*, *105*(21), 7552–7557. <http://doi.org/10.1073/pnas.0802203105>
- Kinde, I., Wu, J., Papadopoulos, N., Kinzler, K. W., & Vogelstein, B. (2011). Detection and quantification of rare mutations with massively parallel sequencing. *Proceedings of the National Academy of Sciences of the United States of America*, *108*(23), 9530–5. <http://doi.org/10.1073/pnas.1105422108>
- Kivioja, T., Vähärautio, A., Karlsson, K., Bonke, M., Enge, M., Linnarsson, S., & Taipale, J. (2012). Counting absolute numbers of molecules using unique molecular identifiers. *Nature Methods*, *9*(1), 72–4. <http://doi.org/10.1038/nmeth.1778>
- Kohl, N. E., Emini, E. A., Schleif, W. A., Davis, L. J., Heimbach, J. C., Dixon, R. A., ... Sigal, I. S. (1988). Active human immunodeficiency virus protease is required for viral infectivity. *Proceedings of the National Academy of Sciences of the United States of America*, *85*(13), 4686–90. Retrieved from <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=280500&tool=pmcentrez&rendertype=abstract>
- Kou, R., Lam, H., Duan, H., Ye, L., Jongkam, N., Chen, W., ... Li, S. (2016). Benefits and Challenges with Applying Unique Molecular Identifiers in Next Generation Sequencing to Detect Low Frequency Mutations. *PloS One*, *11*(1), e0146638. <http://doi.org/10.1371/journal.pone.0146638>
- LaFemina, R. L., Schneider, C. L., Robbins, H. L., Callahan, P. L., LeGrow, K., Roth, E., ... Emini, E. A. (1992). Requirement of active human immunodeficiency virus type 1 integrase enzyme for productive infection of human T-lymphoid cells. *Journal of Virology*, *66*(12), 7414–9. Retrieved from <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=240448&tool=pmcentrez&rendertype=abstract>
- Langmead, B., Trapnell, C., Pop, M., & Salzberg, S. L. (2009). Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biology*, *10*(3), R25. <http://doi.org/10.1186/gb-2009-10-3-r25>
- Lee, H. Y., Giorgi, E. E., Keele, B. F., Gaschen, B., Athreya, G. S., Salazar-Gonzalez, J. F., ... Perelson, A. S. (2009). Modeling sequence evolution in acute HIV-1 infection. *Journal of Theoretical Biology*, *261*(2), 341–60. <http://doi.org/10.1016/j.jtbi.2009.07.038>
- Li, H., & Durbin, R. (2009). Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics (Oxford, England)*, *25*(14), 1754–60. <http://doi.org/10.1093/bioinformatics/btp324>
- Liu, J., Song, H., Liu, D., Zuo, T., Lu, F., Zhuang, H., & Gao, F. (2014). Extensive recombination due to heteroduplexes generates large amounts of artificial gene fragments

- during pcr. *PLoS ONE*, 9(9), e106658. <http://doi.org/10.1371/journal.pone.0106658>
- Liu, L., Li, Y., Li, S., Hu, N., He, Y., Pong, R., ... Law, M. (2012). Comparison of next-generation sequencing systems. *Journal of Biomedicine & Biotechnology*, 2012, 251364. <http://doi.org/10.1155/2012/251364>
- Lundberg, D. S., Yourstone, S., Mieczkowski, P., Jones, C. D., & Dangl, J. L. (2013). Practical innovations for high-throughput amplicon sequencing. *Nature Methods*, 10(10), 999–1002. <http://doi.org/10.1038/nmeth.2634>
- Mansky, L. M., & Temin, H. M. (1995). Lower in vivo mutation rate of human immunodeficiency virus type 1 than that predicted from the fidelity of purified reverse transcriptase. *Journal of Virology*, 69(8), 5087–94. Retrieved from <http://www.ncbi.nlm.nih.gov/pubmed/7541846>
- Masur, H., Michelis, M. A., Greene, J. B., Onorato, I., Stouwe, R. A., Holzman, R. S., ... Cunningham-Rundles, S. (1981). An outbreak of community-acquired *Pneumocystis carinii* pneumonia: initial manifestation of cellular immune dysfunction. *The New England Journal of Medicine*, 305(24), 1431–8. <http://doi.org/10.1056/NEJM198112103052402>
- McMichael, A. J., & Koff, W. C. (2014). Vaccines that stimulate T cell immunity to HIV-1: the next step. *Nature Immunology*, 15(4), 319–322. <http://doi.org/10.1038/ni.2844>
- Miller, M. D., Farnet, C. M., & Bushman, F. D. (1997). Human immunodeficiency virus type 1 preintegration complexes: studies of organization and composition. *Journal of Virology*, 71(7), 5382–90. Retrieved from <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=191777&tool=pmcentrez&rendertype=abstract>
- Miller, R. G. J. (2012). *Simultaneous Statistical Inference*. Springer Science & Business Media. Retrieved from <https://books.google.com/books?hl=en&lr=&id=4C7VBwAAQBAJ&pgis=1>
- Morgan, M., Anders, S., Lawrence, M., Aboyoun, P., Pagès, H., & Gentleman, R. (2009). {ShortRead}: a {B}ioconductor package for input, quality assessment and exploration of high-throughput sequence data. *Bioinformatics*, 25, 2607–2608. <http://doi.org/10.1093/bioinformatics/btp450>
- Nachega, J. B., Marconi, V. C., van Zyl, G. U., Gardner, E. M., Preiser, W., Hong, S. Y., ... Gross, R. (2011). HIV treatment adherence, drug resistance, virologic failure: evolving concepts. *Infectious Disorders Drug Targets*, 11(2), 167–74. Retrieved from <http://www.ncbi.nlm.nih.gov/pubmed/21406048>
- Nowak, M. A. (1992). What is a quasispecies? *Trends in Ecology & Evolution*, 7(4), 118–21. [http://doi.org/10.1016/0169-5347\(92\)90145-2](http://doi.org/10.1016/0169-5347(92)90145-2)
- Ochsenbauer, C., Edmonds, T. G., Ding, H., Keele, B. F., Decker, J., Salazar, M. G., ... Kappes,

- J. C. (2012). Generation of Transmitted/Founder HIV-1 Infectious Molecular Clones and Characterization of Their Replication Capacity in CD4 T Lymphocytes and Monocyte-Derived Macrophages. *Journal of Virology*, 86(5), 2715–2728. <http://doi.org/10.1128/JVI.06157-11>
- OpenStax-College. (2015). *Anatomy & Physiology*. OpenStax CNX. Retrieved from <http://cnx.org/contents/14fb4ad7-39a1-4eee-ab6e-3ef2482e3e22@7.30>
- Ouzounis, C. A., & Valencia, A. (2003). Early bioinformatics: the birth of a discipline--a personal view. *Bioinformatics (Oxford, England)*, 19(17), 2176–90. Retrieved from <http://www.ncbi.nlm.nih.gov/pubmed/14630646>
- Pages, H., Aboyou, P., Gentleman, R., & DebRoy, S. (2017). Biostrings: String objects representing biological sequences, and matching algorithms. Retrieved from <http://bioconductor.org/packages/release/bioc/html/Biostrings.html>
- Palmer, S., Kearney, M., Maldarelli, F., Halvas, E. K., Bixby, C. J., Bazmi, H., ... Coffin, J. M. (2005). Multiple, Linked Human Immunodeficiency Virus Type 1 Drug Resistance Mutations in Treatment-Experienced Patients Are Missed by Standard Genotype Analysis. *Journal of Clinical Microbiology*, 43(1), 406–413. <http://doi.org/10.1128/JCM.43.1.406-413.2005>
- Pau, A. K., & George, J. M. (2014). Antiretroviral therapy: current drugs. *Infectious Disease Clinics of North America*, 28(3), 371–402. <http://doi.org/10.1016/j.idc.2014.06.001>
- Perelson, A. S., Neumann, A. U., Markowitz, M., Leonard, J. M., & Ho, D. D. (1996). HIV-1 dynamics in vivo: virion clearance rate, infected cell life-span, and viral generation time. *Science (New York, N.Y.)*, 271(5255), 1582–6. Retrieved from <http://www.ncbi.nlm.nih.gov/pubmed/8599114>
- Polz, M. F., & Cavanaugh, C. M. (1998). Bias in template-to-product ratios in multitemplate PCR. *Applied and Environmental Microbiology*, 64(10), 3724–30. Retrieved from <http://www.ncbi.nlm.nih.gov/pubmed/9758791>
- Price, M. N., Dehal, P. S., & Arkin, A. P. (2009). FastTree: computing large minimum evolution trees with profiles instead of a distance matrix. *Molecular Biology and Evolution*, 26(7), 1641–50. <http://doi.org/10.1093/molbev/msp077>
- Rambaut, A., Posada, D., Crandall, K. A., & Holmes, E. C. (2004). The causes and consequences of HIV evolution. *Nature Reviews. Genetics*, 5(1), 52–61. <http://doi.org/10.1038/nrg1246>
- Ranwez, V., Harispe, S., Delsuc, F., & Douzery, E. J. P. (2011). MACSE: Multiple Alignment of Coding SEquences Accounting for Frameshifts and Stop Codons. *PLoS One*, 6(9), e22594. <http://doi.org/10.1371/journal.pone.0022594>
- Revolution Analytics, & Weston, S. (2015). foreach: Provides Foreach Looping Construct for R. Retrieved from <https://cran.r-project.org/package=foreach>

- Rigby, R. A., & Stasinopoulos, D. M. (2005). Generalized additive models for location, scale and shape (with discussion). *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 54(3), 507–554. <http://doi.org/10.1111/j.1467-9876.2005.00510.x>
- Rose, P. P., & Korber, B. T. (2000). Detecting hypermutations in viral sequences with an emphasis on G-to-A hypermutation. *Bioinformatics (Oxford, England)*, 16(4), 400–1. <http://doi.org/Doi 10.1093/Bioinformatics/16.4.400>
- Saiki, R., Gelfand, D., Stoffel, S., Scharf, S., Higuchi, R., Horn, G., ... Erlich, H. (1988). Primer-directed enzymatic amplification of DNA with a thermostable DNA polymerase. *Science*, 239(4839), 487–491. <http://doi.org/10.1126/science.2448875>
- Saiki, R. K., Scharf, S., Faloona, F., Mullis, K. B., Horn, G. T., Erlich, H. A., & Arnheim, N. (1985). Enzymatic amplification of beta-globin genomic sequences and restriction site analysis for diagnosis of sickle cell anemia. *Science (New York, N.Y.)*, 230(4732), 1350–4. Retrieved from <http://www.ncbi.nlm.nih.gov/pubmed/2999980>
- Sanger, F., Nicklen, S., & Coulson, A. R. (1977). DNA sequencing with chain-terminating inhibitors. *Proceedings of the National Academy of Sciences of the United States of America*, 74(12), 5463–7. Retrieved from <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=431765&tool=pmcentrez&rendertype=abstract>
- Schierup, M. H., & Hein, J. (2000). Recombination and the Molecular Clock. *Molecular Biology and Evolution*, 17(10), 1578–1579. <http://doi.org/10.1093/oxfordjournals.molbev.a026256>
- Schirmer, M., Ijaz, U. Z., D'Amore, R., Hall, N., Sloan, W. T., & Quince, C. (2015). Insight into biases and sequencing errors for amplicon sequencing with the Illumina MiSeq platform. *Nucleic Acids Research*, gku1341-. <http://doi.org/10.1093/nar/gku1341>
- Shao, W., Boltz, V. F., Spindler, J. E., Kearney, M. F., Maldarelli, F., Mellors, J. W., ... Coffin, J. M. (2013). Analysis of 454 sequencing error rate, error sources, and artifact recombination for detection of Low-frequency drug resistance mutations in HIV-1 DNA. *Retrovirology*, 10, 18. <http://doi.org/10.1186/1742-4690-10-18>
- Sheward, D. J., Murrell, B., & Williamson, C. (2012). Degenerate Primer IDs and the birthday problem. *Proceedings of the National Academy of Sciences of the United States of America*, 109(21), E1330; author reply E1331. <http://doi.org/10.1073/pnas.1203613109>
- Shiroguchi, K., Jia, T. Z., Sims, P. A., & Xie, X. S. (2012). Digital RNA sequencing minimizes sequence-dependent bias and amplification noise with optimized single-molecule barcodes. *Proceedings of the National Academy of Sciences of the United States of America*, 109(4), 1347–52. <http://doi.org/10.1073/pnas.1118018109>
- Simon, V., & Ho, D. D. (2003). HIV-1 dynamics in vivo: implications for therapy. *Nature Reviews Microbiology*, 1(3), 181–190. <http://doi.org/10.1038/nrmicro772>

- Thomas, R. K., Nickerson, E., Simons, J. F., Jänne, P. A., Tengs, T., Yuza, Y., ... Meyerson, M. (2006). Sensitive mutation detection in heterogeneous cancer specimens by massively parallel picoliter reactor sequencing. *Nature Medicine*, *12*(7), 852–5. <http://doi.org/10.1038/nm1437>
- Thompson, J. R., Marcelino, L. A., & Polz, M. F. (2002). Heteroduplexes in mixed-template amplifications: formation, consequence and elimination by “reconditioning PCR”. *Nucleic Acids Research*, *30*(9), 2083–2088. <http://doi.org/10.1093/nar/30.9.2083>
- van der Loo, M. (2014). {stringdist}: an {R} Package for Approximate String Matching. *The R Journal*, *6*(1), 111–122.
- Voelkerding, K. V., Dames, S. A., & Durtschi, J. D. (2009). Next-generation sequencing: from basic research to diagnostics. *Clinical Chemistry*, *55*(4), 641–58. <http://doi.org/10.1373/clinchem.2008.112789>
- Wagner, A., Blackstone, N., Cartwright, P., Dick, M., Misof, B., Snow, P., ... Pendleton, J. (1994). Surveys of Gene Families Using Polymerase Chain Reaction: PCR Selection and PCR Drift. *Systematic Biology*, *43*(2), 250. <http://doi.org/10.2307/2413465>
- Watson, J. D., & Crick, F. H. (1953). Molecular structure of nucleic acids; a structure for deoxyribose nucleic acid. *Nature*, *171*(4356), 737–8. Retrieved from <http://www.ncbi.nlm.nih.gov/pubmed/13054692>
- Wickham, H. (2011). testthat: Get Started with Testing. *The R Journal*, *3*, 5–10. Retrieved from https://journal.r-project.org/archive/2011-1/RJournal_2011-1_Wickham.pdf
- Yourstone, S. M., Lundberg, D. S., Dangl, J. L., & Jones, C. D. (2014). MT-Toolbox: improved amplicon sequencing using molecule tags. *BMC Bioinformatics*, *15*(1), 284. <http://doi.org/10.1186/1471-2105-15-284>
- Yu G, Smith D, Zhu H, G. Y. and L. T. (n.d.). (submitted) ggtree: an R package for visualization and annotation of phylogenetic tree with different types of meta-data. *Methods in Ecology and Evolution*.
- Zagordi, O., Bhattacharya, A., Eriksson, N., & Beerenwinkel, N. (2011). ShoRAH: estimating the genetic diversity of a mixed sample from next-generation sequencing data. *BMC Bioinformatics*, *12*(1), 119. <http://doi.org/10.1186/1471-2105-12-119>
- Zanini, F., Brodin, J., Thebo, L., Lanz, C., Bratt, G., Albert, J., & Neher, R. A. (2015). Population genomics of inpatient HIV-1 evolution. *Populations and Evolution; Genomics*.
- Zhang, J., Kobert, K., Flouri, T., & Stamatakis, A. (2014). PEAR: a fast and accurate Illumina Paired-End reAd mergeR. *Bioinformatics (Oxford, England)*, *30*(5), 614–20. <http://doi.org/10.1093/bioinformatics/btt593>
- Zhang, L., Gilbert, P. B., Capparelli, E., & Huang, Y. (2018). Pharmacokinetics Simulations for Studying Correlates of Prevention Efficacy of Passive HIV-1 Antibody Prophylaxis in

the Antibody Mediated Prevention (AMP) Study. Retrieved from <http://arxiv.org/abs/1801.08626>

Zhou, Q., Chen, D., Pierstorff, E., & Luo, K. (1998). Transcription elongation factor P-TEFb mediates Tat activation of HIV-1 transcription at multiple stages. *The EMBO Journal*, 17(13), 3681–91. <http://doi.org/10.1093/emboj/17.13.3681>

Zhou, Q., & Sharp, P. A. (1995). Novel mechanism and factor for regulation by HIV-1 Tat. *The EMBO Journal*, 14(2), 321–8. Retrieved from <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=398086&tool=pmcentrez&rendertype=abstract>

Zhou, S., Jones, C., Mieczkowski, P., & Swanstrom, R. (2015). Primer ID Validates Template Sampling Depth and Greatly Reduces the Error Rate of Next Generation Sequencing of HIV-1 Genomic RNA Populations. *Journal of Virology*, 89(16), 8540–55. <http://doi.org/10.1128/JVI.00522-15>

Zhou, T., Georgiev, I., Wu, X., Yang, Z.-Y., Dai, K., Finzi, A., ... Kwong, P. D. (2010). Structural basis for broad and potent neutralization of HIV-1 by antibody VRC01. *Science (New York, N.Y.)*, 329(5993), 811–7. <http://doi.org/10.1126/science.1192819>

8 Appendices

8.1 Binning Report (for the 006wpi dataset)

The material presented in this section of the appendix is an edited version of the HTML report produced by MotifBinner for the processing of the 006wpi dataset as described in section 3.2. The edits focused on the formatting of the headings and tables to ensure compatibility with the style of the thesis.

8.1.1 Input Sequences and Motifs Found

Input file: CAP256_2000_006wpi_v1_v3a.fastq

There were 38744 input sequences.

PIDs were found in a total of 36994 sequences.

This is 95.48% of the input sequences.

A total of 4226 unique PID were found.

8.1.2 Sequence Lengths

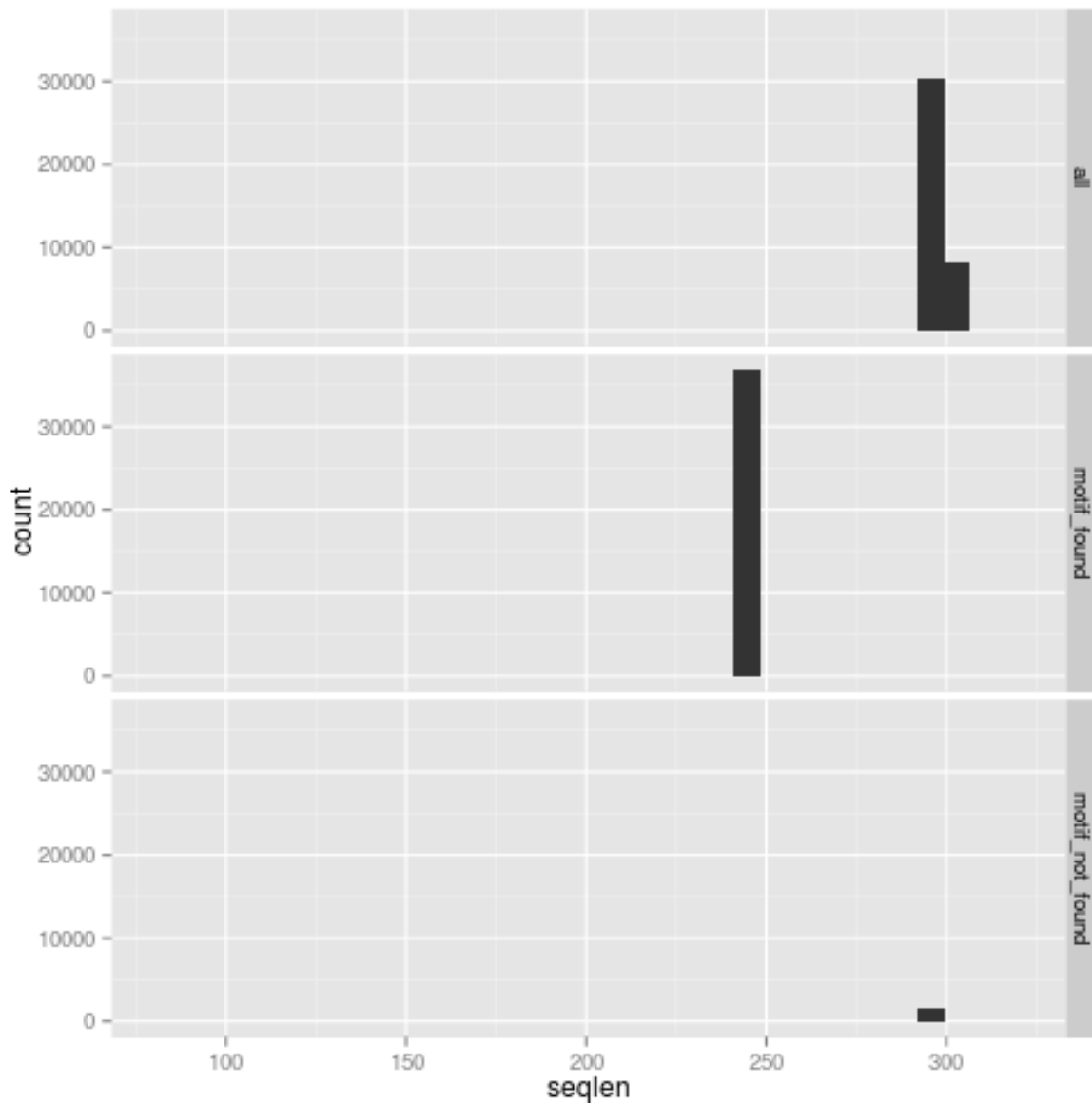


Figure: Histograms of the sequence lengths. The 'all' dataset contains all the input sequences; the 'motif_found' dataset contains all those sequences in which motifs have been found (and trimmed out) and the motif_not_found dataset contains all the sequences in which no motifs were found. Note that the motif_found dataset has the primers trimmed off so its sequences are a little shorter than they were in the input dataset. One would expect that the 'motif_not_found' dataset will contain the majority of the sequences of an unexpected length.

8.1.3 Bin Sizes

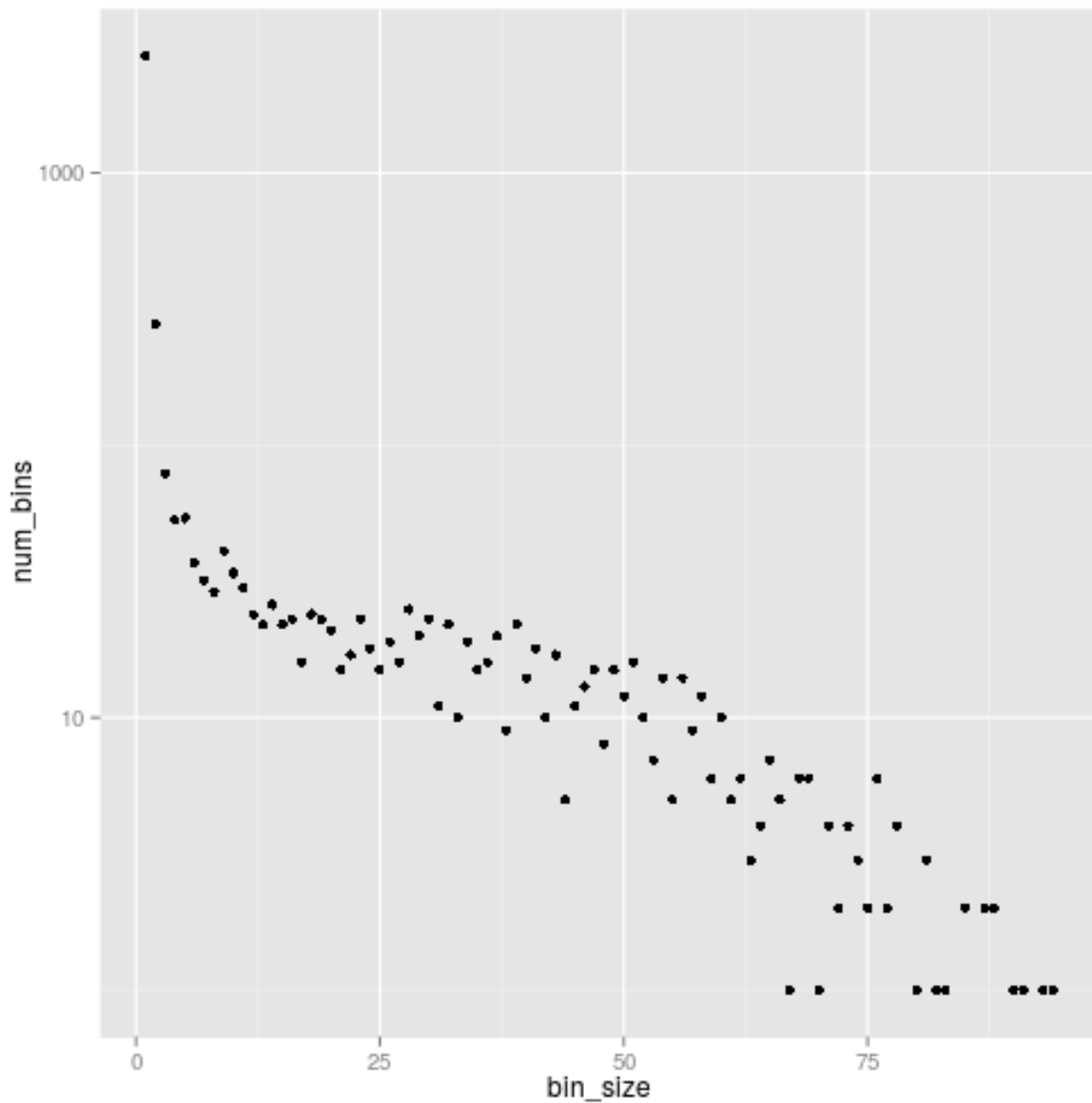


Figure: Number of bins by bin size. The y axis counts the number of bins of the size given on the x-axis. Note the log scale on the y-axis. Note also that this is the bin sizes before outlier removal was applied.

Table: The number of bins that fall in the given size range.

bin_size	num_bins
1	2695
2	278
3	79
4	53
5-10	227
11-20	231
21-400	663
401+	0

8.1.4 Consensus Cutoff

The largest bin has 94. Use this number when computing the consensus cutoff.

A consensus cutoff of 3.6853318 was used.

8.1.5 Chimeric bins

A total of 291 bins out of the 928 bins seem to be chimeric. Note that only larger bins can be tested for chimerism, so these numbers may not match the number of bins reported elsewhere in this report.

8.1.6 Outlier Removal

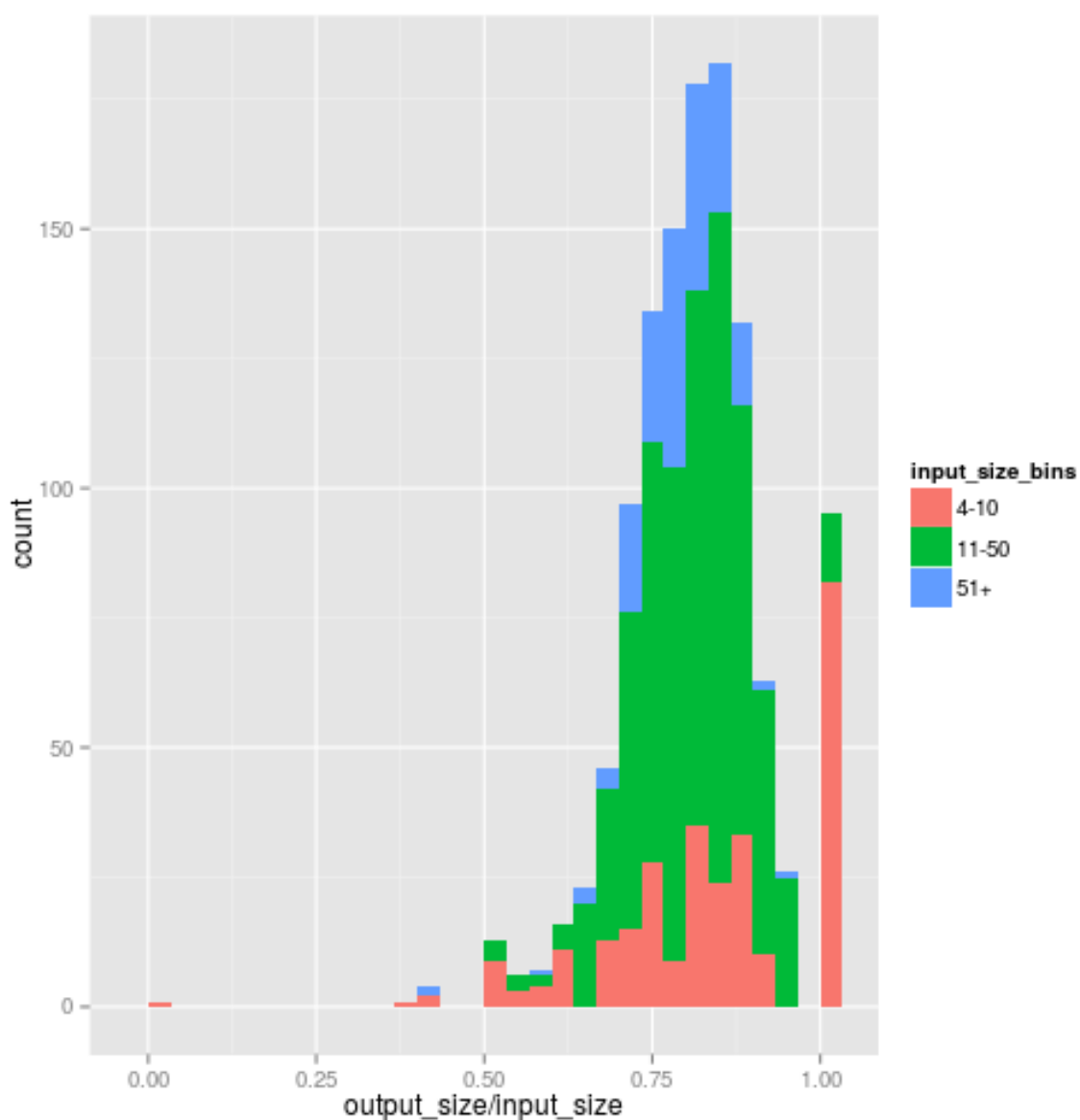


Figure: Histogram showing the effect of outlier removal on the bin sizes. The x-axis shows the size of the bin after outliers were removed as a fraction of the bin size before outliers was removed. The y-axis counts how many bins fall in each category. The fill color indicates the size of the bin

before outliers were removed. Note that bins of size 2 can only have the values 0 and 1. Likewise, bins of size 3 can only have the values 0, 0.67 and 1.

On average for bins with 2 or more sequences, the bin size was reduced by 19.05%. Since this is strongly affected by the size the bin was before outlier removal, this statistic is stratified by input bin size in the table below.

Table: Reduction in bin size due to the outlier removal process and the size of the bins after outlier removal as a fraction of input bin size stratified by input bin size.

input_bin_size	reduction	output_bin_size
4-10	17.16	82.84
11-50	19.27	80.73
51+	21.03	78.97

More details about the outlier removal can be found by inspecting the folder that contains all the bins. For each bin a plot will have been produced carefully illustrating the outlier removal procedure. Bins without plots were either too small or the PCA of the distance matrix failed probably because the sequences were too similar.

Table: The 10 bins that were shrunk the most by the outlier remover.

bin_pid	output_size	input_size	in_max_dist	min_dist
AAAAAAAAA	24	59	37	0
CATCGCCAC	23	54	128	0
ATGCAATAC	38	66	142	0
CGTCAAGCC	42	64	28	0
GCGATGTTC	51	73	34	0
GGACAAGGA	56	78	28	0
CGTCCCAA	55	76	26	0
GAGACAGCC	38	59	32	0
ACCCAAAGG	64	85	34	0
CCCGAGAAT	37	57	27	0

Table: The 10 bins that were shrunk the most by the outlier remover and which were smaller than 5 after outlier removal. If there are bins with large input_sizes in this table, they should probably be investigated.

bin_pid	output_size	input_size	in_max_dist	min_dist
ATGAATGTA	3	8	24	1
AAGCATTAA	0	4	27	13
ACCGCTCGG	4	8	14	1
AACACAACC	3	6	17	3
GATCCCACT	4	7	36	1
CAAAAATGG	4	7	9	2
ATGCTACAC	2	5	13	1
TTAATCAAC	4	7	22	1
CACAATGCC	4	7	26	2
CATATTCCC	2	5	14	4

8.1.7 Degeneracies in Consensus Sequences

Table: The number of sequences with the given number of degeneracies they have.

num_degeneracies	num_consensuses
0	1154
1	12
2	2
4	3
6	2

Table: Frequency table of the number of sequences with the given number of degeneracies they have (listed in the first column) and the size of the bins they were produced from (listed in the first row). Totals are added in the last column and last row.

	1	2	3	4-10	11-50	51+	Sum
0	0	0	33	286	761	74	1154
1	0	2	1	7	2	0	12
2	0	1	0	1	0	0	2
4	0	3	0	0	0	0	3
6	0	2	0	0	0	0	2
Sum	0	8	34	294	763	74	1173



8.1.8 Relatedness of Final Consensuses

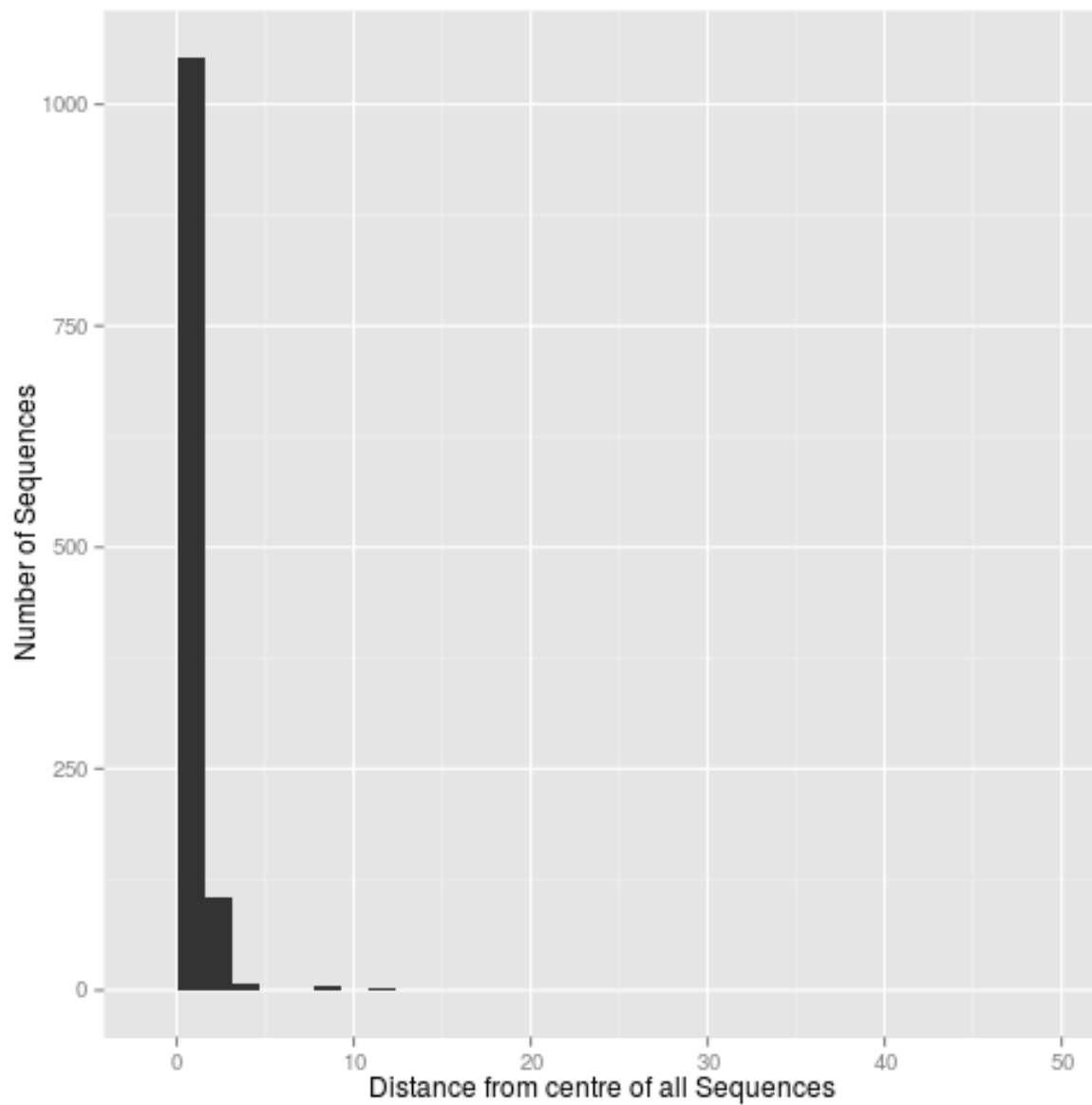


Figure: Histogram of the distances of each sequence from the centre of all the sequences. This plot is useful for spotting outliers.

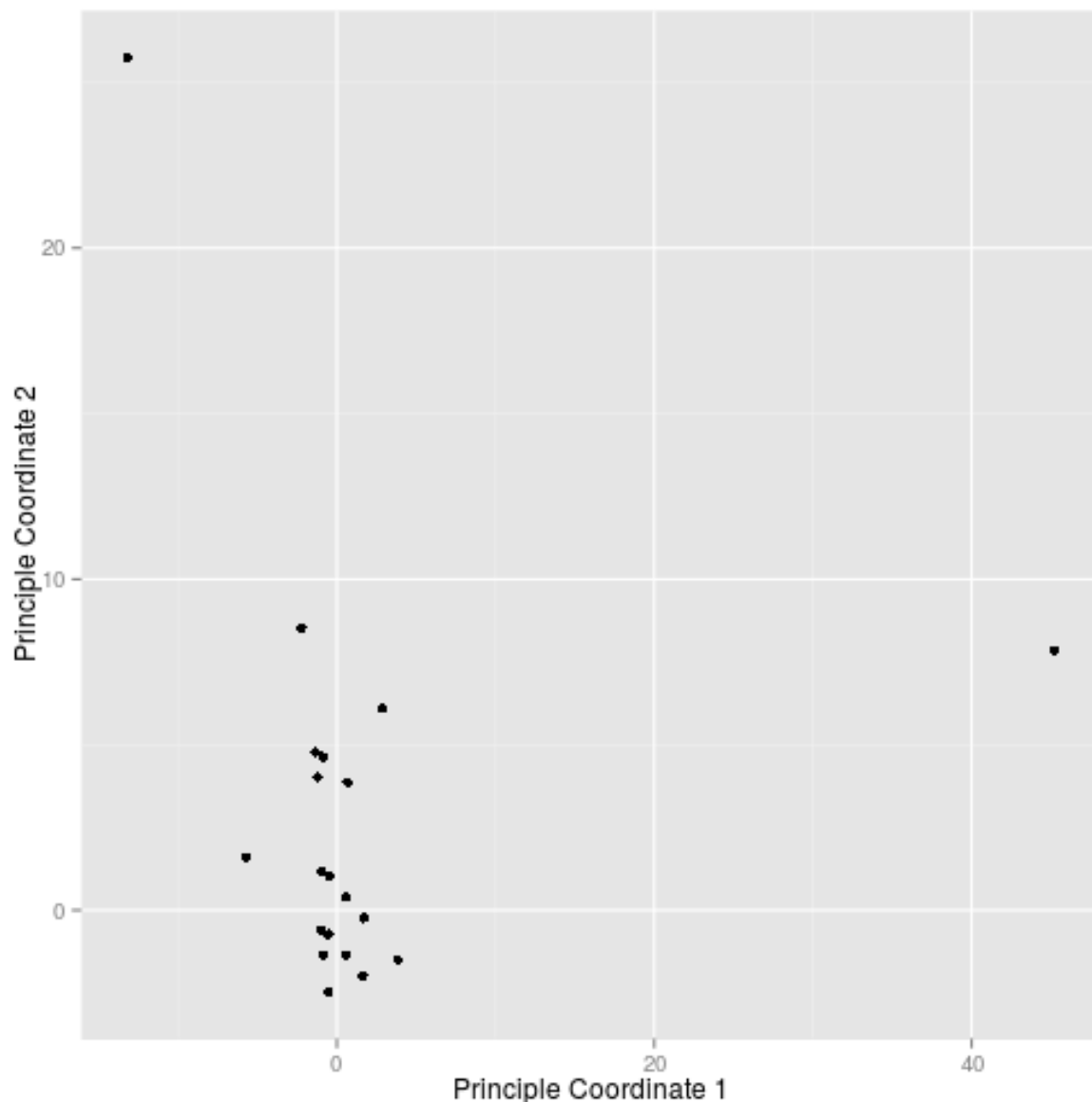


Figure: The approximate distances between the sequences displayed on the first two principle coordinates. Note that only unique sequences are presented on this plot. If the same sequence occurred more than once, only the first one was kept. 93 of the 93 unique consensus sequences are in this plot - The plot cannot be made on more than 2000 sequences, and if there were problems with the plotting, the dataset size was reduced until the plot worked.

Table: The centre most sequences. These sequences has the lowest average distance to all other sequences in the dataset. Rank indicates how many sequences are closer to the center of the dataset than this sequence and distance is the average distance to all other sequences in the dataset.

seq_name	rank	distance
CAP256_2000_006wpi_v1_v3a_CCCCTGCAC_76	527	401.0267
CAP256_2000_006wpi_v1_v3a_GTGAGACCA_75	527	401.0267
CAP256_2000_006wpi_v1_v3a_TCTGACATA_74	527	401.0267
CAP256_2000_006wpi_v1_v3a_AAACCTTGCA_74	527	401.0267
CAP256_2000_006wpi_v1_v3a_TTCCAAAGC_73	527	401.0267
CAP256_2000_006wpi_v1_v3a_TATGCTATA_72	527	401.0267

CAP256_2000_006wpi_v1_v3a_GCGCCTCCT_70	527	401.0267
CAP256_2000_006wpi_v1_v3a_CCGCTGTCG_69	527	401.0267
CAP256_2000_006wpi_v1_v3a_CCACTAGGG_69	527	401.0267
CAP256_2000_006wpi_v1_v3a_ATCACTCCA_68	527	401.0267
CAP256_2000_006wpi_v1_v3a_ATGCTTCCA_68	527	401.0267
CAP256_2000_006wpi_v1_v3a_AACACTTAA_68	527	401.0267
CAP256_2000_006wpi_v1_v3a_GACAAGTGG_68	527	401.0267
CAP256_2000_006wpi_v1_v3a_ACTAAAGAT_66	527	401.0267
CAP256_2000_006wpi_v1_v3a_CGCCAAATG_65	527	401.0267

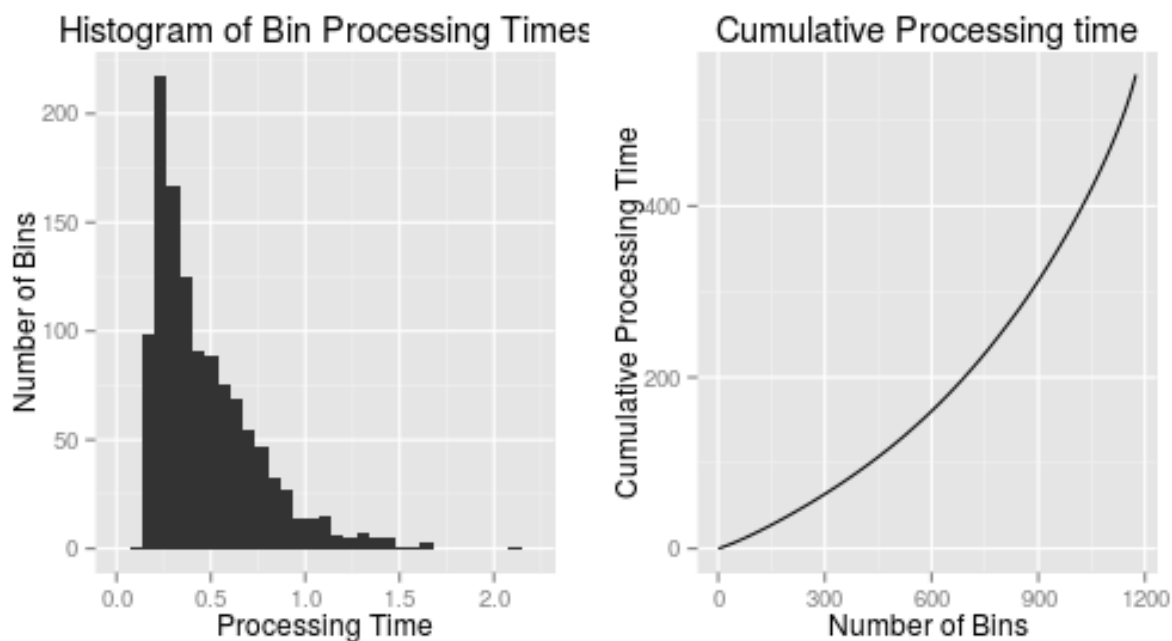
Table: The most outlying sequences. These sequences has the highest average distance to all other sequences in the dataset. Some of these should be manually inspected if they are too far from the other sequences. Rank indicates how many sequences are closer to the centre of the dataset than this sequence and distance is the average distance to all other sequences in the dataset.

seq_name	rank	distance
CAP256_2000_006wpi_v1_v3a_TTTTATATC_3	1159	4298.770
CAP256_2000_006wpi_v1_v3a_CACTATTGT_3	1160	4743.002
CAP256_2000_006wpi_v1_v3a_TCAAAATGG_18	1162	5129.947
CAP256_2000_006wpi_v1_v3a_TGATGACCG_4	1162	5129.947
CAP256_2000_006wpi_v1_v3a_CACCAGCCT_2	1162	5129.947
CAP256_2000_006wpi_v1_v3a_TTTACCCTG_13	1164	5324.097
CAP256_2000_006wpi_v1_v3a_ATCAAGGCA_14	1165	9349.896
CAP256_2000_006wpi_v1_v3a_CATATTCCC_2	1166	9358.296
CAP256_2000_006wpi_v1_v3a_AGACTCTAG_2	1167	9504.469
CAP256_2000_006wpi_v1_v3a_AGGACAACA_2	1168	9842.045
CAP256_2000_006wpi_v1_v3a_AAAATAAAG_2	1169	12965.560
CAP256_2000_006wpi_v1_v3a_GAGTACTACT_2	1170	14078.040
CAP256_2000_006wpi_v1_v3a_GAGTCGCAG_19	1171	20895.305
CAP256_2000_006wpi_v1_v3a_GATCACATA_14	1172	36579.769
CAP256_2000_006wpi_v1_v3a_CTTCAACCAT_38	1173	54967.219

8.1.9 Running Time

Table: The running times of the various steps in the binner.

task_name	end_time	task_running_time	total_running_time
start_run	2015-11-26 13:44:27	0.000 mins	0.000 mins
read_file	2015-11-26 13:44:29	0.032 mins	0.032 mins
motif_finding	2015-11-26 13:46:17	1.799 mins	1.831 mins
bin_by_name	2015-11-26 13:46:32	0.245 mins	2.076 mins
randomize_list	2015-11-26 13:46:32	0.003 mins	2.079 mins
process_bins	2015-11-26 13:49:05	2.561 mins	4.640 mins
format_consensus	2015-11-26 13:49:42	0.606 mins	5.246 mins
save_result	2015-11-26 13:50:34	0.878 mins	6.124 mins



8.1.10 Parameters

Name	Value
file_name	/fridge/data/binner_results/pipeline1/CAP256_2000_006wpi_v1_v3a/n05_pref_trim/CAP256_2000_006wpi_v1_v3a.fastq
output	/fridge/data/binner_results/pipeline1/CAP256_2000_006wpi_v1_v3a/n06_bin
prefix	CAGGAGGGGAYCTAGAARTTACAAC
suffix	CTGAGCGTGTG
motif_length	9
max.mismatch_start	0
max.mismatch	5
threshold	0.0133333333333333
start_threshold	0.0133333333333333
max_sequences	1000
dist_technique	vsearch_stringDist
remove_gaps	TRUE
strip_uids	TRUE
reporting_cutoffs	2
consensus_cutoff	model
n_bins_to_process	0
verbose	TRUE
prefix_for_names	CAP256_2000_006wpi_v1_v3a
pid_location	right
max_suffix_chop	3
pid_seq_name_file	NULL
match_name	sep_space_first
report	html, pdf
sequencing_error_rate	0.01

ncpu	4
dist_fallback_size	5

8.1.11 MotifBinner version

This is the version of MotifBinner that was used to make the report_dat object (aka binning_dat) from which this report was generated. This is not necessarily the same as the version of MotifBinner that was used to generate this report

```
[1] "1.1.6"
```



8.2 Binning Report (for the 193wpi dataset)

The material presented in this section of the appendix is an edited version of the HTML report produced by MotifBinner for the processing of the 193wpi dataset as described in section 3.2. The edits focused on the formatting of the headings and tables to ensure compatibility with the style of the thesis.

8.2.1 Input Sequences and Motifs Found

Input file: CAP256_4260_193wpi_v1_v3a.fastq

There were 44186 input sequences.

PIDs were found in a total of 41957 sequences.

This is 94.96% of the input sequences.

A total of 3725 unique PID were found.



8.2.2 Sequence Lengths

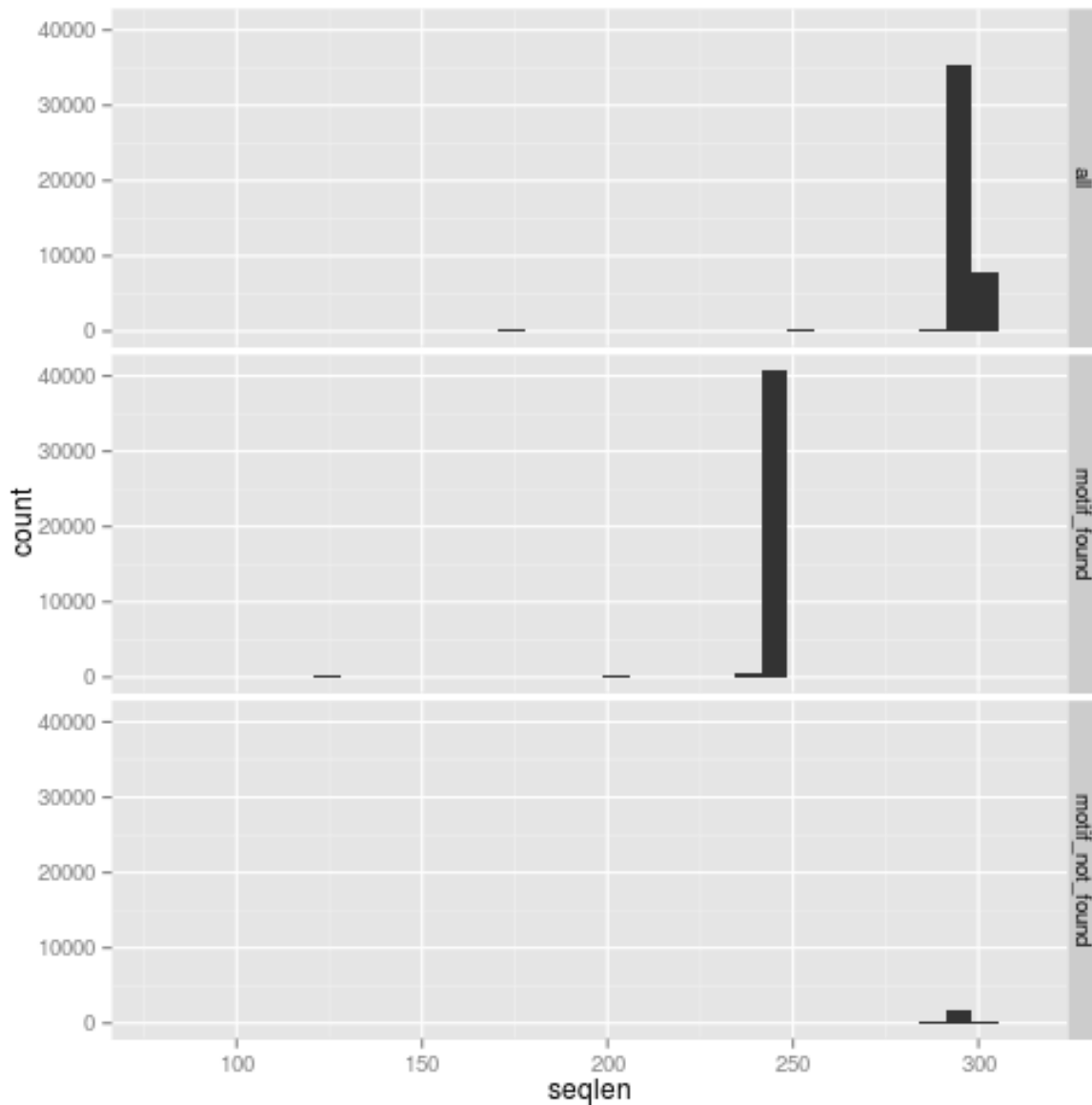


Figure: Histograms of the sequence lengths. The 'all' dataset contains all the input sequences; the 'motif_found' dataset contains all those sequences in which motifs have been found (and trimmed out) and the motif_not_found dataset contains all the sequences in which no motifs were found. Note that the motif_found dataset has the primers trimmed off so its sequences are a little shorter than they were in the input dataset. One would expect that the 'motif_not_found' dataset will contain the majority of the sequences of an unexpected length.

8.2.3 Bin Sizes

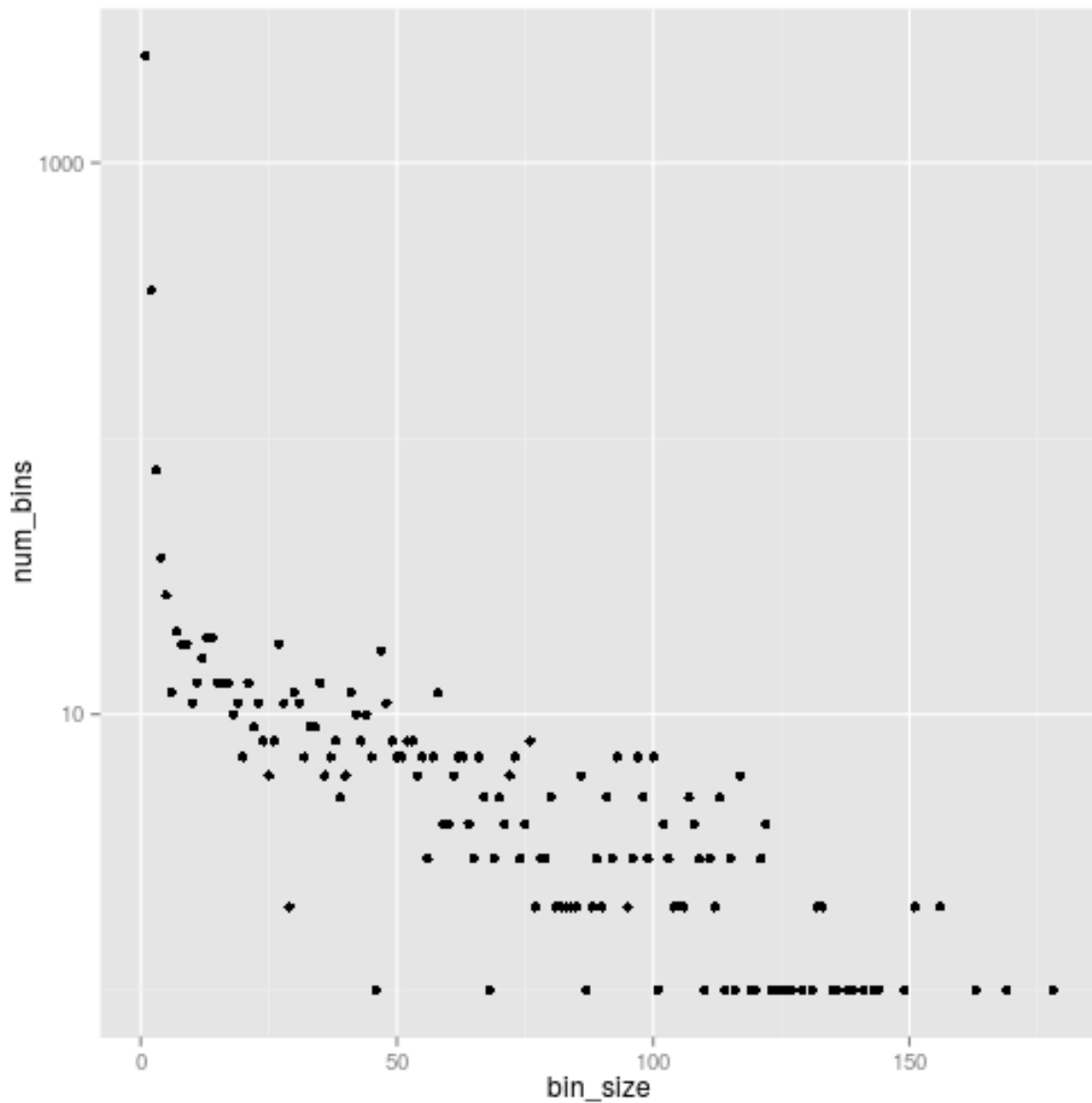


Figure: Number of bins by bin size. The y axis counts the number of bins of the size given on the x-axis. Note the log scale on the y-axis. Note also that this is the bin sizes before outlier removal was applied.

Table: The number of bins that fall in the given size range.

bin_size	num_bins
1	2449
2	344
3	77
4	37
5-10	106
11-20	134
21-400	578
401+	0

8.2.4 Consensus Cutoff

The largest bin has 178. Use this number when computing the consensus cutoff.

A consensus cutoff of 4.6616469 was used.

8.2.5 Chimeric bins

A total of 600 bins out of the 723 bins seem to be chimeric. Note that only larger bins can be tested for chimerism, so these numbers may not match the number of bins reported elsewhere in this report.

8.2.6 Outlier Removal

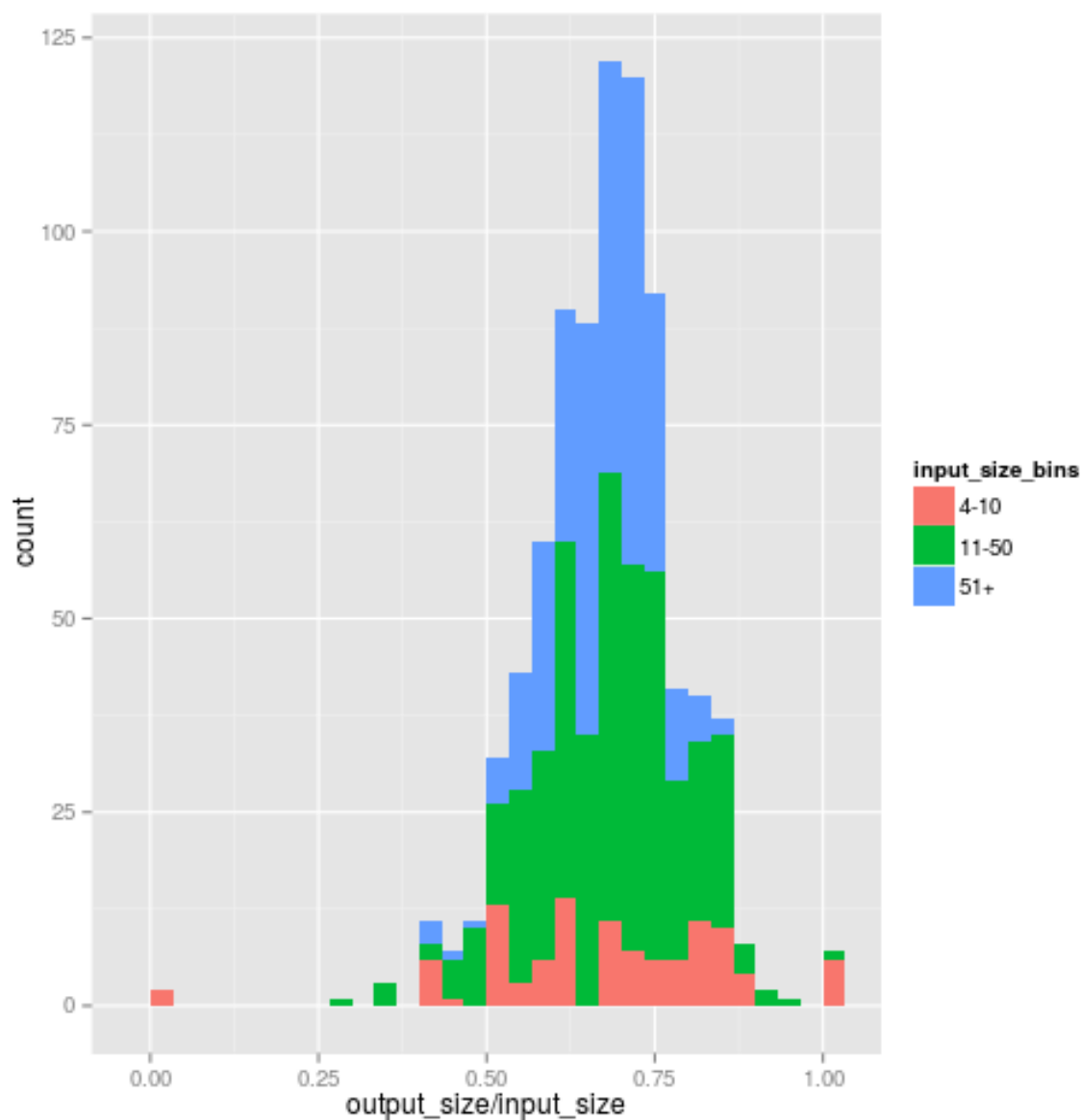


Figure: Histogram showing the effect of outlier removal on the bin sizes. The x-axis shows the size of the bin after outliers were removed as a fraction of the bin size before outliers was removed. The y-axis counts how many bins fall in each category. The fill color indicates the size of the bin before outliers were removed. Note that bins of size 2 can only have the values 0 and 1. Likewise, bins of size 3 can only have the values 0, 0.67 and 1.

On average for bins with 2 or more sequences, the bin size was reduced by 32.54%. Since this is strongly affected by the size the bin was before outlier removal, this statistic is stratified by input bin size in the table below.

Table: Reduction in bin size due to the outlier removal process and the size of the bins after outlier removal as a fraction of input bin size stratified by input bin size.

input_bin_size	reduction	output_bin_size
4-10	32.70	67.30
11-50	32.12	67.88
51+	33.05	66.95

More details about the outlier removal can be found by inspecting the folder that contains all the bins. For each bin a plot will have been produced carefully illustrating the outlier removal procedure. Bins without plots were either too small or the PCA of the distance matrix failed probably because the sequences were too similar.

Table: The 10 bins that were shrunk the most by the outlier remover

bin_pid	output_size	input_size	in_max_dist	min_dist
AGTAGCAGC	58	132	136	0
CACTTGAA	92	163	33	0
CCACGGAAC	108	178	68	0
GTTGTAAAC	93	156	39	0
ACCTCGCCG	53	113	133	0
CCCATACGG	111	169	42	0
TGTGAAGGA	79	133	39	0
CAGATTGAA	98	151	68	0
GCTACCAAT	86	138	47	0
ATCAGGCAC	69	120	52	0

Table: The 10 bins that were shrunk the most by the outlier remover and which were smaller than 5 after outlier removal. If there are bins with large input_sizes in this table, they should probably be investigated.

bin_pid	output_size	input_size	in_max_dist	min_dist
CCAATACTT	4	11	20	1
ATTGGCCCA	4	9	12	1
TGAATTCTA	3	7	12	1
CCATCAGGT	4	8	21	2
CCGCGGAAC	0	4	7	4
GACTTGCCA	4	8	14	1
CGTACCGCC	4	8	10	5
GCAGCTTTA	4	8	24	2
CGGTTAAGA	3	7	18	2
GAAATAACT	4	8	17	0

8.2.7 Degeneracies in Consensus Sequences

Table: The number of sequences with the given number of degeneracies they have.

num_degeneracies	num_consensuses
0	804
1	7
2	3
5	2

Table: Frequency table of the number of sequences with the given number of degeracies they have (listed in the first column) and the size of the bins they were produced from (listed in the first row). Totals are added in the last column and last row.

	1	2	3	4-10	11-50	51+	Sum
0	0	0	10	164	460	170	804
1	0	1	0	5	0	1	7
2	0	1	0	1	1	0	3
5	0	2	0	0	0	0	2
Sum	0	4	10	170	461	171	816



UNIVERSITY *of the*
WESTERN CAPE

8.2.8 Relatedness of Final Consensuses

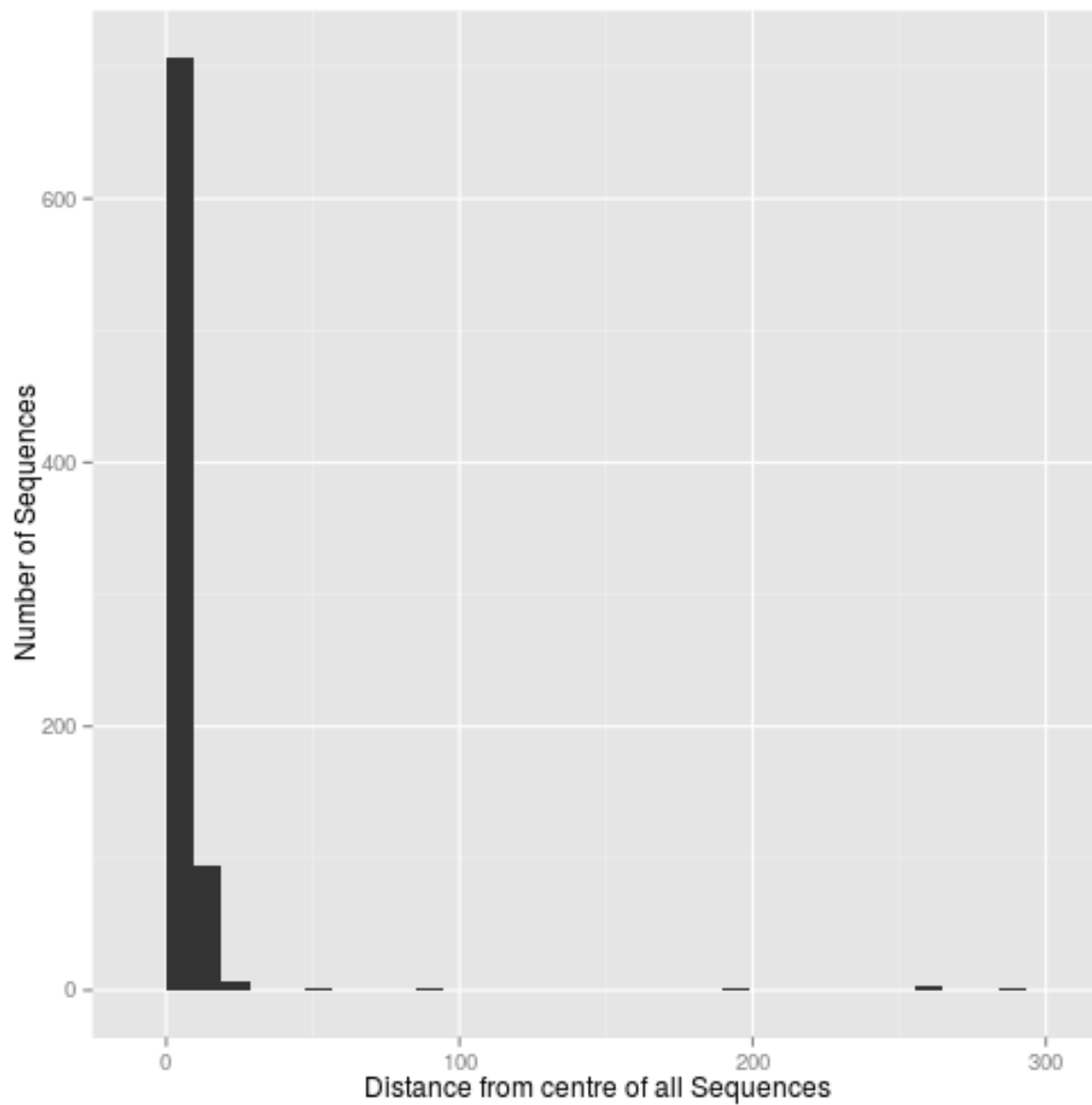


Figure: Histogram of the distances of each sequence from the centre of all the sequences. This plot is useful for spotting outliers.

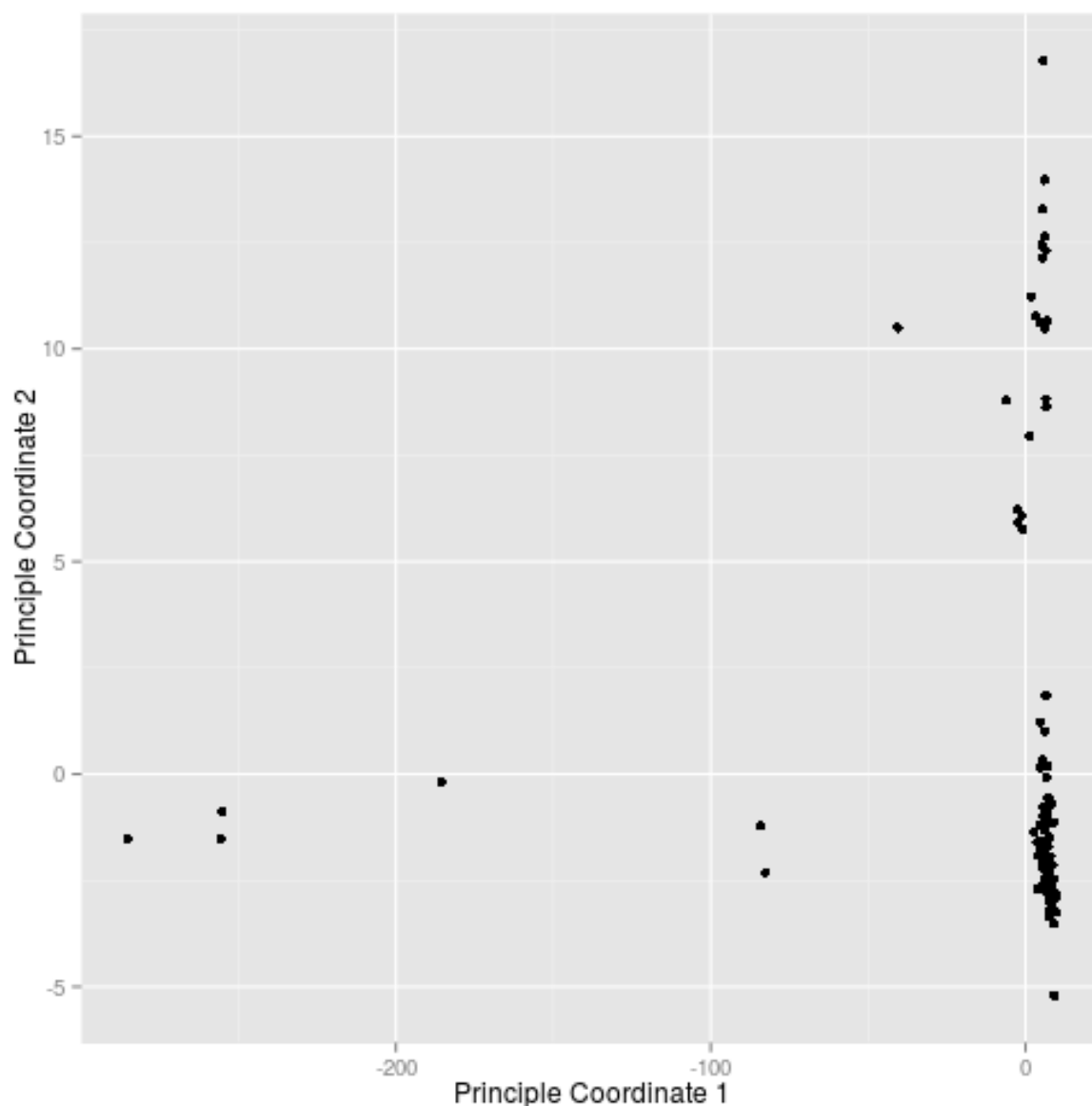


Figure: The approximate distances between the sequences displayed on the first two principle coordinates. Note that only unique sequences are presented on this plot. If the same sequence occurred more than once, only the first one was kept. 189 of the 189 unique consensus sequences are in this plot - The plot cannot be made on more than 2000 sequences, and if there were problems with the plotting, the dataset size was reduced until the plot worked.

Table: The centre most sequences. These sequences has the lowest average distance to all other sequences in the dataset. Rank indicates how many sequences are closer to the centre of the dataset than this sequence and distance is the average distance to all other sequences in the dataset.

seq_name	rank	distance
CAP256_4260_193wpi_v1_v3a_TGCCGATTC_8	1.0	5580.89
CAP256_4260_193wpi_v1_v3a_CTTTAAGAA_112	140.5	5639.71
CAP256_4260_193wpi_v1_v3a_CCCATACGG_111	140.5	5639.71
CAP256_4260_193wpi_v1_v3a_CCTTAATCA_101	140.5	5639.71
CAP256_4260_193wpi_v1_v3a_CAGATTGAA_98	140.5	5639.71
CAP256_4260_193wpi_v1_v3a_GCATGCCTA_96	140.5	5639.71

CAP256_4260_193wpi_v1_v3a_GAAGTCATA_91	140.5	5639.71
CAP256_4260_193wpi_v1_v3a_CACCCATGA_91	140.5	5639.71
CAP256_4260_193wpi_v1_v3a_TGGGGATTG_90	140.5	5639.71
CAP256_4260_193wpi_v1_v3a_CGCTGTA CT_87	140.5	5639.71
CAP256_4260_193wpi_v1_v3a_ATGTATTGA_85	140.5	5639.71
CAP256_4260_193wpi_v1_v3a_TCCCCCCG_85	140.5	5639.71
CAP256_4260_193wpi_v1_v3a_CGCCCTTAC_81	140.5	5639.71
CAP256_4260_193wpi_v1_v3a_AACCCCTTT_81	140.5	5639.71
CAP256_4260_193wpi_v1_v3a_GTAAGCCAC_79	140.5	5639.71

Table: The most outlying sequences. These sequences has the highest average distance to all other sequences in the dataset. Some of these should be manually inspected if they are too far from the other sequences. Rank indicates how many sequences are closer to the centre of the dataset than this sequence and distance is the average distance to all other sequences in the dataset.

seq_name	rank	distance
CAP256_4260_193wpi_v1_v3a_CAGCAGCAA_8	802.0	15555.62
CAP256_4260_193wpi_v1_v3a_CCGGAAGCA_80	805.5	16278.50
CAP256_4260_193wpi_v1_v3a_AGTGATGAT_31	805.5	16278.50
CAP256_4260_193wpi_v1_v3a_ACTGTAGAT_27	805.5	16278.50
CAP256_4260_193wpi_v1_v3a_AACAATCAA_26	805.5	16278.50
CAP256_4260_193wpi_v1_v3a_ACAAGTCAC_19	805.5	16278.50
CAP256_4260_193wpi_v1_v3a_ACAGGAGGT_12	805.5	16278.50
CAP256_4260_193wpi_v1_v3a_AGTAGGAGG_37	809.0	41254.15
CAP256_4260_193wpi_v1_v3a_AAAATGCAG_86	810.0	73742.44
CAP256_4260_193wpi_v1_v3a_CCACGGAAC_108	811.0	75198.40
CAP256_4260_193wpi_v1_v3a_GCAATACTC_5	812.0	156826.29
CAP256_4260_193wpi_v1_v3a_AGAGAAGCC_33	813.0	212898.43
CAP256_4260_193wpi_v1_v3a_ATTTACTTA_110	814.5	213240.67
CAP256_4260_193wpi_v1_v3a_ATTTACTGA_13	814.5	213240.67
CAP256_4260_193wpi_v1_v3a_AGTGGTTAT_38	816.0	237289.53

8.2.9 Running Time

Table: The running times of the various steps in the binner.

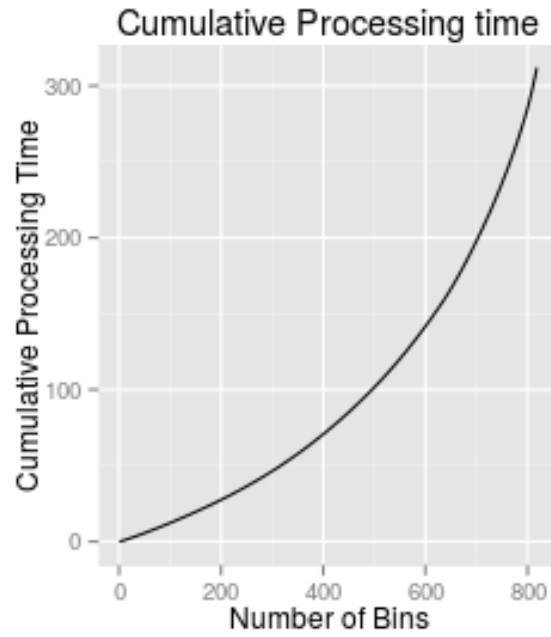
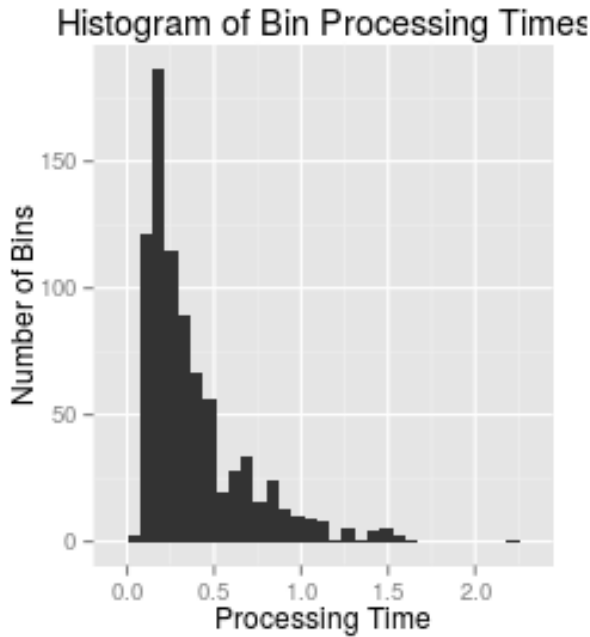
task_name	end_time	task_running_time	total_running_time
start_run	2015-12-03 14:27:25	0.000 mins	0.000 mins
read_file	2015-12-03 14:27:26	0.021 mins	0.021 mins
motif_finding	2015-12-03 14:28:37	1.178 mins	1.198 mins
bin_by_name	2015-12-03 14:28:44	0.120 mins	1.318 mins
randomize_list	2015-12-03 14:28:44	0.001 mins	1.319 mins
process_bins	2015-12-03 14:30:10	1.428 mins	2.748 mins
format_consensus	2015-12-03 14:30:25	0.258 mins	3.006 mins

save_result

2015-12-03 14:30:51

0.427 mins

3.433 mins



8.2.10 Parameters

Name	Value
file_name	/fridge/data/binner_results/pipeline1/CAP256_4260_193wpi_v1_v3a/n05_pref_trim/CAP256_4260_193wpi_v1_v3a.fastq
output	/fridge/data/binner_results/pipeline1/CAP256_4260_193wpi_v1_v3a/n06_bin
prefix	CAGGAGGGGAYCTAGAARTTACAAC
suffix	CTGAGCGTGTG
motif_length	9
max.mismatch_start	0
max.mismatch	5
threshold	0.0133333333333333
start_threshold	0.0133333333333333
max_sequences	1000
dist_technique	vsearch_stringDist
remove_gaps	TRUE
strip_uids	TRUE
reporting_cutoffs	2
consensus_cutoff	model
n_bins_to_process	0
verbose	TRUE
prefix_for_names	CAP256_4260_193wpi_v1_v3a
pid_location	right
max_suffix_chop	3
pid_seq_name_file	NULL
match_name	sep_space_first

report	html, pdf
sequencing_error_rate	0.01
ncpu	4
dist_fallback_size	5

8.2.11 MotifBinner version

This is the version of MotifBinner that was used to make the report_dat object (aka binning_dat) from which this report was generated. This is not necessarily the same as the version of MotifBinner that was used to generate this report


```
[1] "1.1.6"
```



8.3 Detailed methods for section 4.2.

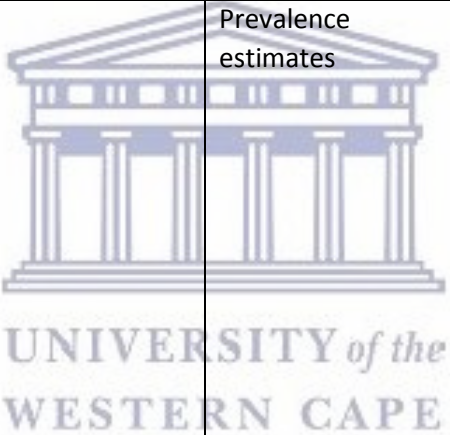
Table 46: Description of the process used to generate all the figures and tables in this chapter of the thesis.

Table Number	Brief Description	Methodology
Table 22	Hamming distance between input templates	Compute a Hamming distance matrix between the sequences of the input templates using the <code>stringdistmatrix</code> function from the <code>stringdist</code> R package.
Figure 45 (left)	PCR product concentration for each sample composition	Scatterplot with the sample composition on the x-axis as a categorical variable and the reported concentration of products from the PCR on the y-axis using the sample preparation protocol to color the points and the duration of the elongation step in the PCR to choose the shape of the object that represents the point.
Figure 45 (middle)	Number of raw sequences for each sample concentration	Scatterplot with the sample composition on the x-axis as a categorical variable and the number of raw sequences produced on the y-axis using the sample preparation protocol to color the points and the duration of the elongation step in the PCR to choose the shape of the object that represents the point. The number represent sequences that went through a few preliminary filtering steps as described in section 4.1.1.
Figure 45 (right)	Number of consensus sequences for each sample concentration	Scatterplot with the sample composition on the x-axis as a categorical variable and the number of consensus sequences (as described in section 4.1.2) produced on the y-axis using the sample preparation protocol to color the points and the duration of the elongation step in the PCR to choose the shape of the object that represents the point.
Figure 46	Distribution of bin sizes	Boxplots showing the distribution of the number of times each PID occurred in the merged sequences excluding singletons. Each boxplot was annotated with the number of PIDs that occurred only once. A separate boxplot was drawn for each sample.

Table 23	Per sample sequence accuracy and quality statistics	Using the arrays of accuracy and quality scores described in section 4.1.3, the error rates were computed for each dataset. The error rate is simply the number of bases that did not match the input template in all the sequences in the dataset, excluding positions where the input templates do not all match, divided by the total number of bases in the dataset corresponding to positions where the input templates are identical. The average quality score is the sum of all the quality scores of all bases at positions where the input templates are identical divided by the number of such datasets in the dataset.
Table 24	 <p>Aggregated sequence accuracy and quality statistics</p>	Using the arrays of accuracy and quality scores described in section 4.1.3, the overall error rates were computed for each of the four amplification protocols on the raw sequences and the consensus sequences. The error rate is the sum, across all datasets amplified with a given protocol, of the number of times the base in the raw (or consensus) sequence did not match the base in the input template at those positions where all three input templates are identical divided by total number of positions in all the datasets that correspond with positions where the input templates were identical. The average quality score is the sum of all quality scores across all positions where the input templates are identical divided by the total number of positions in all the datasets for the given amplification protocol where the three input templates are identical to each other.
Figure 47	Aggregated sequence accuracy and quality statistics	Aggregate error rates were computed using the same approach as for Table 24, but with different groupings of the datasets. Scatterplots were constructed from there error rates by plotting the error rate on the y-axis and some classification of the protocols or datasets on the x-axis as a categorical variable. Three different groupings were considered for the x-axis, the amplification protocol (left pane), the elongation time (middle pane) and the sample composition (right pane).

<p>Figure 48</p>	<p>Error rate by position annotated by quality score</p>	<p>For each position where the input templates were identical to each other, the aggregate error rate was computed across all datasets of a given amplification protocol using the array of accuracy and quality scores described in section 4.1.3. For a position, the total number of times a sequence did not match the input template at the position was tallied. This tally was divided by the total number of sequences in all the datasets of the given amplification protocol to obtain the error rate. The average quality score was also computed for each position for the given amplification protocol by summing all the quality scores associated with a given position and dividing it by the number of sequences. A scatter plot was then produced with the error rate on the y-axis and the position on the x-axis. The points were colored with the average quality score computed for the position.</p>
<p>Table 25</p>	<p>Error rate by categorized quality scores</p>	<p>Using the arrays of accuracy and quality scores described in section 4.1.3, the aggregate error rates were computed for positions in all sequences with quality scores that fall into specified bands. To populate the cell corresponding to the mxPCR protocol and the quality range (20,30], the number of times a base at a position where the input templates were identical to each other in any datasets that was amplified using the mxPCR protocol that also had a quality score greater than 20 and smaller than 31 mismatched the input template was divided by the total number of positions in all the datasets that were amplified with the mxPCR protocol where that position also had a quality score greater than 20 and small than 31. This was performed for the raw sequences only.</p>
<p>Figure 49</p>	<p>Error rate by quality score</p>	<p>Using a process like the one described for Table 25, error rates were computed per quality score (uncategorized) for all datasets that were produced with a given amplification protocol. A scatterplot was produced with quality score on the x-axis and the error rate on the y-axis with separate panes for amplification protocol and type of sequence (consensus or raw)</p>



Figure 50	Consensus sequence accuracy by bin size	Each consensus sequence was compared to the three input templates and flagged as either matching one of the input templates perfectly or not matching any of the input templates perfectly. The percentage of the consensus sequences which were all produced from bins of the same size that matched perfectly was computed for all bin sizes up to bins of size 100. A scatterplot was produced with the bin size on the x-axis and the percentage of error-free (perfectly matching) consensus sequences on the y-axis. Datasets were grouped based on amplification protocol and the number of consensus sequences resulting from the given bin size was used to color the points.
Figure 51	 <p>Prevalence estimates</p>	As described in section 4.1.4, each raw and consensus sequence was compared to the input templates and a set of all probable chimeras resulting from PCR recombination between the input templates. Each reads was then flagged as either having been sequenced from one of the three input templates or as having been sequenced from a chimeric molecule. Using only those sequences that was flagged as resulting from non-chimeric molecules, prevalences can be computed by dividing the number of sequences flagged as resulting from a non-chimeric molecule of a specific input template with the total number of molecules in the relevant individual dataset that were flagged as having been sequenced from non-chimeric molecules. A scatterplot was made from this data with separate panes so that each pane contains a single combination of sample composition and input variant. The x-axis contains the elongation time as a categorical variable with a small amount of jitter added to prevent the points from overlapping and prevalence on the y-axis. This figure shows only prevalences estimated from raw sequences.
Figure 52	Prevalence estimates	As described in the entry for Figure 51 in this table, prevalences were computed for the individual input variants for each individual dataset using the consensus sequences only. This data was used to produce a

		scatterplot with the same arrangement was the one in Figure 51, except using consensus sequences instead of raw sequences.
Figure 53	Prevalence estimates	As described in the entry for Figure 51 and Figure 52 in this table, prevalences were computed for the individual input variants for each individual dataset using the consensus sequences only. This data was used to produce a scatterplot that with an arrangement that only differ from that in Figure 51 and Figure 52 by what is placed on the x-axis. This figure has the amplification protocol on the x-axis instead of the elongation time. This figure shows only prevalences estimated from raw sequences.
Figure 54	Prevalence estimates	As described in the entry for Figure 53 in this table, prevalences were computed for the individual input variants for each individual dataset using the consensus sequences only. This data was used to produce a scatterplot with the same arrangement was the one in Figure 53, except using consensus sequences instead of raw sequences.
Figure 55	Distribution of prevalence estimates	Using the prevalences estimated on the individual datasets as described in the entry for Figure 51 in this table, a box and whisker plot was used to show the distribution of the prevalence estimates (y-axis) for each variant (x-axis) in separate panes so that each pane contains a unique combination of sequence type (consensus or raw) and sample composition.
Figure 56	Variability of prevalence estimates	Using the prevalences estimated on the individual datasets as described in the entry for Figure 51 in this table, the datasets were grouped so that all datasets that are composed of sequences from samples with the same composition fall into the same group. This results in eight observations per group since eight different amplification protocols was applied to each sample. Two standard deviations are computed for the prevalence estimates of each variant for each such grouping of dataset, one standard deviation based on the raw sequences and one based on the consensus sequences. This

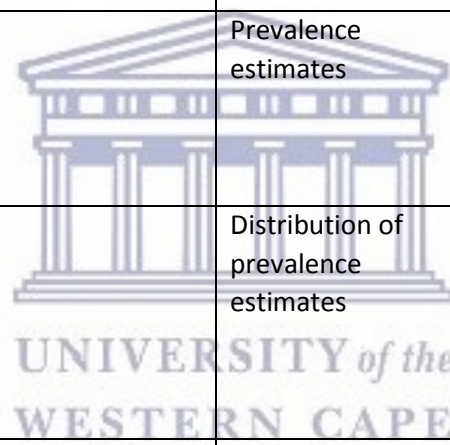
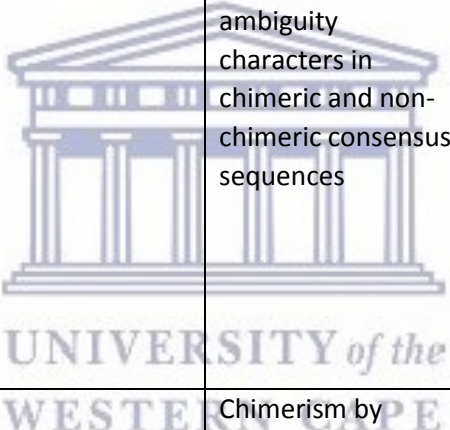
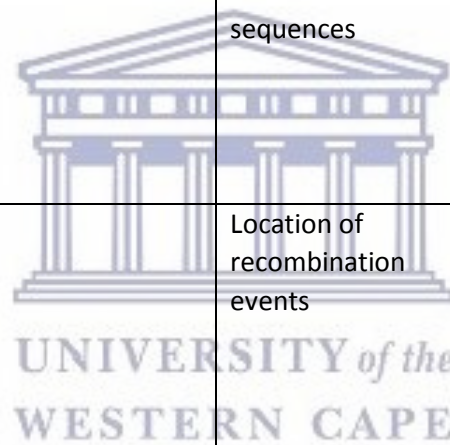
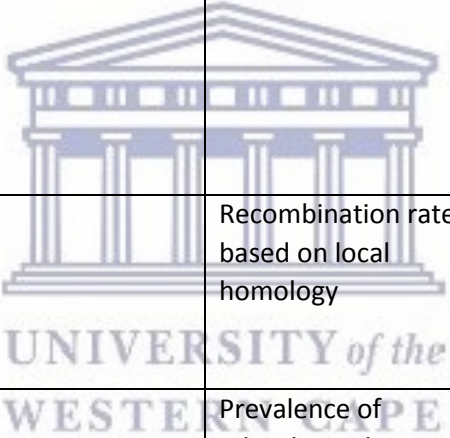


Figure 57	Rates of recombination	The percentage of sequences that were identified as having been sequences from a molecule that resulted from a recombination event between two different input templates was computed for each individual dataset using the process described in section 4.1.4. A scatterplot was produced from this dataset with the percentage of sequences flagged as chimeric on the y-axis and the sample composition on the x-axis. The two panes separate percentages computed based on raw sequences from percentages computed based on consensus sequences.
Figure 58	 <p>Occurrence of ambiguity characters in chimeric and non-chimeric consensus sequences</p>	In general, consensus sequences that contain ambiguity characters were removed from the analysis as stated in section 4.1.2. Here, such sequences were included and the frequency of consensus sequence that include ambiguity characters was computed based on datasets grouped so that all sequences produced from a sample with a given composition, elongation time and chimerism status was grouped together. A scatterplot was produced from these frequencies with the percentage of consensus sequences that contain ambiguity characters on the y-axis, the sample composition on the x-axis and using the chimerism status to color the point and the elongation time to select which object to represent the point with.
Figure 59	Chimerism by prevalence of the variants	Each sequence that was flagged as a sequence that was sequences from a molecule that resulted from a recombination event between two different input templates was in reality assigned to a specific probable chimeric sequence that could result between the given input templates. Thus for each sequence in this study that was flagged as chimeric is also described as matching variant X on the 5' end, then a recombination event occurring between positions Y and Z so that the 3' end after position Z matches variant K. Thus all sequences that matches a given variant at the 3' end can be grouped into a single group (including those sequences that were not chimeric, i.e. they are a representation of that variant possibly having some

		<p>sequencing errors). Using this grouping of all sequences that match a given variant on the 3' end, the percentage of them that are chimeric can be computed. A scatterplot was produced from these percentages with the prevalence of the variant that match the group of sequences on the 3' end on the x-axis and the percentage of the sequences in the group that were flagged as chimeric on the y-axis. The color and shape of the object used to represent the data point was chosen based on the amplification protocol.</p>
<p>Figure 60</p>	<p>Chimerism and raw and consensus sequences</p>	<p>Using the classification of sequences into chimeric and non-chimeric sequences described in section 4.1.4, the percentage of raw sequence that are chimeric and the percentage of consensus sequence that are chimeric is known for each dataset. A scatterplot was produced using these percentages. The composition of the sample was used to select the object to represent the data point and this object was colored based on the protocol with which the sample was amplified.</p>
<p>Figure 61</p>	<p>Location of recombination events</p>	<p>Each sequence that was flagged as a sequence that was sequences from a molecule that resulted from a recombination event between two different input templates was in reality assigned to a specific probable chimeric sequence that could result between the given input templates. Thus for each sequence in this study that was flagged as chimeric is also described as matching variant X on the 5' end, then a recombination event occurring between positions Y and Z so that the 3' end after position Z matches variant K. This means that each recombination event identified in the dataset can be assigned to an interval of positions on the sequence at which it could have occurred. Since the differences between the input templates uniquely determines the intervals to which the events can be assigned, for each given recombination composition, i.e. variant X on the 5' end and variant K on the 3' end, there is a fixed set of possible intervals in which the recombination event must have occurred. This figure presents the frequency which the recombination events occurred in</p>



		<p>the different intervals. The intervals differ in width and the decision was made to scale the heights of the bars so that the total area of the bar represent the percentage of the recombination events that occurred on chimeras of the given characterization occurred in the corresponding interval. Separate panes are used to present the possible characterizations of chimeric reads as well as the different amplification protocols. A local homology measure for a given recombination interval can be defined as the width of the interval plus the widths of the two intervals surrounding it. This measure can be interpreted as the distance that must be moved away from the current interval in order to encounter another two mismatches between the two input templates between which the recombination event must have occurred in order to produce the current chimeric read. The bars in the plot are colored based on this local homology measure.</p>
Figure 62	 <p>Recombination rate based on local homology</p>	<p>Using the data prepared to produce Figure 61, a scatterplot was made relating the local homology measure to the probability that a recombination event will occur in the interval associated with the value of the homology measure. Trend line was added using the loess function from base R without altering any of the defaults.</p>
Figure 63	<p>Prevalence of minority variants</p>	<p>There are twelve positions at which none one of the three input templates matched any other. At each of these positions, each sequence can be precisely characterized as representing a single specific variant. Selecting only those variants and a corresponding sample compositions such that variants occur at prevalences below 20%, the prevalence of these minority variants are plotted for the twelve positions where the reads can be precisely assigned to one of the input templates. Using a line plot with the prevalence of the variant as computed from the each of the twelve points on the y-axis and the position (as a categorical variable) on the x-axis the change in the prevalence of minority variants along the positions of the</p>

		sequences are demonstrated. Separate panes are used to that each pare represents a unique combination of sequence type, elongation time and amplification protocol. The color of the line indicates the variant and the sample composition.
Figure 64	Prevalence of rare minority variants	This plot is identical to Figure 63 except that the variants and sample compositions were chosen so that all variants occur at prevalences below 3%.
Table 29	Detailed investigation of a specific minority mutation	A detailed investigation was performed for a number of infrequently occurring bases at a five positions in the datasets based on the sample composition labelled MAJ_99 that were prepared using either the 2dPCR amplification protocol with short elongation times or using the rcPCR amplification protocol with short elongation times. Each row in the table was constructed with a similar approach and the row corresponding with the occurrence of an A at position 36 in the dataset prepared with the 2dPCR amplification protocol will be used to illustrate this process. There were a total of 5078 bin in this dataset. Each bin was investigated and the probability that the number of As observed at position 36 are explained by sequencing error based on a template that does not contain an A at position 36 was computed. For example in a bin of size 10 if the base in the template is not an A, and there is one A in the bin, then the probability of that A resulting from sequencing error is computed from a binomial distribution with size 10, a single success and the probability of success equal to the probability of a sequencing error resulting specifically in an A. This table describes the distribution of these probabilities by listing the number of bins in which the likelihood of the observed number of As arising due to sequencing error alone is below given cutoffs listed in the top row of the table. In all bins, the probability of the number of As arising due to sequencing error alone is smaller than 1 (10^0). In 3626 bins this probability was



		smaller than 0.1 (10^{-1}) and so forth. This process was repeated for all the listed letters, positions and probability cutoffs.
Table 30	Detailed investigation of a specific minority mutation	The generation of this table is identical to that of Table 29, which was already described in this table. The only difference is that a different set of positions were chosen.



8.4 hypermutR installation instructions for Ubuntu

Make sure you have a recent version of R. Follow the instructions in the following link to set up the correct repository for apt. <http://stackoverflow.com/questions/10476713/how-to-upgrade-r-in-ubuntu>.

Make sure that both r-base and r-base-dev is installed:

```
sudo apt-get install r-base r-base-dev
```

Next, install devtools' dependencies with apt-get:

```
sudo apt-get install libssl-dev libxml2-dev libcurl4-gnutls-dev
```

Then, from within R, install devtools:

```
install.packages('devtools', repo = 'http://cran.rstudio.com/')
```

Finally, install hypermutR. From a local file:

```
library(devtools)
```

```
install_local('/path/to/file/hypermutR_x.y.z.tar.gz')
```

Please note that you must use `install_local` from devtools - `install.packages` will not work. Change `/path/to/file` to the path to the installation file on your computer and `x.y.z` to match the installation file you have.



Or using the bit_bucket repo:

```
library(devtools)
```

```
install_bitbucket('hivdiversity/hypermutR', auth_user = 'username',  
password = 'password')
```

Finally, hypermutR includes a script that can be run from the commandline. You need to put this script somewhere convenient (`/usr/bin` for example)

```
file.symlink(from = file.path(find.package('hypermutR'), 'hypermutR.R'), to  
= '/usr/bin')
```

8.5 hypermutR usage instructions

From within an R session: Within R

```
library(hypermurR)
```

```
help('remove_hypermurR')
```

This will display the help for the main function in hypermutR.

From the command line

```
hypermurR -h
```

or (depending on your installation):

```
hypermurR.R -h
```

This will display help for all the options and an example call to hypermutR.



8.6 Patching PhyML

The pipeline requires a patched version of PhyML. To apply this patch, clone PhyML's git repository from github (<https://github.com/stephaneguindon/phyml>) and make the following edit to the `utilities.c` file in the `src/` folder. In the `Dist_And_BioNJ` function, replace these two lines:

```
tree      = mat->tree;
tree->mat = mat;
```

with the following three lines:

```
tree      = mat->tree;
tree->mat = mat;
Print_Mat(mat);
```

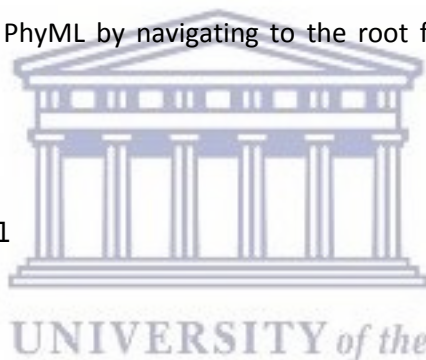
After making this edit, compile PhyML by navigating to the root folder of `phyml` and issuing the commands:

```
sh ./autogen.sh
```

```
./configure --enable-phyml
```

```
make
```

This will generate an executable binary file called `phyml` in the `src/` subfolder of `phyml`. Running `phyml` from this new binary file will result in the distance matrix being printed to `STDOUT` during the execution of the program. At the time of writing, this patch worked on commit number `8eb35001287ab083762aad1d9e68dcc462fdad1f`.





UNIVERSITY *of the*
WESTERN CAPE