Analysis of complex genetic variation using population reference graphs



Sorina Maciuca

Lincoln College

Wellcome Trust Centre for Human Genetics

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Michaelmas 2017

This thesis is dedicated to
my parents,
for paving the way for me to get here

Acknowledgements

I would like to thank my supervisors, Zam Iqbal and Gil McVean, for their infinite patience and insightful comments over the last 4 years. More importantly, I'm very grateful to them for initiating me to scientific research and offering me the opportunity to work on a truly exciting, intellectually stimulating project. Without them, my doctoral journey would have been much more difficult.

I enjoyed collaborating with Robyn Ffrancon, Carlos del Ojo Elias and Jerome Kelleher and I thank them for helping me with various aspects of the software development process.

I also appreciate the collaboration with Dominic Kwiatkowski's group and I thank him for scientific input in matters regarding the Pf3k project. From his group, I am particularly grateful to Jacob, for both work-related advice and endless counselling.

I owe thanks to Phelim and Rachel for many fruitful discussions over lunch and coffee; and to Gil's group for many interesting presentations in group meetings, that introduced me to new scientific topics.

I would like to thank other colleagues for providing entertaining company, making it easy to forget I'm at work: Zammy, Rory, Kathryn, Clare and Holly. In particular Zammy, who kept me well-fed during this tiring journey.

I would also like to acknowledge all my other closed ones, without whom Oxford would not be the same: Clément, Madhana, Scott, Vlad, Stefan, Bayar and Andrei. In particular to Clément, for support and understanding during the difficult final months.

Last, but not least, I thank my parents for encouragement and my grandmother, Elena.

Statement of Originality

The work described in this thesis was entirely carried out by the author, as part of the DPhil in Genomic Medicine and Statistics course at the University of Oxford, under the supervision of Dr. Zamin Iqbal and Prof. Gil McVean. The thesis is the product of the author's own work, except collaborative pieces that are explicitly specified in text.

Sorina Maciuca
November 2017

Abstract

In this thesis, I study the problem of genome inference from short-read DNA sequencing data, with the goal of accurate characterisation of genomic regions with high sequence diversity. I describe a set of novel algorithms based on a generalised reference genome that captures sequence variation within a species. In Chapter 3, I propose a novel data structure that extends the traditional reference genome with known variants, providing a compressed representation of genetic diversity. I present algorithms to match sequencing reads to this extended reference structure and infer a personalised reference genome within close genetic distance from the sample under analysis. Coupled with existing variant calling tools, this personalised reference confers increased power to detect complex variants in diverse regions, compared to the traditional reference genome. In Chapter 4, I evaluate the performance of the method on simulated data and show that it is viable for megabase-sized genomes such as the malaria parasite *Plasmodium falciparum* – a typical sample can be analysed in 5.7h on a single CPU, using a small amount of memory. I suggest a number of future optimisations to improve computational efficiency.

In Chapter 5, I apply my method to 1300 *Plasmodium falciparum* samples from across the world and study two hyper-diverse genes that encode surface antigens in *Plasmodium falciparum*. I show that the personalised references recover variants of these genes that are missed by standard techniques of mapping reads to the traditional reference genome. Next, I build the first global variation catalogue incorporating dimorphic alleles of a region of functional interest and study their frequency patterns.

Contents

1 Introduction **1**

2 Background **7**

2.1 Brief overview of diversification mechanisms 8

2.2 Complex variation and its biological significance 10

2.3 High-throughput DNA sequencing 16

2.4 Classic HTS data analysis 17

2.4.1 *De novo* assembly 17

2.4.2 Read mapping 19

2.4.3 Variant calling 22

2.5 Transition to graph references 23

2.6 *Plasmodium falciparum* genome 33

2.7 Thesis overview 34

3 Gramtools: Genome inference from a population reference graph **36**

3.1 Introduction 36

3.2 Publication and collaboration notes 37

3.3 Terminology 37

3.4 Background 38

3.4.1 Notations 38

3.4.2 Bit vectors, *rank* and *select* operations 39

3.4.3 Burrows-Wheeler Transform 39

3.4.4 Suffix Arrays 40

3.4.5 Backward search with an FM-index 41

3.4.6 Wavelet Trees 45

3.5 `gramtools` pipeline overview 47

3.6 vBWT data structure 48

3.7 Read quasimapping 51

3.7.1 Variation-aware exact matching . . . 51
3.7.2 *k*-mer indexing . . . 55
3.8 Personalised reference inference . . . 55
3.9 Remapping to graph-inferred reference and variant discovery . . . 58
3.10 Implementation . . . 60
3.10.1 Software design . . . 60
3.10.2 Implementation details . . . 61
3.11 Limitations . . . 62

4 Gramtools performance and optimisations 65
4.1 Introduction . . . 65
4.2 Collaboration note . . . 65
4.3 Initial performance measures . . . 66
4.3.1 Graph and index construction . . . 66
4.3.1.1 Whole-genome *Plasmodium falciparum* . . . 67
4.3.1.2 Whole-genome human . . . 68
4.3.2 Quasimapping runtime . . . 71
4.4 Quasimapping speed optimizations . . . 73
4.4.1 Ranks storage . . . 73
4.4.2 BWT bit mask . . . 75
4.4.3 *k*-mer indexing . . . 77
4.4.3.1 *Plasmodium falciparum* genome . . . 78
4.4.3.2 Controlling for genome content: *P.falciparum*-sized PRG on human genome sequence . . . 81
4.5 Final performance measures . . . 83
4.5.1 Whole-genome *Plasmodium falciparum* . . . 83
4.5.2 Human control genome . . . 83
4.6 Impact of graph structure on performance . . . 84
4.7 Scaling . . . 88
4.8 Conclusions . . . 90

5 Malarial surface antigens application: *DBLMSP* and *DBLMSP2* 94
5.1 Introduction . . . 94
5.2 *DBLMSP* and *DBLMSP2* allelic dimorphism . . . 97
5.3 PRG construction from Pf3k reference panel . . . 103
5.4 *DBLMSP* sequence inference for clonal samples . . . 107

5.4.1 PacBio assemblies validate the personalised references inferred by `gramtools` in *DBLMSP* . . . 108
5.4.2 Measures across pileups in field samples . . . 111
5.4.2.1 Initial measures show read pileups are closer to the personalised references than to 3D7 . . . 111
5.4.2.2 Discovery of 2 novel haplotypes . . . 112
5.4.2.3 Updating the reference graph with 2 novel haplotypes further improves sequence inference . . . 113
5.5 Population frequencies . . . 114
5.6 Impact of size of reference panel on inference . . . 115
5.7 *DBLMSP* and *DBLMSP2* joint analysis . . . 119
5.7.1 Motivation . . . 119
5.7.2 PRG construction . . . 119
5.7.3 Sequence inference for clonal samples . . . 121
5.7.3.1 PacBio assemblies validate the personalised references inferred by `gramtools` in *DBLMSP* and *DBLMSP2* . 121
5.7.3.2 Read pileups are closer to the personalised references than to 3D7 in *DBLMSP* and *DBLMSP2* . . . 122
5.7.4 Gene conversion between *DBLMSP* and *DBLMSP2* . . . 124
5.7.5 Population frequencies . . . 125
5.7.5.1 Allele frequencies . . . 125
5.7.5.2 Linkage disequilibrium between allelic classes of *DBLMSP* and *DBLMSP2* . . . 129
5.8 Summary of results . . . 138
5.9 Discussion . . . 139

6 Final discussion **142**
6.1 Contributions . . . 142
6.2 Limitations . . . 144
6.3 Future applications for `gramtools` . . . 145
6.4 Challenges and future directions for reference genome graphs . . . 148

A Supplementary figures **150**

B Methods **167**
B.1 Graph input 167
B.1.1 Initial *DBLMSP*-only graph 167
B.1.2 Joint *DBLMSP* and *DBLMSP2* graph 168
B.2 Construction of reference graph 169
B.3 Defining the edit distance between a read pileup and a reference sequence 174
B.4 Discovery of novel contigs 175

Bibliography **179**

List of Figures

1.1 IGV screenshot of a read pileup on top of the standard *Plasmodium falciparum* reference genome in *MSP1* gene. Large segments of the gene have no read coverage, rendering the corresponding sequence inaccessible. 4

2.1 Example of complex variant with four different representations from four different mappers, which can cause data sets to appear to call different variants when in reality they are the same variant. In this case, the six bases CAGTGA are replaced by the five bases TCTCT at location 114841792-114841797 on chromosome 1 of the human genome. The four sets of reads are from sample NA12878, sequenced with different technologies, as follows: Illumina mapped with `BWA`, 454 mapped with `ssaha2`, Complete Genomics mapped with `CGTools`, and Illumina mapped with `Novoalign`. Source: [Zook et al., 2014] 15

2.2 De Bruijn graph of order 7 for the sequence **ATGGAAGTCGATGGAAG**. First, all adjacent 7-mers overlapping by 6 bases are extracted from the sequence. Unique 7-mers label the vertices of the graph and edges connect adjacent 7-mers. Source: `http://www.homolog.us/Tutorials/index.php?p=2.1&s=1` 18

2.3 A summary of required operations in HTS analysis pipelines after replacing the standard reference genome with a structure of multiple references. Source: [The Computational Pan-Genomics Consortium, 2016] 25

2.4 Example of the POA representation. (a) Pairwise protein sequence alignment. (b) A single sequence (top in (a)) in PO-MSA format. (c) The sequences in (a), in PO-MSA format, aligned to each other. Dashed circles indicate two nodes are aligned. (d) PO-MSA representation of the pairwise alignment in (a). Dashed circles indicate two nodes are aligned. Source: [Lee et al., 2002] 27

2.5 Reference multi-genome constructed by superimposing three sample genomes with BWBBLE. SNPs are encoded in the reference branch as additional characters, with **A**–>**C** and **T**–>**G** being represented by **M** and **K** respectively. Indels are appended at the end of the reference branch after some padding. In this example, the length of the padded sequence corresponds to reads of length 4. Adapted from: [Huang et al., 2013] 29

2.6 Description of PRG for the MHC region. (a) Multiple sources of information about genetic variation, including alternative reference haplotypes (lines), classical HLA alleles (rectangles), and SNPs or short indels (triangles), are aligned. Colors indicate divergent sequence; dashes indicate gaps. (b) A PRG is constructed from the alignment, resulting in a generative model for variation within the region. SNPs, indicated by diamonds, are added as alternative paths to all valid backgrounds (excluding sequence with gaps or a third allele at the position). (c) The PRG is compared to the de Bruijn graph constructed from reads obtained from a sample. Informative k-mers (those that are found at only one level in the PRG) are identified (dark blue). k-mers found elsewhere in the genome (yellow) are ignored. (d) An HMM is used to infer the most likely pair of paths through the PRG, allowing for read errors, resulting in an individualized reference chromotype for the sample. (e) Two haploid genomes are constructed from the reference chromotype, with arbitrary phasing between adjacent bubbles, and reads (light blue lines) from the sample are aligned and assigned (on the basis of mapping quality) to a reference, thus identifying places where the sample contains new variation (red circles; only one path through the chromotype is shown). (f) Newly discovered variants modify the reference chromotype, resulting in the inferred chromotype for the sample. Source: [Dilthey et al., 2015] 32

3.1 Example of BWT construction for the string $T =$ **GATAGACA**. First, take all rotations of the string, i.e. repeatedly move its last letter to the front. Then the string rotations are sorted and the last column of the obtained matrix is the BWT. 40

3.2 Suffix array for the example string $T =$ **GATAGACA** and its relation to the BWT. 41

3.3 Backward search example for looking up the substring P = **AGA** in text T = **GATAGACA**. Matching the first character from the end of P, **A**. For simplicity, the entire Burrows-Wheeler matrix is shown, but only its first and last column need to be stored. The numbers in the first column represent the ranks of characters in the column. The numbers in the last column represent the ranks of characters in the original text T. 43
3.4 Backward search example for looking up the substring P = **AGA** in text T = **GATAGACA**. Matching the second character from the end of P and extending to **GA**. The numbers represent the ranks of characters in the first and last columns respectively. 44
3.5 Backward search example for looking up the substring P = **AGA** in text T = **GATAGACA**. Matching the third character from the end of P and extending to **AGA**. 44
3.6 Wavelet tree encoding of a string that is the same as the BWT from Fig. 3.9 below. Calculating the *rank* of the marked **A** is performed by repeated *rank* calls moving down the binary tree until the alphabet remaining is just 2 characters. Note that only the bit vectors are stored in the tree, the corresponding strings are only shown here for clarity. 46
3.7 A simple PRG linearised according to our encoding. The first site has 3 alleles, which do not here look at all similar, and the second is a SNP. Note that the reference sequences do not need to be aligned. This example will be used in subsequent figures. Source: [Maciuca et al., 2016] 49
3.8 PRG graph structure. The sequences shown in a) could be represented either as 3 separate mutations (shown in b)), or enumerated as 8 small haplotypes, shown in c). Both are supported by our encoding. Similarly, the sequences in d) could be represented in our implementation as shown in e). However, we do not support nesting of alleles, as shown in f). Source: [Maciuca et al., 2016] 50

3.9 Backward search across the vBWT of the linear PRG in Figure 3.7. We start at the right-hand end of the read **GTTATTTAC**, with the character **C**, and as we extend we hit the character **5**, signalling the start or end of a variation site. We check the suffix array to get the coordinate in the linear PRG, and find it is the end. Therefore, the read must now continue into one of the alleles, signalled by the symbol **6**. Continuing in this manner (the shorter arrows signify multiple intermediate steps not shown) we are able to align across the site. Source: [Maciuca et al., 2016] 52

3.10 Toy example of a PRG and the different ways a read can quasimap to it and contribute to the coverage scoring of alternative alleles. 57

3.11 Schematic figures of how novel variants on top of the PRG (marked with an $\times$) can affect inference in the context of reads containing the variants failing to quasimap. On the left, toy examples of read mappings to the PRG; on the right, toy examples of resulting read pileups over the PRG-inferred reference. In a), there is a single SNP on top of an alternative allele. This produces a coverage loss, but the correct allele is likely to be selected based on quasimapped reads surrounding the SNP, enabling its discovery in the final step of full mapping. In b), there are 2 novel SNPs, separated by a distance smaller than the read length. No reads will quasimap in the region spanned by the SNPs within the alternative alleles. Hence, any of them could be selected arbitrarily based on reads quasimapping in the margins delimited by the SNPs. This can produce a gap in the pileup with respect to the inferred reference. In c), there are 2 novel SNPs separated by a distance larger than the read length. There will be a coverage loss in the region spanned by the SNPs, but some reads will quasimap in between, informing the choice of a close allele. The SNPs can then be discovered in the step of remapping to the personalised reference. 59

3.12 Diagram of the data flow through the `gramtools` pipeline. 61

4.1 Time and memory requirements to construct the FM-index for the *Plasmodium falciparum* PRG. Coloured blocks represent parsing the linear PRG (green), constructing the suffix array (red), BWT (purple), wavelet tree (pink), sampling from the suffix array (grey), sampling from the inverse suffix array (olive). 67

4.2 Time and memory requirements to construct the FM-index for the human PRG with common variants from 1000 Genomes Project. Coloured blocks represent parsing the linear PRG (green), constructing the suffix array (red), BWT (purple), wavelet tree (pink), sampling from the suffix array (grey), sampling from the inverse suffix array (olive). . . . 69
4.3 Time and memory requirements to construct the FM-index for the human PRG with full 1000 Genomes variant set. Coloured blocks represent parsing the linear PRG (green), constructing the suffix array (red), BWT (purple), wavelet tree (pink), sampling from the suffix array (grey), sampling from the inverse suffix array (olive). 71
4.4 Median (in red) and range of the number of suffix array intervals generated per backward search iteration (or per character matched) during quasimapping simulated reads to the *Plasmodium falciparum* PRG (across 10,000 reads of 150-bp length). Only the first 75 iterations are shown. 80
4.5 Median (in red) and range of the number of suffix array intervals generated per backward search iteration (or per character matched) during quasimapping simulated reads to the human control PRG (across 10,000 reads of 150-bp length). Only the first 75 iterations are shown. 82
4.6 PRG graph structure options. The sequences shown in **a)** could be represented either as 3 separate mutations, shown in **b)**, or enumerated as 8 small haplotypes, shown in **c)**. `gramtools` supports both options. 85

5.1 Worldwide prevalence of *P. falciparum* antigen haplotypes that are components of malaria vaccines. Source: [Barry et al., 2009] 96
5.2 25 genes with ≥10 SNPs showing highest values of Tajimas D index in a genome-wide analysis of sequences of 65 Gambian *P. falciparum* isolates. Top balancing selection signal is in PF10_0355, an alternative name for the *DBLMSP2* gene. PF10_0348 is gene *DBLMSP*. Source: [Amambua-Ngwa et al., 2012] 98

5.3 Variation in nucleotide diversity over the core genome of *P. falciparum*. The peak of nucleotide diversity on Chromosome 10 is expanded to show four distinct peaks due to genes encoding merozoite surface antigens *MSP3*, *MSP6*, *DBLMSP*, and *DBLMSP2*. All labeled loci (with the exception of *AMA1*) are sites of complex variation where read mapping often fails due to highly diverged alleles from the reference sequence. For these loci, *de novo* assembly of sequence reads is required to determine the nonreference alleles. Source: [Miles et al., 2016] . . . 100

5.4 Genome sequencing of 434 *P. falciparum* isolates reveals *DBLMSP* and *DBLMSP2* are dimorphic and divergent. **(a)** Cluster dendrogram of the 15 *DBLMSP* haplotypes (left) and 41 *DBLMSP2* haplotypes (right) identified in the study. Both trees were obtained using `ClustalW` [Sievers et al., 2011]. **(b)** A schematic representation of the amino acid multiple sequence alignment of six representative exemplar haplotypes for the DBL domain of *DBLMSP* between residues Cys144 and Arg344, and the *P. reichenowi* ortholog. Note the long non-recombining region has two alternate allelic forms (orange and yellow), of which the yellow is relatively close to the *P. reichenowi* sequence (shown in brown). A phylogenetic tree indicating the relationships of the sequences built by clustering haplotypes based on a distance matrix (where distance corresponds to the number of SNP and indel differences) is shown on the left. The alleles similar to 3D7 are labelled class 1 and the ones belonging to the dissimilar clade are labelled class 2. **(c)** A schematic representation of the amino acid multiple sequence alignment of 10 representative exemplar haplotypes for the DBL domain of *DBLMSP2* between residues Glu102 and Gly550. Note the long non-recombining region has two alternate allelic forms (orange for class 1 and purple/dark green for subclasses 2.1/2.2, respectively). A phylogenetic tree indicating the relationships of the sequences built by clustering haplotypes based on a distance matrix (where distance corresponds to the number of SNP and indel differences) is shown on the left. Source: [Crosnier et al., 2016] 101

5.5 Schematic showing workflow for variant detection from a reference graph. **(a)** Firstly, current information about the variation catalogue, pooled from PacBio assemblies, long variant calls from `Cortex` and SNPs or short indels called by `GATK`, is integrated via multiple sequence alignments at variable loci. The figure depicts the DBL domain locus and surrounding SNPs. Colours represent different haplotypes, with the red spectrum representing the 3D7-like dimorphic class and the blue spectrum representing the other class. **(b)** Secondly, a variant site in the reference graph is created by listing observed alternative sequences from the alignment and their mosaics, representing recombinants, onto distinct edges. Adjacent variant sites are joined in the graph by an edge representing the non-variable sequence in between them. In the DBL domain, recombinants are allowed within dimorphic class (red/red, blue/blue) but not between classes (red/blue) – refer to Methods, Section B.2 for details. This graph will act as a generative model for sequences. **(c)** Thirdly, reads are aligned to the graph and coverage counts on alternative alleles are collected. **(d)** Finally, a sequence path consisting of alleles with highest read support is extracted from the reference graph and used as a personalised reference for the sample under analysis. Novel variants that were not present in the catalogue used to construct the graph can be called on top of this personalised reference with standard tools. 105

5.6 Bar plot outlining the reference graph structure. Each bar represents a variable site within the *DBLMSP* gene, plotted at its genomic coordinates, with height proportional to the logarithm (base 2) of the number of alternative sequences and width proportional to their length. The *log* scale is used for the y-axis because the site containing the dimorphic region of the DBL domain consists of 3,383 alternative haplotypes. . . 106

5.7 Validation strain GN01 has an inter-class recombinant in the dimorphic region of the DBL domain. **(a)** IGV screenshots of pileups with respect to the 3D7 reference (top) and the personalised reference (bottom) show a gap in the DBL domain, suggesting that the GN01 sequence is divergent from 3D7, but also from the haplotypes in the reference graph. Note that the appearance of reduced depth of coverage in the bottom pileup is a consequence of the IGV scaling. In fact, the raw values are the same. **(b)** Pairwise alignment between 3D7 and the GN01 sequence obtained from the PacBio assembly shows a section similar to 3D7 (red) and a divergent section (blue) in the dimorphic region. 109

5.8 IGV screenshots of read pileups in the *DBLMSP* region for 4/5 validation samples. For 7G8, the pileup with respect to 3D7 has a gap of 476 nucleotides in the DBL domain, whereas the pileup with respect to the graph-inferred reference shows even coverage and 9/9 SNPs removed. For GB4, the pileup with respect to 3D7 presents a dip in coverage between coordinates 1415015-1415030 (circled), which gets smoothed out in the pileup with respect to the graph-inferred reference; in addition 7/8 SNPs are removed. For KH02 and KE01, the pileups with respect to 3D7 do not show any gaps, but show clustered SNPs which could produce alignment problems; the pileup with respect to the graph-inferred reference shows 18/20 SNPs removed for KH02 and 6/6 for KE01. Note that in these last 2 samples, the appearance of reduced depth of coverage in the pileups on the right compared to the ones on the left, is a consequence of the IGV scaling. In fact, the raw values are the same. Enlarged versions of each pileup can be found in Appendix Figs. A.3 - A.10. 110

5.9 Histograms of measures calculated from pileups with respect to the 3D7 reference (orange) and the personalised references (green): number of gaps (left) and number of mismatches (right). Both gaps and mismatches are reduced when mapping to the personalised reference compared to 3D7. 111

5.10 Genomic coordinates of regions without read support when mapping to the personalised references in samples where the DBLMSP sequence cannot be fully resolved. The majority of problematic samples (including the GN01 strain used for validation) contains a gap in the DBL domain between coordinates 1413958 and 1414100. We identified 2 samples where `Cortex` had assembled contigs spanning this gap. . . . 113
5.11 Histograms of gaps and mismatches calculated from pileups with respect to the 3D7 reference and the personalised references inferred from the graph updated with 2 additional haplotypes discovered from raw `Cortex` contigs. Inference is further improved, with the personalised references closing the gaps produced by mapping to 3D7 in 132 additional samples compared to the ones resolved by the previous reference graph in Fig. 5.9. 114
5.12 Population level frequencies of haplotypes in the DBL domain of the gene. The phylogenetic tree on the bottom shows they come from two deeply diverged lineages that are preserved across populations, suggesting they are under balancing selection. `ClustalW` was used to build the phylogenetic tree. 116
5.13 Measures calculated from pileups with respect to personalised references inferred from graphs built out of 1, 2, 10, 50 and 100 samples respectively. The top plot shows the number of mismatches gradually decreases as more samples are added to the reference graph, with one exception: the graph consisting of 100 samples performs worse than the one consisting of 50 samples. Mean number of mismatches in each plot, from left to right, is 26.7, 6.3, 2.9, 1.5 and 1.9 respectively. The bottom plot shows the biggest improvement in number of gaps is observed when extending 3D7 with one extra sample containing a class 2 haplotype in the DBL domain, as expected from the dimorphic structure. Mean number of gaps in each plot, from left to right, is 154, 70, 72.2, 71.7 and 71.7 respectively. 118
5.14 Bar plot outlining the reference graph structure in *DBLMSP* (top) and *DBLMSP2* (bottom). Each bar represents a variable site plotted at its genomic coordinates, with height proportional to the number of alternative sequences and width proportional to their length. 120

5.15 Histograms of measures calculated from pileups in *DBLMSP* with respect to the 3D7 reference and the personalised references inferred from the joint graph: number of gaps (left) and number of mismatches (right). Results are consistent with the ones obtained from the single gene graph. 123
5.16 Histograms of measures calculated from pileups in *DBLMSP2* with respect to the 3D7 reference and the personalised references inferred from the joint graph: number of gaps (left) and number of mismatches (right). In the majority of samples, both gaps and mismatches are reduced to 0 when mapping to the personalised references. 123
5.17 Phylogenetic tree of all fully resolved alleles from the dimorphic region of the DBL domains of *DBLMSP* and *DBLMSP2*, obtained from 1092 samples. Alleles with the "m4" prefix come from *DBLMSP* and alleles with the "m8" prefix come from *DBLMSP2*. We used the `bionj` function from the *ape* package in R [Paradis et al., 2004] to build the tree. 126
5.18 Multiple sequence alignment of a few Class 2 alleles from both *DBLMSP* and *DBLMSP2* (belonging to green clade in Fig. 5.17). Alleles with the "m4" prefix come from *DBLMSP* and alleles with the "m8" prefix come from *DBLMSP2*. The alignment was obtained with `ClustalW`. . 127
5.19 Within-country frequencies of dimorphic classes in *DBLMSP* (right) and *DBLMSP2* (left). Overall frequencies: 68% class 1 and 32% class 2 in *DBLMSP*, and 74% class 3 and 26% class 2 in *DBLMSP2*. 129
5.20 Population level frequencies of haplotypes in the dimorphic region of the DBL domain of *DBLMSP*. Alleles at the bottom are sorted in the order imposed by their phylogenetic tree. 130
5.21 Population level frequencies of haplotypes in the dimorphic region of the DBL domain of *DBLMSP2*. Alleles at the bottom are sorted in the order imposed by their phylogenetic tree. 131
5.22 Meta-analysis of linkage disequilibrium between the dimorphic status of *DBLMSP* and *DBLMSP2* per country. $OR = 2.03$, with a 95% confidence interval $CI = (1.07, 3.84)$. The random effects model `meta.DSL` in the R package *rmeta* was used for this analysis. 133

5.23 Joint frequency heatmap of alleles present in *DBLMSP* and *DBLMSP2*. Note that rows and columns are sorted by dendrograms built from the genetic distance between alleles and not from the distance calculated from their frequency patterns. 135
5.24 LD patterns in Ghana. Each row represents a sample, with a red dot in the column corresponding to its *DBLMSP* allele and a blue dot in the column corresponding to its *DBLMSP2* allele. Samples are sorted according to genotype similarity, whereas alleles are sorted based on their distance in the phylogenetic tree. 136
5.25 LD patterns in Cambodia. Each row represents a sample, with a red dot in the column corresponding to its *DBLMSP* allele and a blue dot in the column corresponding to its *DBLMSP2* allele. Samples are sorted according to genotype similarity, whereas alleles are sorted based on their distance in the phylogenetic tree. 137

A.1 Genomic coordinates of *DBLMSP* regions without read support when mapping to the personalised references in samples where the gene's sequence cannot be fully resolved. The inference was performed on the reference graph updated with 2 novel DBL domain haplotypes discovered from `Cortex` contigs, described in Sections 5.4.2.2 and B.4. The consistent gap in Fig. 5.10, produced by inference from the reference graph without these 2 contigs, is eliminated. 151
A.2 Genomic coordinates of *DBLMSP2* regions without read support when mapping to the personalised references in samples where the gene's sequence cannot be fully resolved. The inference was performed on the joint *DBLMSP* and *DBLMSP2* reference graph. 152
A.3 IGV screenshot of read pileup in the *DBLMSP* region, with respect to the 3D7 reference for validation sample 7G8. The pileup has a gap of 476 nucleotides in the DBL domain and additional clustered SNPs. . 153
A.4 IGV screenshot of read pileup in the *DBLMSP* region, with respect to the personalised reference for validation sample 7G8. The pileup shows even coverage and 9/9 SNPs removed compared to the figure above. . 153
A.5 IGV screenshot of read pileup in the *DBLMSP* region, with respect to the 3D7 reference for validation sample GB4. The pileup presents a dip in coverage between coordinates 1415015-1415030. 154

A.6 IGV screenshot of read pileup in the *DBLMSP* region, with respect to the personalised reference for validation sample GB4. The dip in coverage in the figure above gets smoothed out and 7/8 SNPs are removed. 154
A.7 IGV screenshot of read pileup in the *DBLMSP* region, with respect to the 3D7 reference for validation sample KH02. The pileup does not show any gaps, but shows clustered SNPs which could produce alignment problems. 155
A.8 IGV screenshot of read pileup in the *DBLMSP* region, with respect to the personalised reference for validation sample KH02. The pileup shows 18/20 SNPs removed from the figure above. 155
A.9 IGV screenshot of read pileup in the *DBLMSP* region, with respect to the 3D7 reference for validation sample KE01. 156
A.10 IGV screenshot of read pileup in the *DBLMSP* region, with respect to the personalised reference for validation sample KE01. The pileup shows 6/6 SNPs removed from the figure above. 156
A.11 LD patterns in Gambia. Each row represents a sample, with a red dot in the column corresponding to its DBLMSP allele and a blue dot in the column corresponding to its DBLMSP2 allele. Samples are sorted according to genotype similarity, whereas alleles are sorted based on their distance in the phylogenetic tree. 157
A.12 LD patterns in Senegal. Each row represents a sample, with a red dot in the column corresponding to its DBLMSP allele and a blue dot in the column corresponding to its DBLMSP2 allele. Samples are sorted according to genotype similarity, whereas alleles are sorted based on their distance in the phylogenetic tree. 158
A.13 LD patterns in Mali. Each row represents a sample, with a red dot in the column corresponding to its DBLMSP allele and a blue dot in the column corresponding to its DBLMSP2 allele. Samples are sorted according to genotype similarity, whereas alleles are sorted based on their distance in the phylogenetic tree. 159
A.14 LD patterns in Guinea. Each row represents a sample, with a red dot in the column corresponding to its DBLMSP allele and a blue dot in the column corresponding to its DBLMSP2 allele. Samples are sorted according to genotype similarity, whereas alleles are sorted based on their distance in the phylogenetic tree. 160

A.15 LD patterns in Congo. Each row represents a sample, with a red dot in the column corresponding to its DBLMSP allele and a blue dot in the column corresponding to its DBLMSP2 allele. Samples are sorted according to genotype similarity, whereas alleles are sorted based on their distance in the phylogenetic tree. 161
A.16 LD patterns in Vietnam. Each row represents a sample, with a red dot in the column corresponding to its DBLMSP allele and a blue dot in the column corresponding to its DBLMSP2 allele. Samples are sorted according to genotype similarity, whereas alleles are sorted based on their distance in the phylogenetic tree. 162
A.17 LD patterns in Laos. Each row represents a sample, with a red dot in the column corresponding to its DBLMSP allele and a blue dot in the column corresponding to its DBLMSP2 allele. Samples are sorted according to genotype similarity, whereas alleles are sorted based on their distance in the phylogenetic tree. 163
A.18 LD patterns in Thailand. Each row represents a sample, with a red dot in the column corresponding to its DBLMSP allele and a blue dot in the column corresponding to its DBLMSP2 allele. Samples are sorted according to genotype similarity, whereas alleles are sorted based on their distance in the phylogenetic tree. 164
A.19 LD patterns in Myanmar. Each row represents a sample, with a red dot in the column corresponding to its DBLMSP allele and a blue dot in the column corresponding to its DBLMSP2 allele. Samples are sorted according to genotype similarity, whereas alleles are sorted based on their distance in the phylogenetic tree. 165
A.20 LD patterns in Bangladesh. Each row represents a sample, with a red dot in the column corresponding to its DBLMSP allele and a blue dot in the column corresponding to its DBLMSP2 allele. Samples are sorted according to genotype similarity, whereas alleles are sorted based on their distance in the phylogenetic tree. 166

B.1 Toy example of shared k-mer collapsing. On the left, a multiple sequence alignment of 5 sequences sharing a k-mer (black line). After collapsing, there are 2 MSAs are left, bridged by the k-mer . The pseudocode for this step is described in Algorithm 5. Note that some details have been omitted from the algorithm for clarity. 171

B.2 Mapping of first novel contig to 3D7 **(a)**, and to the graph-inferred reference **(b)**. The contig section covering the pileup gaps is highlighted in blue. Top diagram and extract from SAM file show how most of the contig maps in DBLMSP, but its tail (including blue section) has a better match in the DBL domain of DBLMSP2. Bottom diagram and SAM extract show the contig maps entirely in *DBLMSP* when the graph-inferred reference is used. 176
B.3 Mapping of second novel contig to 3D7 **(a)**, and to the graph-inferred reference **(b)**. The contig section covering the pileup gaps is highlighted in blue. Top diagram and extract from SAM file show how most of the contig maps in DBLMSP, but its tail (including blue section) has a better match in the DBL domain of DBLMSP2. Bottom diagram and SAM extract show the contig maps entirely in *DBLMSP* when the graph-inferred reference is used. 177
B.4 IGV screenshot of pileups of Illumina reads mapped to the PacBio assembly for sample GN01, in the *DBLMSP* region (top) and the *DBLMSP2*region (bottom). White reads map equally well in both genes because of a 278 bp sequence shared by their DBL domains. . . 178

Chapter 1

Introduction

Reference genomes provide the basis of analysing genetic variation and functional annotations nowadays. First, they serve as a guide for building new genomes when individuals from the same species are re-sequenced. The output of sequencing techniques is a collection of DNA fragments (called *reads*) in arbitrary order, whose location in the original genome is unknown. This typically results in a costly effort to construct the first genome of a species, as its sequence needs to be patched up from scratch from the reads, a process called *de novo assembly*. Once this reference genome is obtained, the process of reconstructing other sequenced genomes is heavily simplified by lining up reads against it, a process called *reference-assisted assembly* through *read mapping*. The reference sequence essentially acts as a map, analogous to the picture of a puzzle box when putting together the pieces. By comparing new genomes to the reference and measuring the differences, small genetic variants can be identified quickly and cheaply. This is an essential step that is regularly performed, preceding studies that attempt to link genetic characteristics to phenotypic traits. Hence, reference genomes represent a backbone that the majority of bioinformatic pipelines rely on. They serve not only as a guide to assemble new genomes, but also as a coordinate system in terms of which variants are defined and annotations are

reported.

Sequencing the first human genome (i.e. the human reference [Lander et al., 2001]) was a historic scientific milestone that was achieved by an international group of hundreds of scientists, costed 3 billion dollars and lasted 15 years [Collins, 2003]. Since then, the costs have dropped rapidly with the emergence of Next-Generation Sequencing (NGS) technologies, to $1000 per genome (https://www.illumina.com/systems/sequencing-platforms/hiseq-x.html). As a result, thousands of genomes have been sequenced already. That figure is set to double each year, as genome sequencing is becoming routinely used in the clinics as a diagnostic test for pediatric and cancer diseases. In an initiative towards personalised medicine, hundreds of thousands of genomes are expected to be reached in the next few years (https://www.genomicsengland.co.uk/the-100000-genomes-project/). Large amounts of complete genomes have also been generated for non-human organisms, enabling detailed comparative genomics analyses [Tettelin et al., 2005], [Tettelin et al., 2008], [Nguyen et al., 2014], [Dutilh et al., 2014], [Lapierre and Gogarten, 2009]. Sequencing projects have greatly enhanced the knowledge of human and non-human genetic variation [The 1000 Genomes Project Consortium, 2015], [Weigel and Mott, 2009], [Aflitos et al., 2014], [Miles et al., 2016] and much of this knowledge is publicly available, but it is not used when new genomes are assembled based on the reference sequence. This results in gaps of understanding certain forms of genetic variation. While it has provided a simple and convenient framework to organise genomic data, with the fast pace of data aquisition, it makes sense to upgrade the traditional reference genome with the knowledge gained from this data.

The inherent limitation of the reference genome in its current form is that it represents a single account of the DNA sequence of a species, and yet variants from a large number of individuals are defined with respect to it. This poses two types of issues. One refers to the accuracy of variant detection when the sample under analysis

is compared to only one genome. If there are regions that are very different, reads won't match the reference sequence, similarly to pieces from a different puzzle not fitting in. This means that large or complex variants and their potential effects on phenotypes of interest are often missed. This issue needs to be addressed and will be the main focus of this thesis, as many organisms contain biologically relevant regions that are highly diverse (enumerated in Section 2.2). The other issue refers to the efficiency of widely-used bioinformatic pipelines, which routinely spend computational resources detecting common variants that have been found before (96%-99% of the variants observed in a typical human genome are common [The 1000 Genomes Project Consortium, 2015]). This means that the more genomes are sequenced, the more the variants they contain were previously discovered. This issue is more prevalent in humans than in other species, because their genomes are bigger, but extremely similar to one another [The 1000 Genomes Project Consortium, 2015]. The first issue is more prevalent in pathogens, which are generally much more diverse, as they have higher mutation and recombination rates. However, humans also have hotspots of high sequence variation that are of interest (detailed in Section 2.2), despite their low overall diversity.

To illustrate the shortcomings of a single reference genome in characterising variants in high-diversity regions, we show an example from the malaria parasite *Plasmodium falciparum*. Figure 1.1 shows a read pileup from a parasite isolate in the *MSP1* gene, the leading blood-stage malaria vaccine candidate, which is known to be unusually diverse [Miller et al., 1993]. A pileup is a visual representation of the way a sample's sequencing reads match a reference and is typically used to inspect genetic variants. In Fig. 1.1, the top horizontal axis represents the *Plasmodium falciparum* reference sequence and its coordinates, and grey horizontal segments represent reads aligned to it. Each row of reads corresponds to a copy of the original genome that was sequenced. Coloured lines within a read represent mismatches between the

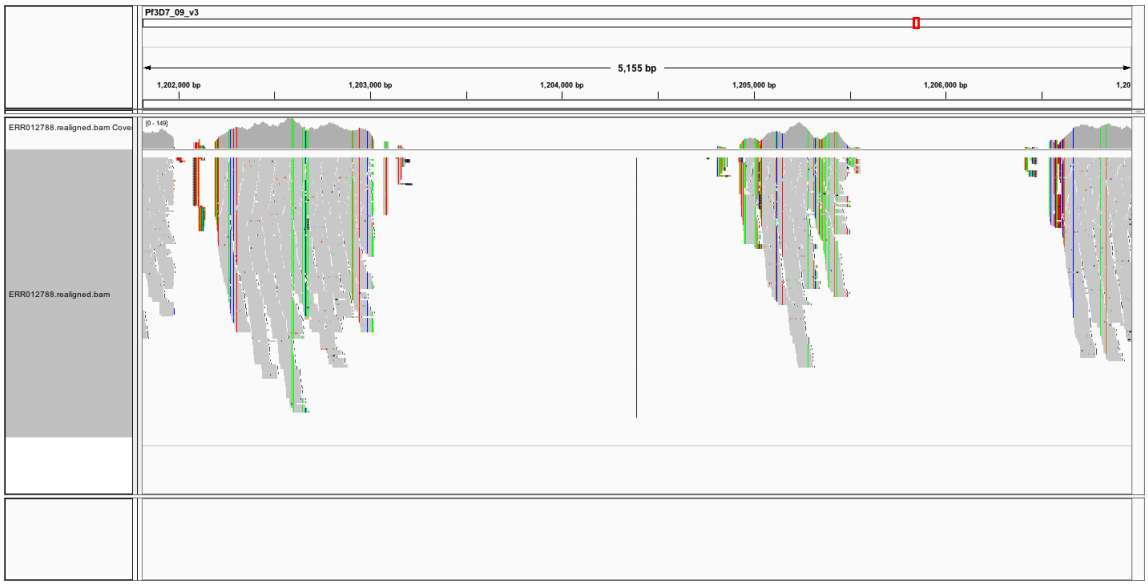


Figure 1.1: IGV screenshot of a read pileup on top of the standard *Plasmodium falciparum* reference genome in *MSP1* gene. Large segments of the gene have no read coverage, rendering the corresponding sequence inaccessible.

respective read and the reference, caused either by sequencing errors or a genetic variant contained in the sample. True variants are distinguished when the majority of reads covering their position in the reference show a mismatch (a vertical coloured column). The band in between the reads and the reference represents an aggregate of the reads – the distribution of depth of coverage is plotted in grey and coloured vertical lines mark that the read consensus indicates there is a variant in the sample at that position. There is a gap in the pileup between positions 1,203,000-1,205,000, meaning that no reads can be aligned to that section of the reference, likely due to the sample having very divergent sequence in that region. Consequently, this sample's variant sequence in that region remains unknown.

In order to circumvent these limitations, there has been increasing interest in generalising the concept of a single linear reference to a richer data structure that captures information about variation and similarity between genomes that have already been sequenced. In other words, the idea is to use all known genomes as reference when interpreting new sequencing data. In theory, this seems like an obvious solution: the

more genomes in the reference, the more likely it is for reads that don't match the usual reference sequence to match one of the other genomes. However, implementing it implies a paradigm shift that adds layers of complexity to all downstream analyses. In particular, the reference concept is moving from a linear sequence towards a graph-based representation where different paths correspond to alternative sequences previously encountered in variable genomic regions. Not only does this increase the search space for reads from new samples, but it also requires a complete rethink of all steps in the pipeline leading to genome inference. At such an increased scale of the problem, memory usage and computing time are a major challenge. As such, there is not yet a practical toolkit, or even a theoretical framework for genome analysis based on multiple references, that has been widely adopted by the bioinformatic community. Moreover, the effectiveness of an extended reference in the final goal of variant detection and genome inference has not been investigated extensively in real data yet.

The aim of this thesis is to build a compressed encoding of many known genomes from the same species, which is called a population reference graph (PRG) and acts as a generative model for sequences. Given sequencing reads from a new sample, this data structure should facilitate the inference of the underlying genome. We will present a method designed to enable this approach. First, reads are matched to the PRG and the sequence path with most coverage is extracted. This inferred sequence is essentially a mosaic of the genomes included in the PRG, that best represents the sample's reads. The rationale behind this relies on one key mechanism that drives genomes evolution. Due to recombination[1], homologous chromosomes exchange chunks of DNA, inducing in offspring a combination of genetic traits from their parents. Therefore, once many genomes have been sequenced (and added to the PRG), any new genome is expected to look like a mosaic of the previous ones. If this mosaic can be inferred based on the reads, it is likely to be similar enough

[1] more details in Section 2.1

to the sample to enable variant detection in the traditional way. Hence, we use the sequence inferred from the reads matched to the PRG as a "personalised reference" with standard tools for variant detection based on a single reference. The advantage of this approach is that it exploits the additional variant information given by the genomes that have already been sequenced, while still utilising established bioinformatic pipelines. We will show that this approach leads to improved genome inference in *Plasmodium falciparum*.

Chapter 2

Background

In this chapter, we give some background information on studying genetic variation using short-read DNA sequencing data. First, we briefly go through the evolutionary processes that may have given rise to patterns of genetic variation we observe in the present-day. Then, we describe different types of complex sequence variation that are not yet clearly understood due to technical shortcomings of current tools interpreting sequencing data. We also argue the biological importance of characterising this class of variants. In Section 2.3, we introduce high-throughput sequencing (HTS) and the problem of reconstructing a genome from its output, a set of DNA reads. We present the two widespread solutions to this problem: reference-assisted and *de novo* assembly. Then we outline the immediately downstream analysis called variant calling. In Section 2.5, we introduce recent proposals to rethink these pipelines in the context of a graph-based reference structure that integrates prior information about variation. We detail the challenges posed by this approach and review the advances made in the direction of overcoming them. In Section 2.6, we list distinctive features of the *Plasmodium falciparum* genome, that make it particularly hard to characterise with standard HTS analysis pipelines and make it a suitable application for the advanced type of approach described in Section 2.5. We hypothesise that using a graph-based

reference when analysing HTS data from *Plasmodium falciparum* would recover some of the variants missed by the standard methods. At the end of the chapter, we give an overview of the remainder of this thesis.

2.1 Brief overview of diversification mechanisms

In this section, we give a very brief summary of biological mechanisms that generate genetic diversity, which need to be taken into account when analysing and interpreting variants at population level. The content of this section will come up in a range of contexts throughout the thesis. Some of the material on selection will be relevant in the next section, when highlighting the reasons why certain forms of genetic diversity are of interest and worth describing in detail. The different modes of balancing selection will be relevant in Chapter 5, when interpreting population scale results on certain genetic variants in *Plasmodium falciparum*. The material on recombination serves for understanding how certain patterns between variants are formed and why it is important for this process to be mimicked by a reference graph representation that aims to be a generative model for new sequences. It will also be relevant for interpreting the patterns of genetic variation observed in our study in Chapter 5. This is a very high-level introduction to population genetics. More more detail, refer to [Casillas and Barbadilla, 2017].

Mutation is a permanent change in the nucleotide sequence of a DNA segment, usually due to replication errors. *Natural selection* is an evolutionary process that is manifested through differential reproduction of organisms with genetic changes that alter phenotypes in a way that affects survival and procreation success. As a result, the frequency of genetic variants that are under selection changes over time. Natural selection can take multiple forms. *Positive selection* favours the increase in frequency of a variant that confers an advantageous trait that increases the likelihood of survival

for the individual possessing it. *Negative selection*, or *purifying selection*, decreases the frequency of a variant that has a deleterious effect on the fitness of the individual possessing it, decreasing its likelihood of survival. *Balancing selection* maintains the frequency of variants at intermediate values. This can occur via 3 factors. One is *heterozygote advantage*, present in diploid organisms when individuals carrying both versions of a variant (heterozygotes) are fitter that individuals carrying two copies of a single version (homozygotes). Another is *diversifying selection*, occurring when multiple divergent versions of a variant are maintained in the population because they all produce fit phenotypes in certain environmental conditions. The third factor is *frequency-dependent selection*, when the fitness of a variant is relative to the frequency composition of other versions of the variant in the population. Genomic regions under balancing selection are frequently difficult to characterise from short-read sequencing data with reference-based techniques and are a principal application target for multiple-reference-based techniques. One such region will be the subject of study in Chapter 5.

Recombination is another major force that shapes the patterns of genetic variation we currently observe in the population. It involves two chromosomes exchanging pieces of DNA. In sexual division it is also known as *crossing over*, as homologous chromosomes align and swap DNA sections, resulting in offspring having a novel combination of genes from both parents. Recombination can also occur during asexual division or any time homologous chromosomes align. It is one of the mechanisms employed in DNA repair, as the homologous chromosome can serve as a template for correcting a double-strand breaks. Recombination normally requires similarity between the two DNA molecules that line up. Occasionally, misalignments can occur between pieces of DNA that have high sequence similarity, but are not alleles of the same genomic region; instead, they are usually paralogous sequences that previously evolved through segmental duplication. Recombination resulting from these

misalignments is called *non-allelic homologous recombination*. These misalignments can sometimes generate *unequal crossing over*, which results in the deletion of one of the paralogous sequence in one chromosome and the addition of an extra paralogous sequence in the other. Another particular form of recombination is called *gene conversion*, which involves a one-way crossing over that causes a DNA section to replace a homologous one in a "copy-paste" manner. *Non-allelic gene conversion* can also occur when DNA is copied from a paralogous locus.

2.2 Complex variation and its biological significance

Genetic diversity is an essential foundation for natural selection to act upon, and it provides populations with flexibility to adapt to changing environmental factors. Across long evolutionary timescales, a number of biological mechanisms that generate genetic diversity have been developed. Based on their size, genetic variants fall in two categories: small scale (<50bp) and large scale (>50bp) [Feuk et al., 2006]. Small scale genetic variants are most frequent and usually involve single nucleotide polymorphisms (SNPs), which change one letter in the DNA sequence, small indels, which insert/delete a few letters, and short tandem repeats, which repeat a sequence a variable number of times. Large scale genetic variants, or structural variants, are rearrangements of large stretches of DNA sequence that change the structure of a chromosome. These can be: large insertions or deletions; copy-number variants (CNV), which are large DNA segments repeated a variable number of times, usually resulting from unequal crossing over events; translocations, which are DNA segments that occur at a different position in the chromosome; inversions, which are DNA segments with reversed orientation compared to the rest of the chromosome [Feuk et al., 2006]. SNPs and small indels can usually be revealed by HTS analysis pipelines based on a single

reference genome, but can be problematic if they are densely clustered. Structural variants affect genomic regions that are larger than sequencing reads (which are typically 75-150 bp), so they are more difficult to characterise. They require completely different algorithmic approaches because the sequence induced is very divergent from the reference [Sudmant et al., 2015]. We will now give examples of complex variants found across a range of organisms and explain their biological relevance.

In humans, the most distinctive example is the Major Histocompatibility Complex (MHC) region, 4 Mb in size, which has a key role in response to infection, autoimmune diseases and transplant medicine [Lechler and Warrens, 2000]. The MHC region is characterised by extremely high levels of polymorphism [Horton et al., 2008], with certain stable haplotypes despite the diversity. It contains the hypervariable *HLA* genes, some of which can occur in 3000-4000 allelic forms (`http://www.ebi.ac.uk/imgt/hla/stats.html`, accessed date). They have a major role in modulating the adaptive immune system, presenting fragments of pathogens from inside or outside human cells to the immune cells responsible for their destruction. Because of this function, it is generally assumed that the high diversity of these genes is a product of balancing selection, enabling the *HLA* molecules to bind to a wide range of pathogens. This has been backed up by observing patterns of allelic variation, such as predominant alleles at intermediate frequencies and ancient alleles that have been maintained since before the human/chimpanzee split [Hughes, 2002]. Genome-wide scans for positive selection have also found hits in the *HLA* region [Pickrell et al., 2009]. These findings suggest that getting a clear picture of the sequence variants in this region is crucial in understanding immune-linked diseases. With next-generation sequencing data, this remains a challenge because of extreme structural variation, multiple paralogous loci with high degree of sequence similarity and limited recombination/gene conversion between some of these loci.

The glycophorin gene family is another instance of structural polymorphism, with

signatures of ancient balancing balancing selection predating the human/chimpanzee split [Leffler et al., 2013]. Its complex variation patterns define the MNS blood group system [Blumenfeld and Huang, 1995] and a particular CNV is thought to cause resistance to severe malaria [Leffler et al., 2017]. The region consists of 5 paralogous genes with high sequence similarity, undergoing frequent gene conversion that increases nucleotide diversity [Ko et al., 2011].

Other structurally diverse regions in the human genome are: the *KIR* gene cluster, which regulates the killing of infected human cells by interacting with the MHC molecules [Jiang et al., 2012]; the olfactory gene cluster which comprises a diverse repertoire of paralogous loci produced by multi-chromosomal duplications [Trask et al., 1998], with function to discriminate between a large number of odors; an ancient inversion of length 900 Kb that is common and selected for in europeans [Stefansson et al., 2005].

Large inversions are also highly abundant in the *Anopheles* mosquitoes, vectors of malaria [Sharakhov et al., 2006]. These inversions have far-reaching epidemiological consequences, as they provide the necessary diversity to adapt to different ecological conditions. This allows the vector to penetrate new environments, leading to an increase in the geospatial distribution of disease-carrying mosquitoes and consequently, in malaria transmission [Coluzzi, 2002].

Perhaps the most pronounced case of medically relevant structural variation is found in bacteria, which have the capacity to acquire genetic material from other bacterial cells through plasmids. Plasmids are dispensable genetic elements separated from the main chromosome. They serve as platforms where new genes are acquired and rearranged through a variety of recombination systems, enabling adaptations to new conditions, including the presence of antibiotics. Plasmids act as vehicles for spreading antibiotic resistance genes, sometimes to a broad range of bacterial hosts that include very different species. Bacteria can also develop antibiotic resistance

genes through other mobile genetic elements called transposons, which enable the movement of DNA from one location to another within the same cell. They can jump within the main chromosome, or from the main chromosome to plasmids and the other way around, or from plasmid to plasmid, without even requiring sequence homology like normal recombination processes. The information in this paragraph was compiled from [Bennett, 2009].

Beta-lactamases are bacterial mobile elements that produce enzymes capable of breaking down the structure of common beta-lactam antibiotics, with the potential to generate extended-spectrum resistance. They are often plasmid mediated and characterised by extensive sequence diversity. The classically known beta-lactamase families, which are also most common and multi-drug resistant, are *TEM* (over 130 allelic variants known), *SHV* (over 50 variants) and *CTX-M* (over 40 variants) [Jacoby and Munoz-Price, 2005]. A more recently evolved multi-drug resistant family is *OXA*, with 150 heterogeneous gene variants already reported, some with very low sequence identity [Poirel et al., 2010]. They are found on a variety of genetic platforms that facilitate their transfer and unsurprisingly, they have spread worldwide [Poirel et al., 2010]. This is of major concern as some variants can break down carbapenems antibiotics, the last line of treatment against multi-drug resistant infections [Evans and Amyes, 2014].

As genome sequencing becomes less costly, it promises to provide additional insight into the evolution of drug resistance on different genetic backgrounds, enabling epidemiological studies for tracking the spread of resistance [Evans and Amyes, 2014]. Furthermore, genome sequencing is anticipated to be the most efficient alternative for typing bacterial isolates [Didelot et al., 2012] and could contribute to global bacterial surveillance. As such, computational problems in characterising genetic regions of high-diversity present a roadblock from translating whole-genome sequencing into clinical practice. One solution using population-based genome graphs has been shown

to be viable for typing antibiotic resistance in *Staphylococcus aureus* and *Mycobacterium tuberculosis* from sequencing data [Bradley et al., 2015].

Extreme levels of polymorphism are encountered in genomic regions that undergo recurrent rearrangement, such as the *var* gene family in *Plasmodium falciparum*, which drives immune evasion by repeatedly switching expression of one of its members. It has been shown that these genes undergo frequent non-allelic homologous recombination (NAHR) during mitotic division inside human blood cells. The crossovers are error-free despite relatively low levels of sequence similarity between different *var* members, and the recombinant gene preserves the overall domain structure. This mechanism could generate millions of rearranged antigenic sequences each day through the course of an infection, in order to avoid immune recognition. The information in this paragraph was compiled from [Claessens et al., 2014]. NAHR and recurrent rearrangements at duplication sites have also been reported in humans [Sudmant et al., 2015]. However, they tend to occur in a less structured manner, generating CNVs responsible for certain genetic syndromes [Liu et al., 2012], [Lupski and Stankiewicz, 2005].

Even in structurally stable genomic regions, mapping based methods have difficulties disentangling complex variants containing indels. This is due to different ways of representing them, via multiple equivalent alignments with respect to the reference. As a result, spurious SNPs can be called if the precise position and size of the indel are not determined by the alignment. An example of this is given in Fig. 2.1 (from[1] [Zook et al., 2014]), where the variant CAGTGA—>TCTCT is represented in 4 correct, but different ways by 4 major read mappers. Consequently, the same variant appears in different formats in the output VCF files, making it hard to compare and integrate call sets. In some cases, manual inspection of the read pileups is the only resort.

[1]copyright permission granted

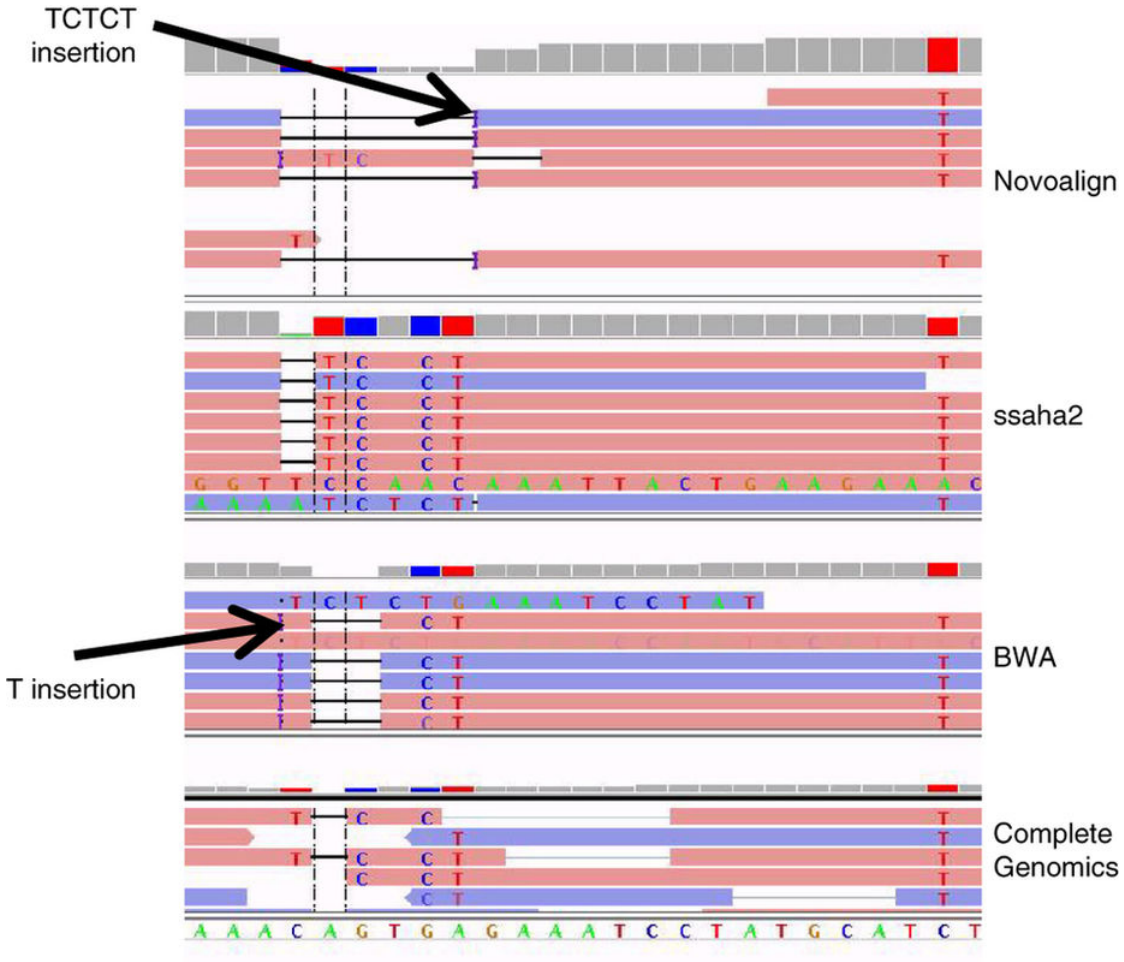


Figure 2.1: Example of complex variant with four different representations from four different mappers, which can cause data sets to appear to call different variants when in reality they are the same variant. In this case, the six bases CAGTGA are replaced by the five bases TCTCT at location 114841792-114841797 on chromosome 1 of the human genome. The four sets of reads are from sample NA12878, sequenced with different technologies, as follows: Illumina mapped with `BWA`, 454 mapped with `ssaha2`, Complete Genomics mapped with `CGTools`, and Illumina mapped with `Novoalign`. Source: [Zook et al., 2014]

2.3 High-throughput DNA sequencing

DNA sequencing is the process that determines the ordered sequence of nucleotides in a strand of DNA. High-throughput sequencing (HTS), or next-generation sequencing (NGS) technologies have made whole-genome sequencing widely accessible to scientists in the past 15 years due to their time and cost effectiveness, revolutionising biomedical research.

Many HTS platforms are available (Roche/454, Illumina, SoliD, Ion Torrent), with Illumina being the most common on the market. They all tend to broadly follow a similar workflow: library preparation, clonal amplification, followed by thousands of parallel sequencing reactions. Library preparation generally consists of setting up the DNA into a form that is compatible with the sequencing platform and involves fragmenting the DNA to a short length (varying with the platform but typically less than 500 bp). DNA fragments are then cloned many times and individual clones are sequenced in parallel, generating millions of short sequence reads concurrently. While this sequencing process is fast and accurate (less than 1% error rate), it imposes the computational burden of reconstructing long, contiguous sequences from millions of sequenced fragments. This is a difficult problem, particularly with repetitive sequences larger than the read length. The material in this section was compiled from [Reuter et al., 2015].

"Third-generation" sequencing technologies that produce longer reads ($> 10,000$ bp) have been developed, such as Pacific Biosciences and Oxford Nanopore, but cannot be yet used routinely either because of prohibitive cost (former) or lower accuracy (latter). This thesis is mainly concerned with the analysis of raw output from short-read HTS data (Illumina specifically), but will make use of long-read sequencing data in Chapter 5.

2.4 Classic HTS data analysis

The main analysis challenge with HTS data is reconstructing the original genome that was sequenced from millions of short, arbitrary sequence reads, i.e. genome assembly. It is essential that solutions to this problem are computationally efficient in order to match the fast pace of data being generated. In this section, we present the two classic approaches: *de novo* assembly, or reference-free assembly, which does not rely on a previously known sequence, and reference-assisted assembly, which is based on mapping reads to a reference genome. The read mapping problem consists of identifying the exact location of a read in the reference genome and a base-per-base alignment between the read and the reference sequence at that respective location.

High-quality *de novo* assembly is typically used to obtain the reference genome from the first sequenced sample(s) of a species. Subsequently, when additional individuals are sequenced as part of re-sequencing projects, their genomes are obtained by mapping reads to the known reference and flagging different bases. The variant calling problem consists of identifying the differences between a sequenced genome and a reference from HTS experiments.

2.4.1 *De novo* assembly

The dominant data structure used in *de novo* assembly of short HTS reads is the de Bruijn graph [Pevzner et al., 2001]. The graph nodes are obtained from the reads by sliding a window of size k along each read and recording subsequences of size k (k-mers), overlapping by $k-1$ bases. The graph edges represent the adjacency between k-mers within reads (see Fig. 2.2). The de Bruijn graph representation handles repetitive sequences compactly, as all repeated k-mers are collapsed onto one node. This solution was introduced in [Chaisson and Pevzner, 2008] and [Zerbino and Birney, 2008], with the graph encoded as a hash table of unique k-mers .

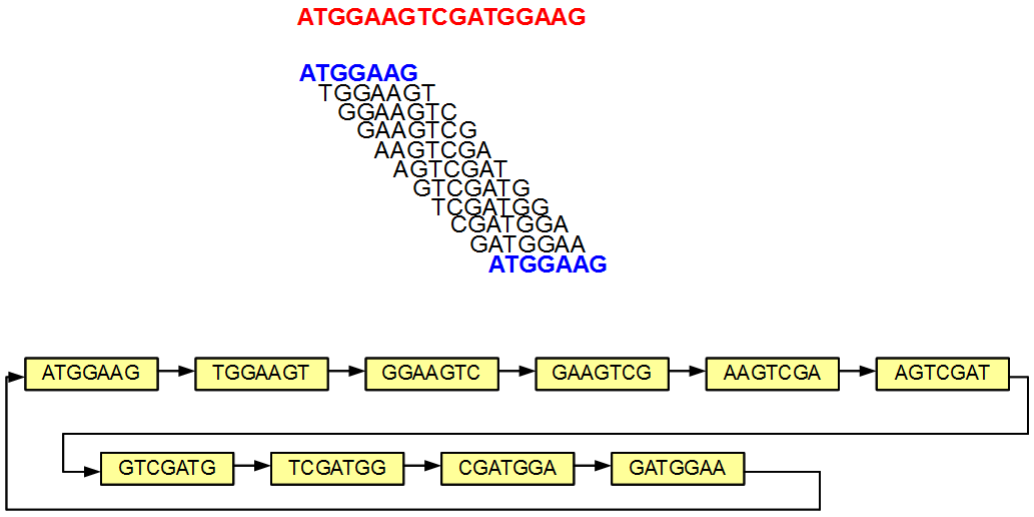


Figure 2.2: De Bruijn graph of order 7 for the sequence **ATGGAAGTCGATGGAAG**. First, all adjacent 7-mers overlapping by 6 bases are extracted from the sequence. Unique 7-mers label the vertices of the graph and edges connect adjacent 7-mers. Source: `http://www.homolog.us/Tutorials/index.php?p=2.1&s=1`

While the graph construction, k-mer search and graph walk were fast, the amount of memory required to store the graph was a main limitation. `Velvet`'s [Zerbino and Birney, 2008] explicit hash table implementation made it impractical to use for large genomes like the human one. `ABySS` [Simpson et al., 2009] was more efficient, using a sparse hash table that could be distributed across a computer cluster system, so that it could run on large data sets. `SOAPdenovo` [Li et al., 2010] reduced memory use by performing error correction of reads before assembly, thus eliminating spurious k-mers and reducing the number of vertices in the graph. By contrast, `ALLPATHS-LG` [Gnerre et al., 2011] only performed error correction of graph paths at the last step in the assembly, attempting to maximise accuracy for a specific type of input reads.

A shortcoming of de Bruijn graph-based assemblers is that they break up reads into k-mers , losing spatial link information contained within the reads. An alternative approach was taken by Simpson [Simpson and Durbin, 2010], who developed a compressed representation of a string graph representing overlaps between reads [Myers, 2005]. It uses the Burrows-Wheeler Transform and FM-index, two data structures

widely used in the read mapping field[2], to assemble reads directly instead of k-mers . The resulting `SGA` assembler [Simpson and Durbin, 2012] became very popular due to its computational performance.

Despite these advances, the cost of computational resources remained high (about 50-100 GB of memory for a human sample with high sequencing depth). Subsequently, compressed representations of de Bruijn graphs have been proposed, using sparse bit vectors [Conway and Bromage, 2011] or probabilistic data structures such as Bloom filters [Pell et al., 2012], [Chikhi and Rizk, 2012], or extensions of the Burrows-Wheeler Transform [Bowe et al., 2012]. The `Minia` implementation [Chikhi and Rizk, 2012], [Chikhi and Rizk, 2013] reduced the memory usage of a human genome assembly to 5 GB, followed by `DBGFM` [Chikhi et al., 2014], which further reduced it to 1.5 GB.

Other tools [Iqbal et al., 2012], [Li, 2012] have focused on the practical aspect of accurate genome inference. They have incorporated variant calling methods based on the assembled output, demonstrating the viability of using *de novo* assembly in resequencing individuals from the same species. They will be discussed further in Section 2.4.3.

2.4.2 Read mapping

The read mapping problem essentially reduces to the approximate/inexact string matching problem from computer science. This consists of searching for the read sequence in the reference while allowing for some small deviations between the two in the form of mismatches and gaps, in order to account for sequencing errors and true genetic variants. The aim is to find the most likely position in the reference where the read originates from, while ensuring reasonable computational performance. Indeed, while there are cases of ambiguity given by repetitive genomic regions, many mappers compromise on exhaustive searches in order to achieve high speed.

[2]which will be detailed later on, in Section 2.4.2 and Section 3.4.4

There are two main approaches underlying mappers in the task of approximate string matching: filtering and indexing. Filtering methods rely on quickly discarding large sections of the reference that do not contain any rough matches for the read. Typically exact matches of small k-mers in the reads (called *seeds*) are searched for and only regions containing clusters of hits are further processed. Indexing methods have been borrowed from the generic text search field and involve a preprocessing step that converts the reference and/or reads into a form that enables fast exact searches for subsequences. The fastest options are suffix-based indices (suffix tree, suffix array, FM-index), which organize the suffixes of a reference in a way that allows regions with multiple copies to be searched only once. The FM-index is very popular because it is space-economical and has the advantage of supporting searches for sequences of any length, whereas searches with k-mer indices are limited by the size of k. Moreover, once the FM-index has been constructed, the search time grows linearly with the read length and is independent of the reference genome length. An alternative index is a hash table of overlapping k-mers in the reference or reads, used in the proprietary software `Novoalign` (`http://www.novocraft.com/products/novoalign/`). The information in this paragraph and the next one was compiled from [Reinert et al., 2015].

The FM-index is the data structure most frequently employed by read mappers and will keep being referred to throughout this chapter, as it is used in a range of other HTS analysis tools, such as *de novo* assembly, phasing and imputation. Further introductory material that describes the theoretical and algorithmic details of the FM-index, with a worked example, will be presented at the beginning of Chapter 3. We chose to place this material there rather than in this section because we think it would make the novel algorithms we developed, contained in that chapter, easier to follow – our method builds directly on the classical FM-index encoding and search.

Whether they employ a filtering or indexing approach, or a combination of both,

many mappers follow one common framework: search for exact occurences of short seeds, narrow down candidate regions that contain seeds covering most of the read, verify the distance between the read and each candidate region and progress to full base-by-base alignment. While mappers vary greatly in the criteria they use for breaking up the reads into seeds and for filtering out seed matches and selecting candidate regions, the third verification step is typically done by dynamic programming, using accelerated versions of the Smith-Waterman algorithm [Smith and Waterman, 1981].

An interesting algorithmic approach was taken by `BWA-MEM` [Li, 2013], which indexes the reference genome with the FMD-index, a variation of the FM-index adapted to the double-strand structure of DNA. The FMD-index combines forward and reverse complement sequence into a single index, enabling fast bidirectional searches. Using this index, it finds super maximal exact matches (SMEMs) between the read and the reference and uses them as seeds for alignments. Maximal exact matches (MEMs) are substrings of the read that match the reference sequence exactly and cannot be extended in either direction of the match (i.e. the left and right characters adjacent to the MEM do not match the reference sequence anymore). Super maximal exact matches (SMEMs) are MEMs that are not contained in other larger MEMs in the read sequence. This advanced seeding algorithm results in high-speed mapping for reads of variable lengths and will be revisited in Section 3.7.1.

Over 60 mapping software are currently available [Fonseca et al., 2012], with options tailored to different HTS technologies that generate input reads with different characteristics. The field has had to adapt to increasing read lengths: the first wave of mappers was handling very short reads ($<$ 100 bp), with limited gap allowance [Li et al., 2008], [Li and Durbin, 2009], [Langmead et al., 2009], whereas the second wave progressed to longer reads, supporting longer gaps and split/partial alignments [Li and Durbin, 2010], [Langmead and Salzberg, 2012], [Li, 2013], [Liu et al., 2014]. Many mappers take advantage of additional information intrinsic to the sequencing

process when ranking multiple candidate alignments. They take into account base quality scores or systematic error modes of the sequencing platform [Fonseca et al., 2012], [Li and Homer, 2010] to determine the most likely location that could have generated the read. Among this plethora of options, there are variable trade-offs between computational efficiency and mapping accuracy, and there is not a single best one that outperforms all others in a range of applications [Hatem et al., 2013].

2.4.3 Variant calling

The variant calling problem was originally tightly linked to the mapping problem, as it involved processing its output, the read alignments, in order to find repeated differences between the reads and the reference. Then, probabilistic models are used to assess whether enough reads disagree with a reference base in order to distinguish a true variant from sequencing errors. The theoretical details of one such model are given in [Li, 2011] and a review of available methods is in [Nielsen et al., 2011]. Popular variant calling software are `GATK`[3], [DePristo et al., 2011], `SAMtools` [Li et al., 2009], `FreeBayes` [Garrison and Marth, 2012]. Again, none of them is unanimously "the best", but each is optimised for certain experimental conditions. Despite some of them being able to exploit additional information from multiple samples under analysis or information about haplotype structure in the population [Garrison and Marth, 2012], [Rimmer et al., 2014], they all struggle in regions that are divergent from the reference because of low quality mappings. For instance, misalignments around indels tend to produce false SNP calls [Li and Homer, 2010], [Zook et al., 2014].

Variant callers based on *de novo* assembly have also been developed and have been shown to improve indel calling [Li, 2012] and complex structural variants calling [Iqbal et al., 2012]. While they differ in the underlying assembly method, both of these tools

[3]will be used in Chapter 5

map *contigs* (contiguous stretches of the genome that were fully assembled from the reads) to the standard reference in order to identify the position of variants. The idea is that some mapping ambiguity is removed in this scenario because contigs are much longer than the reads, conferring robustness to repetitive or divergent sequence. These ideas, together with the `Cortex` variant caller [Iqbal et al., 2012] will be used extensively in Chapter 5.

Apart from the computational burden, there is one significant drawback with the *de novo* assembly-based variant calling: it is not set up to support input reads coming from multiple genomes. This is important in pathogen genomic studies, where samples that are being sequenced could contain multiple genomes due to mixed infections. Most assemblers struggle to distinguish heterozygous variants from errors in the simple case of a diploid genome where there is an even number of reads supporting alternative alleles. Analysing a malaria mixed sample for example is more challenging because it may contain three genomes in the ratio 10:3:1 and each of them could display a different allele of a variant.

2.5 Transition to graph references

The methods described in the previous section, applied to increasing amounts of HTS data, have led to extensive characterisation of genetic variation in a range of species of interest. Many custom algorithms for different types of variants have been developed [Guan and Sung, 2016], but the power to access complex variants like the ones mentioned in Section 2.2 remains limited. A general approach to this problem would be to integrate known variants into the standard reference genome, then map reads and call variants based on this extended structure.

An initial attempt to do this was introducing partial assemblies representing alternative versions of highly variable loci into the human reference genome. Recent

updates of the human reference assembly, have been augmented with multi-kilobase long alternative sequences [Church et al., 2011], [Church et al., 2015]. They are anchored at boundary locations in the primary reference. The `BWA-MEM` read mapper has been modified to be aware of these alternative sequences and report alignments with respect to them when they are closer to the sample, while also being able to project all alignments to the primary reference. However, this solution is not viable for representing detailed variation from large scale population studies because of the huge number of long alternative sequences that would need to be incorporated. Another key issue is that long alleles of the same locus are likely to have high levels of sequence sharing in some regions, producing multimapped reads that are hard to interpret. The homologies and differences between a set of related sequences are naturally represented by a mathematical graph. In its most general form, the nodes represent sequences, with edges connecting nodes with successive sequence [Paten et al., 2017a].

The concept of a graph-based reference that represents the sequence relationships between genomes that have already been constructed has attracted a lot of interest lately, but implementing it in a practical and scalable way is subject to active research. There are a number of challenges that make graph-references hard to incorporate in standard bioinformatic practice, as they involve rethinking many parts of the HTS analysis pipeline. These challenges are listed below, and summarised in Fig. 2.3, taken from[4] [The Computational Pan-Genomics Consortium, 2016], which discusses their consequences in detail.

1. We need a compact data structure that can represent a large number of genomes and the differences between them, accepting input in various formats retrieved by existing software.

2. We need efficient algorithms that can quickly match reads to this increased

[4]article distributed under the CC BY license, which allows reuse with appropriate citation

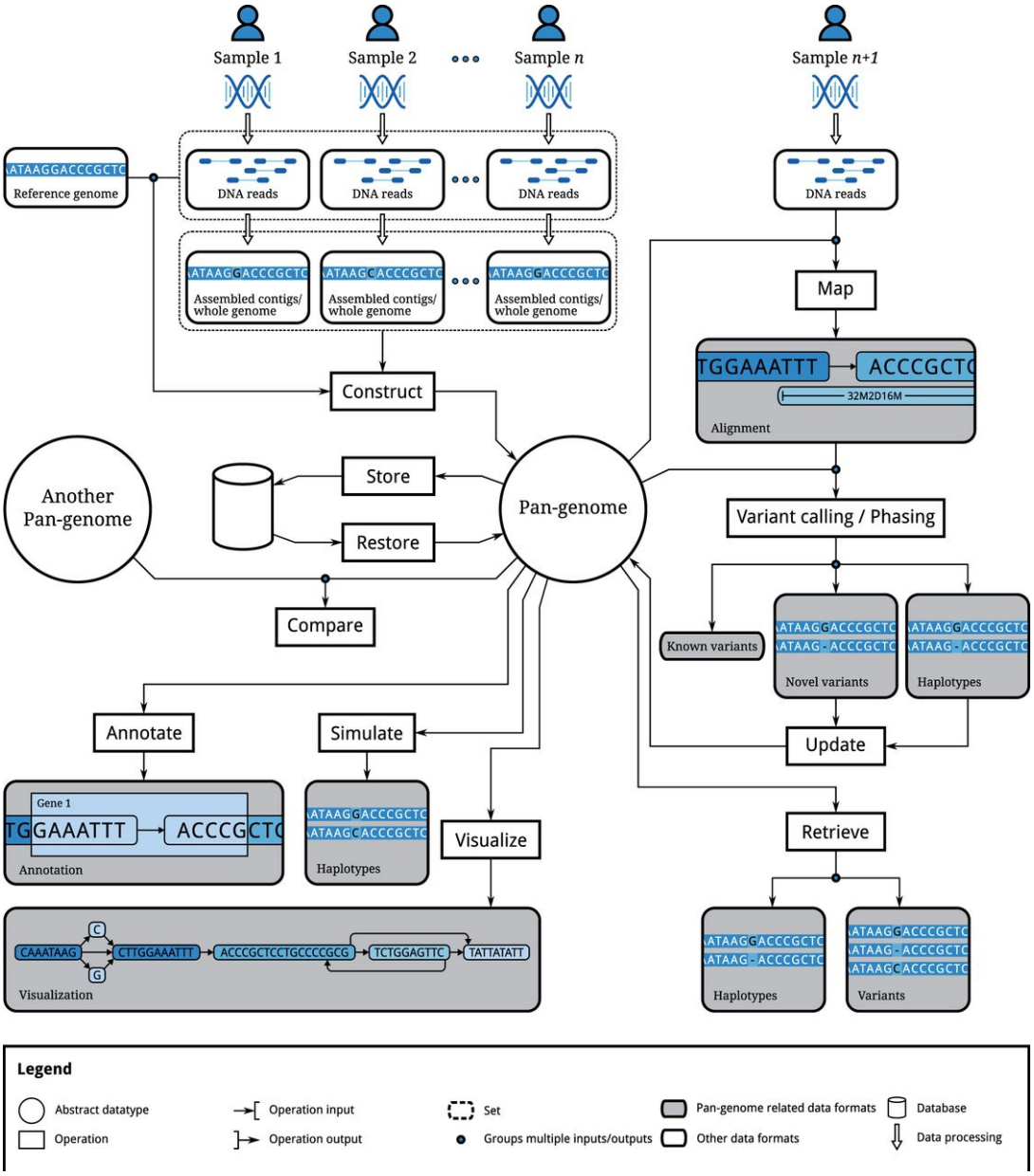


Figure 2.3: A summary of required operations in HTS analysis pipelines after replacing the standard reference genome with a structure of multiple references. Source: [The Computational Pan-Genomics Consortium, 2016]

search space in a way that is aware of alternative sequences.

3. We need variant calling methods that can interpret the output of this new type of mapping to detect novel variants that were not included in the reference structure.

4. We need an unambiguous coordinate system that allows defining variants with respect to alternative sequences in the reference structure.

5. We need to be able to project coordinates from the new reference structure onto the standard reference genome, where functional annotations are defined.

6. File formats representing the new reference structure need to be consistent with community standards, such that its output is compatible with existing tools for downstream analysis.

7. We need to develop benchmarking frameworks that can evaluate the results returned by the new reference structure and compare them with results obtained with the standard reference genome.

Although there has been work done to address each of these issues individually, no solution has been able to tackle all of them simultaneously, while maintaining computational performance and generating validated results on real data sets. In the rest of this section, we list the progress that has been made to date.

The idea of representing a set of related sequences as a graph dates back to 2002, when [Lee et al., 2002] proposed a partial order graph to represent a multiple sequence alignment (MSA). They argued that standard MSA methods are lossy because they reduce the intermediate MSAs to a linear representative with each sequence added to the alignment, potentially missing the optimal alignment. In contrast, their method `POA` (partial order alignment) represents the intermediate MSA as a graph, eliminating the redundancy within the set of sequences by fusing identical, aligned letters in

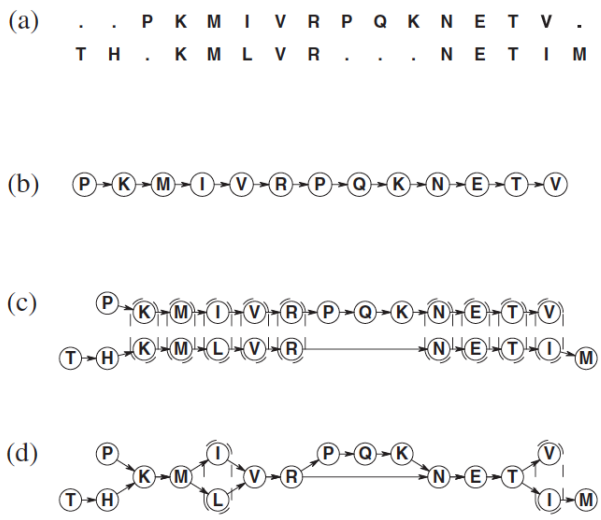


Figure 2.4: Example of the POA representation. (a) Pairwise protein sequence alignment. (b) A single sequence (top in (a)) in PO-MSA format. (c) The sequences in (a), in PO-MSA format, aligned to each other. Dashed circles indicate two nodes are aligned. (d) PO-MSA representation of the pairwise alignment in (a). Dashed circles indicate two nodes are aligned. Source: [Lee et al., 2002]

a single vertex (see Fig. 2.4, from[5] [Lee et al., 2002]). This fusion permits the alignment to switch between sequences at regions of sequence identity, naturally modelling homologous recombination. When an additional sequence is added to the graph MSA (or PO-MSA), its pairwise alignments with all the sequences in the intermediate MSA and their recombinants are considered. This guarantees that the optimal alignment is considered, unlike `Clustal` [Higgins and Sharp, 1988] methods that perform greedy progressive alignments. However, the explicit graph representation of a PO-MSA is not scalable to large genomes.

[Schneeberger et al., 2009] provided the first encoding of multiple references, based on a k-mer index that stores positions of k-mers in each reference genome, with associated algorithms to map reads to it. They showed that the number of mapped reads and variants recalled increases when using a graph of 3 *Arabidopsis thaliana* genomes (of size 135 Mbp) compared to the single reference.

[5]copyright permission granted

[Huang et al., 2013] followed with a practical implementation that scales to many human genomes, `BWBBLE`. They build a multi-reference structure from 1000 Genomes pilot phase variants [Durbin et al., 2010] and show an increase in the number of mapped reads compared with `BWA`, but do not approach the issues past the alignment step. `BWBBLE` encodes SNPs as extra characters from the extended IUPAC alphabet (see Fig. 2.5, adapted from[6] [Huang et al., 2013]). It encodes small indels by padding them with context sequence up to the read length and appending them at the end of the reference genome, separating the alternative sequences with the symbol #. It encodes large structural variants similarly, but only appends the sequences from their boundaries at the end of the reference. The idea is to obtain a multi-reference structure linearised into an extended string, such that a modified version of the FM-index algorithm underlying standard mappers can be applied. In the case of non-SNP variants, standard mapping should work unchanged because the reference structure is designed such that reads containing divergent sequence should map entirely between two # symbols. Due to the inconsistency between treatment of different variant types, the main issue with this approach is that its mapping output is hard to interpret in the context of variant calling for non-SNP variants. By concatenating alternative sequences from different loci at the end of the reference, information about their location in the genome is lost and it is unclear whether it could be recovered in order to do full genome inference. Furthermore, the padding around indels can quickly increase the size of the multi-reference structure, resulting in a high memory overhead. The `BWBBLE` representation is the most conceptually similar to our approach, so the performance of our method will be measured against it in Chapter 4.

A general solution that represents all types of variants naturally was proposed by [Sirén et al., 2014]. They generalised the FM-index for linear sequences into a method for indexing graphs. Their graph index allows queries of paths in arbitrary

[6]article distributed under the CC BY license, which allows reuse with appropriate citation

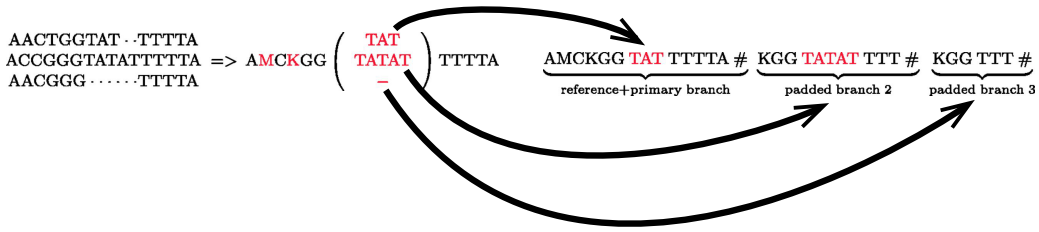


Figure 2.5: Reference multi-genome constructed by superimposing three sample genomes with `BWBBLE`. SNPs are encoded in the reference branch as additional characters, with **A**–>**C** and **T**–>**G** being represented by **M** and **K** respectively. Indels are appended at the end of the reference branch after some padding. In this example, the length of the padded sequence corresponds to reads of length 4. Adapted from: [Huang et al., 2013]

graphs, in an analogous way to FM-index queries for subsequences in a sequence. In this way, the graph can explicitly represent deviations of any type from the reference sequence and reads can be mapped along alternative paths. The bottleneck of this method was the high memory requirement during index construction, due to the combinatorial explosion in the number of paths in complex genomic regions with many nearby variants. For a human graph with common variants, `GCSA` initially required over 1 TB of RAM, reduced to 215GB after pruning the number of haplotypic combinations allowed for nearby variants [Sirén et al., 2014]. More recently, `GCSA2` [Sirén, 2017] has been published, based on a combination between k-mer index and generalised FM-index, further reducing the memory consumption. A whole-genome index of the human reference and 1000 Genomes variants [The 1000 Genomes Project Consortium, 2015] requires 96GB, with an additional cost of 1-2TB of disk I/O. Once the index is built, it can support fixed-length exact matching queries to a graph of arbitrary sequence variants in time comparable to standard FM-index queries on linear sequences. This core functionality is underlying the graph mapper in the `vg` toolkit, which anchors alignments with maximal exact matches (MEMs), followed by Smith-Waterman extension, in a similar way to `BWA-MEM`. `vg` (`https://github.com/`

vgteam/vg) is a community effort to build production quality pipelines, analogous to `HTSlib` and `SAMtools`, based on genome graphs, that will eventually replace the current reference genome. It has not been published yet, but it is under public, active development.

Another tool that combines `GCSA` with a read mapper and also attempts to reduce the requirements of index construction is `HISAT2` (https://ccb.jhu.edu/software/hisat2/index.shtml). It uses a global graph FM-index composed of a collection of small, local FM-indices that covers the entire genome. The method itself is not documented in the literature, but the software implementation has been used to index a graph of the human genome plus 12 million common SNPs (with a memory footprint of 6.7 GB) and analyse RNA-seq reads [Pertea et al., 2016].

A completely different approach that avoids mapping entirely was taken by [Dilthey et al., 2015]. Its reference data structure, called PRG (population reference graph), is similar to that in [Schneeberger et al., 2009], indexing unique k-mers in a graph equivalent to a multiple sequence alignment (Fig. 2.6, from[7] [Dilthey et al., 2015]). However, instead of mapping reads, it matches the k-mer counts from the sample's reads to the graph k-mers . Based on this, it infers the most likely diploid path in the graph using a Hidden Markov Model (HMM) for the k-mer emissions. Then the sequence denoted by this representative path is used as a "personalised reference", onto which reads can be mapped and novel variants can be called with existing software based on a linear reference. Ideas on how to incorporate a similar HMM-based inference in our method will be discussed at the end of Chapter 3. Importantly, this study showed the value of using variation-aware references for the final aim of genome inference, showing substantially improved genome inference in diverse parts of the MHC region. However, the implementation details were quite specific to the MHC, 4 Mbp in size, and would not scale to whole-genomes due to high memory

[7]article distributed under the CC BY license, which allows reuse with appropriate citation

consumption. [Valenzuela et al., 2015] have also advocated the use of a personalised reference per sample. They showed in a set of simulations that mapping reads to an MSA and extracting the mosaic sequence with highest coverage to be use as reference, improves variant detection.

Another radically different representation for a reference structure is the cactus graph, which can naturally capture general genome rearrangements and the nesting relationships between sequence variants, imposed by large structural events [Paten et al., 2011]. The cactus graph is a generalisation of `POA` that relaxes the directed acyclic graph constraint, by building a representation containing simple cycles only (cycles that are not contained in other larger cycles). It can encode whole-genome alignments for small genomes, but as with `GCSA`, the general graph structure imposes a high memory burden (>150GB to run on mammalian-sized genomes). The same group recognised the instability of the current coordinate system within a data structure that encodes complex rearrangements. They proposed a distinct concept of mapping, where reference bases are unambiguously identified by their flanking sequence instead of positional coordinate [Novak et al., 2015]. These two methods, together with the PRG method described earlier, were evaluated within the `vg` framework in a recent preprint, based on the variant calls they lead to in regions of interest of the human genome [Novak et al., 2017b].

In this section, we described several notable works in the field of graph reference genomes. A few more are available, and a full account of the methods, together with what distinguishes them is given in [Paten et al., 2017a]. Most of these works have been applied in the context of human data, however, solving the problem of encoding multiple references and efficiently integrating various sources of biological information is expected to enable advances in the fields of microbial, viral and plant genomics, metagenomics, phylogenomics, epigenomics, rare diseases and cancer genomics [The Computational Pan-Genomics Consortium, 2016].

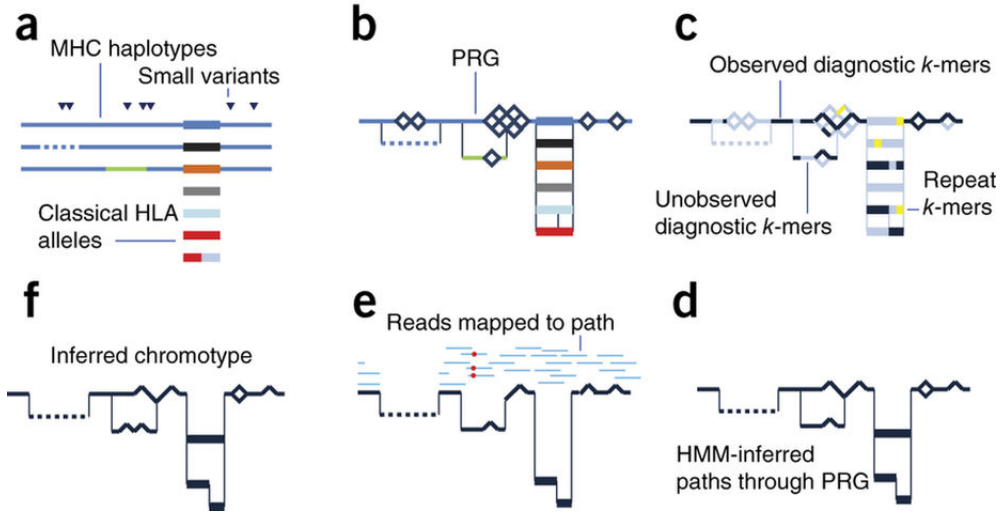


Figure 2.6: Description of PRG for the MHC region. (a) Multiple sources of information about genetic variation, including alternative reference haplotypes (lines), classical HLA alleles (rectangles), and SNPs or short indels (triangles), are aligned. Colors indicate divergent sequence; dashes indicate gaps. (b) A PRG is constructed from the alignment, resulting in a generative model for variation within the region. SNPs, indicated by diamonds, are added as alternative paths to all valid backgrounds (excluding sequence with gaps or a third allele at the position). (c) The PRG is compared to the de Bruijn graph constructed from reads obtained from a sample. Informative k-mers (those that are found at only one level in the PRG) are identified (dark blue). k-mers found elsewhere in the genome (yellow) are ignored. (d) An HMM is used to infer the most likely pair of paths through the PRG, allowing for read errors, resulting in an individualized reference chromotype for the sample. (e) Two haploid genomes are constructed from the reference chromotype, with arbitrary phasing between adjacent bubbles, and reads (light blue lines) from the sample are aligned and assigned (on the basis of mapping quality) to a reference, thus identifying places where the sample contains new variation (red circles; only one path through the chromotype is shown). (f) Newly discovered variants modify the reference chromotype, resulting in the inferred chromotype for the sample. Source: [Dilthey et al., 2015]

2.6 *Plasmodium falciparum* genome

Plasmodium falciparum is an eukaryotic pathogen responsible for the deadliest form of malaria. Its genome comprises 23 Mbp and presents extreme AT bias, with 80.6% AT content overall and 90% in noncoding regions [Gardner et al., 2002]. The unusual AT content results in amplification biases during the sequencing process, producing uneven depth of read coverage across the genome, which challenges the analyses of HTS data. The effect of this will be illustrated in an analysis in Chapter 4. Another complication is given by large tracts of high repetitiveness, with short tandem repeats and low complexity sequence that limit genome accessibility [Gardner et al., 2002], [Zilversmit et al., 2010]. Accessibility is limited further by regions of extreme sequence diversity, whose products are involved in immune evasion. An example is the *var* gene family, previously described in Section 2.2.

These factors have determined a subdivision of the *Plasmodium falciparum* genome into 3 distinct categories: a conserved core genome that is accessible to read alignments (90% of the genome); hypervariable regions that present gaps in read coverage due to poor mappings caused by high divergence from the reference sequence or many paralogous copies (8% of the genome); subtelomeric regions, where read mappings are limited by repetitive sequence (2% of the genome).

Even in the relatively stable core genome, *Plasmodium falciparum* exhibits some unusual features. Indels are unusually abundant compared to other species - there are 3 times more indels than SNPs in non-coding regions and twice as many indels as SNPs in coding regions. Expansions and contractions of short tandem repeats are the source of most indels. Furthermore, despite low overall nucleotide diversity, the core genome presents concentrated variable regions where local diversity is hundreds of times higher. Interestingly, many of these divergent alleles are found at loci coding for surface antigens exposed to the immune system when the parasite is floating within human blood. They will be described in more detail at the beginning of Chapter 5.

The material in this section was compiled from [Miles et al., 2016].

2.7 Thesis overview

The aim of this thesis is to provide an effective method to analyse HTS reads in the context of available catalogues of genomic variation, in order to improve genome inference at loci of high diversity that are inaccessible to current techniques based on a single reference sequence.

In Chapter 3, we detail the FM-index data structure that is used in existing tools of indexing the reference genome, then show how it can be extended to index a richer reference structure in the form of a restricted type of graph that represents alternative sequences at variable loci. Based on this index, we present a variation-aware algorithm to search for sequences in the reference graph. The search results are then used to extract a personalised reference sequence per sample, in the form of a path in the graph highlighted by matched sequencing reads. We couple this personalised reference with existing tools for variant calling. We describe the implementation of the entire approach in a software called `gramtools`.

In Chapter 4, we evaluate the performance of `gramtools` on simulated reads against reference graphs constructed from real variants of the human and *Plasmodium falciparum* genome. We then introduce a number of optimisations designed to improve computational efficiency. We show that `gramtools` can be readily used for *Plasmodium falciparum* data and investigate factors that impact scalability to other species, including the whole human genome.

In Chapter 5, we apply `gramtools` to a large-scale data set of 1300 *Plasmodium falciparum* samples and show that it consistently improves genome inference in two hyper-diverse surface antigenic genes of biological interest. We confirm previous reports of allelic dimorphism in a functional domain of both genes and characterise the

extent of variability they present at a higher resolution than ever before. Based on these results, we perform preliminary population level analyses of these two related genes that offer insights into the mechanisms that shape the variation patterns they exhibit. We find evidence suggesting that the genes undergo concerted evolution and discuss potential immune pressures that could act upon them.

Chapter 3

Gramtools: Genome inference from a population reference graph

3.1 Introduction

In this chapter, we show how to compactly encode a set of genomes, or a reference genome plus known genetic variants, in a way that preserves genomic coordinates and the notion of alternative alleles. This encoding supports read alignment in a manner that is aware of variation, crossing variant site junctions and allowing recombinants to arise naturally. The information collected from the reads enables the inference of a genome that is more similar to the sample studied than the traditional reference genome. Therefore, this inferred genome is expected to be a better substrate for variant calling if used as reference with traditional mapping/calling tools. This entire technique is implemented in our software called `gramtools` and all of its components will be detailed within the chapter.

3.2 Publication and collaboration notes

The work described in this chapter was previously published in [Maciuca et al., 2016] and is mainly the work of the author, under the supervision of her DPhil advisors, Zamin Iqbal and Gil McVean. The vast majority of the `gramtools` code was developed by the author, and all of it was designed/specified by the author. However, some specific sections were developed in collaboration with Carlos del Ojo Elias. Specifically, the function that parses the input reads in BAM/SAM/CRAM or FASTA/FASTQ formats and the function that generates k-mers from a linear PRG were written by Carlos del Ojo Elias. The functionality described in Section 3.7.2, that precalculates and stores index information for small k-mers from the linear PRG, was implemented by both Carlos del Ojo Elias and the author.

3.3 Terminology

A *genome* is defined as a long string with letters $\{A, C, G, T\}$, with chromosomes concatenated for simplicity. A DNA sequencing *read* is a short substring of a genome, which may contain a limited number of errors. *Mapping* refers to the process of aligning reads base-by-base to a reference genome in order to determine their point of origin and highlight any differences. When describing string matching algorithms underlying this process, we will use terms from the computer science literature like 'querying a pattern in text' or 'matching the substring of a string'. In NGS applications, the 'pattern' is a read or a substring of the read and the 'text' is a reference sequence.

A *pan-genome* refers to a representation (with unspecified properties) of a number (greater than 1) of genomes within a species, to be analysed jointly. A *Population Reference Graph (PRG)* is an encoding of a pan-genome that enables matching of sequence data to the datastore, inference of nearest mosaic with the appropriate ploidy

as new reference, and then discovery of new variants not present in the PRG. A *linear PRG* is a PRG that can be represented as a long string, similarly to a genome, but over an extended alphabet containing DNA letters and additional symbols to signal the presence of variants.

A *variant site*, *variation site* or just *site* is a region of the chromosome where there are a number of alternative options for what sequence can be present. These alternatives are termed *alleles* and might be as short as a single character, or could be many hundreds of characters long.

3.4 Background

In this section, we are going to describe some data structures and algorithms that enable efficient search of patterns in a large text. These concepts underlie widely used tools for traditional read mapping to a single reference genome, such as `BWA` [Li and Durbin, 2009] and `Bowtie` [Langmead et al., 2009]. They both index the reference sequence in a compressed way that also enables fast exact matching queries. The problem of exact matching consists of finding the number of occurrences of a word (or pattern) in a text, and locating the position of each occurrence. In bioinformatic language, this is equivalent to mapping reads perfectly to the reference genome. We will first detail the standard exact matching method and then show how it can be extended to handle multiple references.

3.4.1 Notations

Let $T = t_1 t_2 \dots t_n$ be a string, i.e. a concatenation of characters from an alphabet Σ, with character t_i at position i (i.e. $T[i] = t_i$). The size of the alphabet is denoted $|\Sigma|$ and the size of the string is denoted $|T|$, which equals n in this case. $T[i..j]$, or $T_{i,j}$, represents the substring $t_i t_{i+1} \dots t_{j-1} t_j$ of T, induced by the indexes in $[i..j]$.

Substrings of length k will often be referred to as k-mers.

3.4.2 Bit vectors, *rank* and *select* operations

A bit vector is an array $B[1..n]$ of Boolean values, i.e. $B[i] \in \{0, 1\}$. Two elementary operations on B are

$$rank_B(c, i) = |\{i' | 1 \leq i' \leq i, B[i'] = c\}| \tag{3.1}$$

and

$$select_B(c, i) = min\{i' \in [1..n] : rank_B(c, i') = i\}, \tag{3.2}$$

where $c \in \{0, 1\}$. In other words, $rank_B(c, i)$ indicates the number of times character c occurs in B up to position i and $select_B(c, i)$ indicates the index of the i-th occurence of character c in B. These operations can be generalised for non-binary vectors.

In practice, for bit vectors, $rank_B(i)$ and $select_B(i)$ are used to denote $rank_B(1, i)$ and $select_B(1, i)$ respectively. There are compressed bit vector representations [Raman et al., 2002] that can support *rank* and *select* queries in constant time, and they will form the basis of many data structures presented in this thesis.

3.4.3 Burrows-Wheeler Transform

The Burrows-Wheeler Transform (BWT) of a string is a reversible permutation of its characters that was originally developed for compression [Burrows and Wheeler, 1994]. It tends to cluster together identical symbols, so it can be efficiently stored with techniques such as run-length encoding or move-to-front coding[1]. More recently, it has been applied to full-text indexing because it allows the search for a substring in large texts in linear time with respect to the length of the substring, with a small memory footprint. The BWT of a string $T = t_1 t_2 \dots t_n$ is constructed by sorting its n

[1] https://www.cs.helsinki.fi/u/tpkarkka/opetus/12k/dct/lecture08.pdf

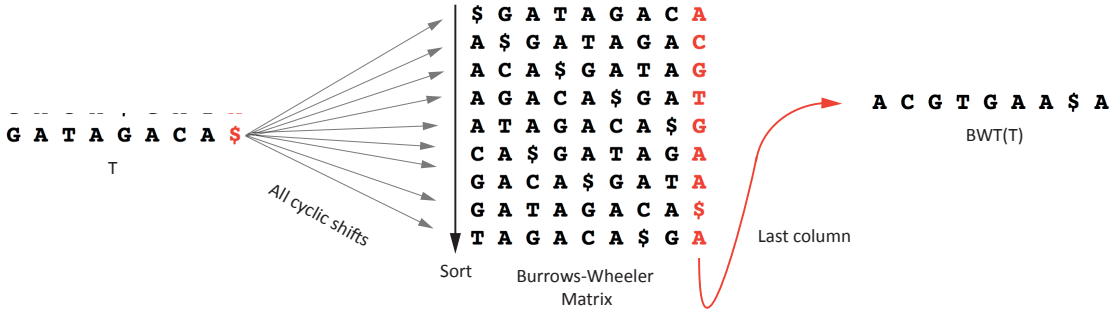


Figure 3.1: Example of BWT construction for the string $T =$ **GATAGACA**. First, take all rotations of the string, i.e. repeatedly move its last letter to the front. Then the string rotations are sorted and the last column of the obtained matrix is the BWT.

cyclic shifts $t_1t_2 \dots t_n$, $t_2 \dots t_nt_1$, $\dots$, $t_nt_1 \dots t_{n-1}$ in lexicographic order. The matrix obtained is called the Burrows-Wheeler matrix (BWM) and the sequence from its last column is the BWT. An example is given in Figure 3.1. The last character t_n is always a unique terminating symbol \$ that is lexicographically smaller than all the symbols in T and is essential for decoding. Storing the first and last column of the BWM is sufficient for finding the number of exact matches of a query substring in T. For locating the position of the matches in T an additional data structure is required, the suffix array. For further details, we recommend you read [Mäkinen et al., 2015].

3.4.4 Suffix Arrays

The suffix array of a string T is an array of integers that provides the starting position of T's suffixes, after they have been ordered lexicographically. Formally, if $T_{i,j}$ is the substring $t_it_{i+1} \dots t_j$ of T and SA is the suffix array of T, then $T_{SA[1],n} < T_{SA[2],n} < \dots < T_{SA[n],n}$ (example in Fig. 3.2). It is related to the BWT, since looking at the substrings preceding the terminating character \$ in the BWM rows gives the suffixes of T in lexicographical order (see Fig. 3.2). In fact, the BWT can be derived in linear time from the suffix array by looping through its values and recording the character

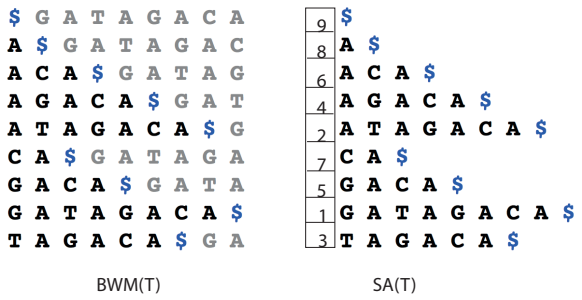


Figure 3.2: Suffix array for the example string $T =$ **GATAGACA** and its relation to the BWT.

occurring just before each suffix in the original text, i.e. $BWT[i] = T[SA[i] - 1]$ if $SA[i] \neq 1$ and $BWT[i] =$ \$ if $SA[i] = 1$. The suffix array coupled with the BWT and two small additional data structures form the FM-index [Ferragina and Manzini, 2000]. The FM-index is a space-efficient data structure that enables fast backward search queries of a pattern P in a text T, in time that is independent of the size of T. To learn more, read [Mäkinen et al., 2015].

3.4.5 Backward search with an FM-index

Any occurrence of a pattern P in text is a prefix for some suffix of T, so all occurrences will be adjacent in the Burrows-Wheeler matrix rows and in the suffix array of T. The reason is that all suffixes starting with P are grouped together by the lexicographical sorting. Therefore, the start positions of matches of P in T will occur in an interval in the suffix array, called a suffix array interval or SA-interval. It consists of a pair of integers $[l(P), r(P)]$ denoting the first and last index in the suffix array that represent a position where P occurs in T.

The backward search starts with the last character of P and successively extends it to longer suffixes, calculating their SA-intervals, until the SA-interval of the entire query is reached. Let $C[a]$ be the total number of occurrences in T of characters smaller than a in the alphabet, the C-array essentially representing the first column

of the BWM, with $C[a]+1$ being the row index of the first suffix starting with a. Then if P' is a suffix of the query P and $[l(P'), r(P')]$ is its corresponding SA-interval, then the search can be extended to aP' by calculating the new SA-interval:

$$l(aP') = C[a] + rank_{BWT}(a, l(P') - 1) + 1 \tag{3.3}$$

$$r(aP') = C[a] + rank_{BWT}(a, r(P)) \tag{3.4}$$

The search starts with the SA-interval of the empty string, $[1, n]$ and successively adds one character of P in backward order. When the search is completed, it returns a SA-interval $[l, r]$ for the entire query P. If $r \geq l$, there are $r - l + 1$ matches for P and their locations in T are given by $SA[i]$ for $l \leq i \leq r$. Otherwise, the pattern does not exist in T. If the C-array and the ranks have already been stored, the backward search can be performed in $O(|P|)$ time in strings with small DNA alphabet. We provide a worked example below.

Suppose the text is $T =$ **GATAGACA** and the query pattern is $P =$ **AGA**. Start with the shortest suffix of P: its last character **A**. Find all rows in the BWM beginning with an **A**, i.e. the SA-interval of **A**. This is straightforward from the C-array (Fig. 3.3).

The next suffix is **GA**. From the rows beginning with an **A**, select the ones having a **G** in the last column, as they indicate **G** precedes **A** in the original text (Fig. 3.4, left). Find the corresponding **G**'s in the first column, such that they point to the same character in the original text. This is done via a key property of the BWM called the LF mapping, which states: the i-th occurrence of a character c in L has the same position in the original text as the i-th occurrence of c in F. Therefore, find the **G**'s in F with equal ranks to the ones in L, 0 and 1 in this case (Fig. 3.4, right). Now, the SA-interval of **GA** has been reached and the same process is repeated to

extend to longer suffixes until the entire P has been matched.

The next suffix is **AGA**. From the rows beginning with **GA**, select the ones having an **A** in the last column, as they indicate **A** precedes **GA** in the original text (Fig. 3.5, left). Find the corresponding **A**'s in the first column, i.e. the **A** with *rank* 2 (Fig. 3.5, right). Now, the SA-interval of **AGA** has been reached ($l = r = 4$) and its size indicates that P has 1 match in T. The position of this match in T is given by the suffix array $SA[4] = 4$ (see Fig. 3.2 again).



Figure 3.3: Backward search example for looking up the substring P = **AGA** in text T = **GATAGACA**. Matching the first character from the end of P, **A**. For simplicity, the entire Burrows-Wheeler matrix is shown, but only its first and last column need to be stored. The numbers in the first column represent the ranks of characters in the column. The numbers in the last column represent the ranks of characters in the original text T.

To summarise, the FM-index stores the first and last column of the Burrows-Wheeler matrix (the C-array and BWT), the suffix array and a matrix Occ containing the ranks of characters in the BWT. Occ is an $n \times |\Sigma|$ matrix where $Occ(i, c) = rank_{BWT}(c, i)$, i.e. the number of occurences of symbol c in $BWT[1, i]$. In practice, only a fraction of the Occ rows are stored and ranks during the backward search are calculated from the nearest available row using the LF mapping. The same technique is used for reducing the cost of storing the suffix array. The first column of the BWM can be compactly represented by the C-array using $|\Sigma|$ integers. As mentioned in Section 3.4.3, the repetitive structure of the BWT makes it very compressible. Therefore,

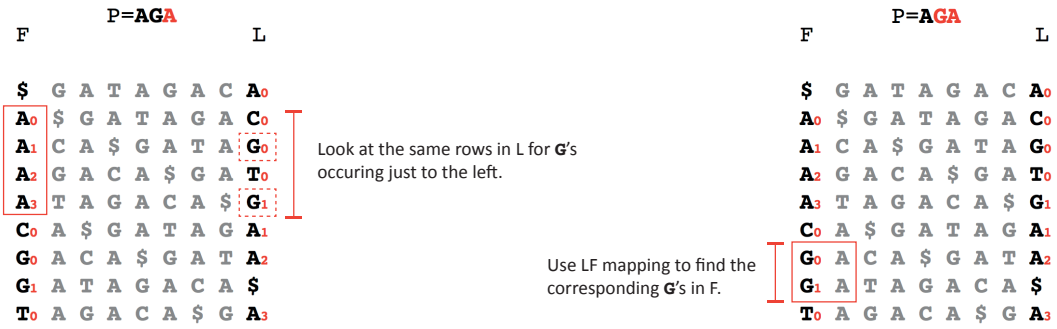


Figure 3.4: Backward search example for looking up the substring $P = \mathbf{AGA}$ in text $T = \mathbf{GATAGACA}$. Matching the second character from the end of P and extending to **GA**. The numbers represent the ranks of characters in the first and last columns respectively.

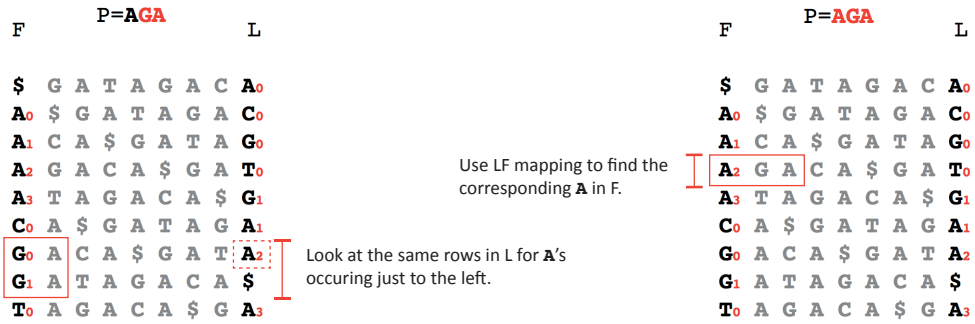


Figure 3.5: Backward search example for looking up the substring $P = \mathbf{AGA}$ in text $T = \mathbf{GATAGACA}$. Matching the third character from the end of P and extending to **AGA**.

the FM-index can perform fast pattern matching with a low memory footprint if the text being searched has a small alphabet.

3.4.6 Wavelet Trees

Rank queries become more computationally expensive as the alphabet becomes larger, since they scale linearly with the alphabet size. The wavelet tree [Grossi et al., 2003] is a data structure designed to store strings with large alphabets efficiently and provide *rank* calculations in logarithmic time. A wavelet tree converts a string into a balanced binary-tree of bit vectors, whose root is built by taking the sorted alphabet and replacing the lower half of smaller symbols with a 0, and the other half of larger symbols with a 1 in the string. This creates ambiguity initially, but at each tree level, each half of the parent node's alphabet is re-split into 2 and re-encoded, so the ambiguity lessens as the tree is traversed in depth. At the leaves, there is no ambiguity at all.

The tree is defined recursively as follows: take the lexicographically ordered alphabet, split it into 2 equal halves; in the string corresponding to the current node (start with original string at root), replace the first half of letters with 0 and the other half with 1; the left child node will contain the 0-encoded symbols and the right child node will contain the 1-encoded symbols, preserving their order from the original string; reapply the first step for each child node recursively until the alphabet left in each node contains only one or two symbols (so a 0 or 1 determines which symbol it is). An example is given in Figure 3.6, originally published in[2] [Maciuca et al., 2016].

In order to answer a *rank* query over the original string with large alphabet, repeated *rank* queries over the bit vectors in the wavelet tree nodes are used as a guide to the subtree that contains the leaf where the queried symbol is non-ambiguously encoded. The *rank* of the queried symbol in this leaf is equal to its *rank* in the original

[2]copyright held by the author

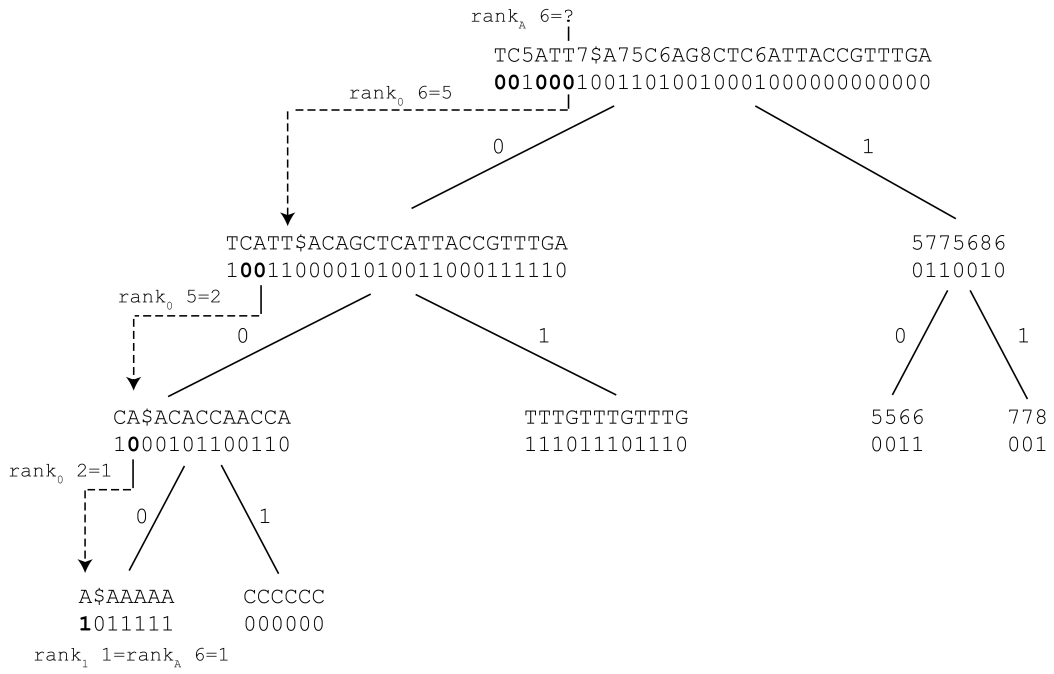


Figure 3.6: Wavelet tree encoding of a string that is the same as the BWT from Fig. 3.9 below. Calculating the *rank* of the marked **A** is performed by repeated *rank* calls moving down the binary tree until the alphabet remaining is just 2 characters. Note that only the bit vectors are stored in the tree, the corresponding strings are only shown here for clarity.

string. The number of *rank* queries needed to reach the leaf is equal to the height of the tree, i.e. $\log_2 |\Sigma|$ if we let Σ be the set of symbols in the alphabet. Computing ranks over binary vectors can be done in constant time, so a *rank* query in a wavelet tree-encoded string has complexity $O(\log_2 |\Sigma|)$. Therefore, using a wavelet tree of the BWT string reduces the time complexity of exact pattern matching to a text with large alphabet from $O(|P||\Sigma|)$ to $O(|P| \log_2 |\Sigma|)$.

In the rest of the chapter, we will show how the data structures described in this section can be used to store known variation inside a reference genome in a way that supports read mapping.

3.5 `gramtools` pipeline overview

The `gramtools` method consists of encoding multiple reference sequences in an FM-index, then using an extension of the backward search algorithm to match reads and inferring a personalised reference composed of best supported alternative alleles. This is done in 3 main stages:

- `gramtools --build`
    - constructs PRG and auxiliary data structures from known variants
    - constructs FM-index for PRG (called vBWT index)
    - calculates an additional k-mer index
- `gramtools --quasimap`
    - exact matching of reads to the PRG
    - converts read information into coverage scores for alternative alleles
- `gramtools --infer`
    - chooses alleles with highest coverage from each variation site

- composes new reference sequence
- runs standard tools to call novel variants on top of inferred reference

The details of each of these steps are described below.

3.6 vBWT data structure

We propose a linear PRG conceptually equivalent to a directed, acyclic, partial order graph, that is generated from a reference sequence and a set of alternative sequences at given variation sites. The graph is linearised into a long string over an alphabet extended with new symbols marking the variants, for which the FM-index can be constructed. Building this data structure requires multiple steps.

1. First, homologous regions of shared sequence between the input reference genomes must be identified. These will act like anchors for the coordinates of the variation sites.

2. Second, for any site between two anchor regions, the set of possible haplotypes must be determined from the input genomes, but they do not need to be aligned. Indels are naturally supported by haplotypes of different lengths.

3. Each variation site is assigned two unique numeric identifiers, one even and one odd, which we call variation markers. The odd identifiers will mark site boundaries and will sometimes be referred to as site markers. The even identifiers will mark alternative allele boundaries and will sometimes be referred to as allele boundary markers.

4. For each variation site, its left anchor is added to the linear PRG, followed by its odd identifier. Then each sequence coming from that site, starting with the reference sequence, is successively added to the linear PRG, followed by the even site identifier, except the last sequence, which is followed by the odd identifier.

5. Convert the linear PRG to integer alphabet ($A \rightarrow 1$, $C \rightarrow 2$, $G \rightarrow 3$, $T \rightarrow 4$, variation site identifiers $\rightarrow$ 5,6,...)

6. The FM-index (suffix array, BWT, wavelet tree over BWT) of the linear PRG is constructed and we will call this the vBWT.

An illustration of these steps on a toy example is given in Figure 3.7, originally published[3] in [Maciuca et al., 2016].



Figure 3.7: A simple PRG linearised according to our encoding. The first site has 3 alleles, which do not here look at all similar, and the second is a SNP. Note that the reference sequences do not need to be aligned. This example will be used in subsequent figures. Source: [Maciuca et al., 2016]

This vBWT construction allows variation-aware mapping through a modified backward search algorithm. The variation markers that surround homologous alleles prevent wrong alignment across their boundaries and ensure the right path is taken when a read crosses a region with multiple alternative sequences. Since these markers are unique to each variation site, we can locate the beginning and end of a site in the Burrows-Wheeler matrix after the permutation of the linear PRG string. This enables reads to cross site junctions and map to the linear PRG when they span multiple sites of variation, allowing for new recombinations of known haplotypes. It also makes the method potentially adjustable for long reads. More importantly, the markers force the ends of alternative sequences coming from the same site to be sorted together in a separate block in the Burrows-Wheeler matrix, even if they do not have high sequence similarity. Therefore, homologous alleles from each site can be queried concurrently instead of looping through every one of them and checking if

[3]copyright held by the author

they match the read, which would happen if the markers were the same for all sites and non-homologous alleles got mixed up.

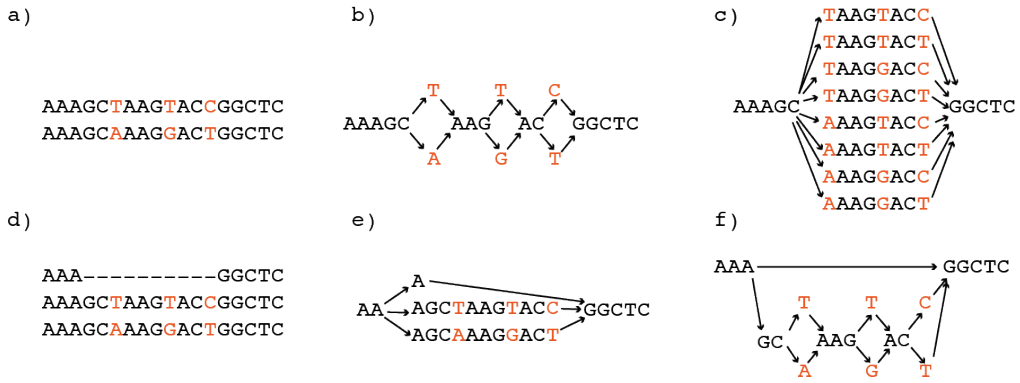


Figure 3.8: PRG graph structure. The sequences shown in a) could be represented either as 3 separate mutations (shown in b)), or enumerated as 8 small haplotypes, shown in c). Both are supported by our encoding. Similarly, the sequences in d) could be represented in our implementation as shown in e). However, we do not support nesting of alleles, as shown in f). Source: [Maciuca et al., 2016]

We show in Figure 3.8a) two sequences which differ by 3 SNPs and give two graph encodings in 3.8b) and 3.8c). Both represent the sequence content equally well, and we allow both. In 3.8d) we have an example where a long deletion lies over two other alleles. We would encode this in our PRG as shown in 3.8e). This works but results in many alternate alleles. An alternative would be to allow nested variation, where variants lie on top of other alleles, as shown in Figure 3.8f). This could be encoded in our system, but we do not allow it for our initial implementation, as it would potentially impact mapping speed. Figure 3.8 was originally published in[4] [Maciuca et al., 2016].

[4]copyright held by the author

3.7 Read quasimapping

3.7.1 Variation-aware exact matching

In this section, we present a modified backward search algorithm for exact matching against the vBWT that is aware of alternative sequence paths. We give a sketch of the pseudocode in Algorithm 1 below, and outline the idea in Fig. 3.9, both published originally in[5] [Maciuca et al., 2016].

When reads align to the non-variable part of the PRG or when an allele is long enough to enclose the entire read, the usual backward search algorithm can be used. Otherwise, when the read must cross variation site junctions, site identifiers and some alternative alleles must be ignored by the search. This means a read can align to multiple substrings of the linear PRG that may not be adjacent in the BWM, so the search can return multiple SA-intervals.

At each step in the backward search, before extending to the next character, we need to check whether the current matched read substring is preceded by a variation marker anywhere in the linear PRG. A scan for symbols larger than 4 (range_search_2d in Algorithm 1) must be performed within the range given by the current SA-interval. With a wavelet tree this range search can be done in $O(d \log(|\Sigma|/d))$ time, where d is the size of the output. If a variation marker is found and it is an odd number, the read is about to cross a site boundary. The suffix array can be queried to find the position of the two odd numbers (start/end of the site) in the linear PRG.

If the search cursor is next to the start of the site, it is just the site marker that needs to be skipped; so the SA-interval (size 1) of the suffix starting with that marker needs to be added to the set of intervals that will be extended with the next character in the read. If the search cursor is next to the end of a site, all alternative alleles from that site need to be queried. Their ends are sorted together in the BWM because of

[5]copyright held by the author

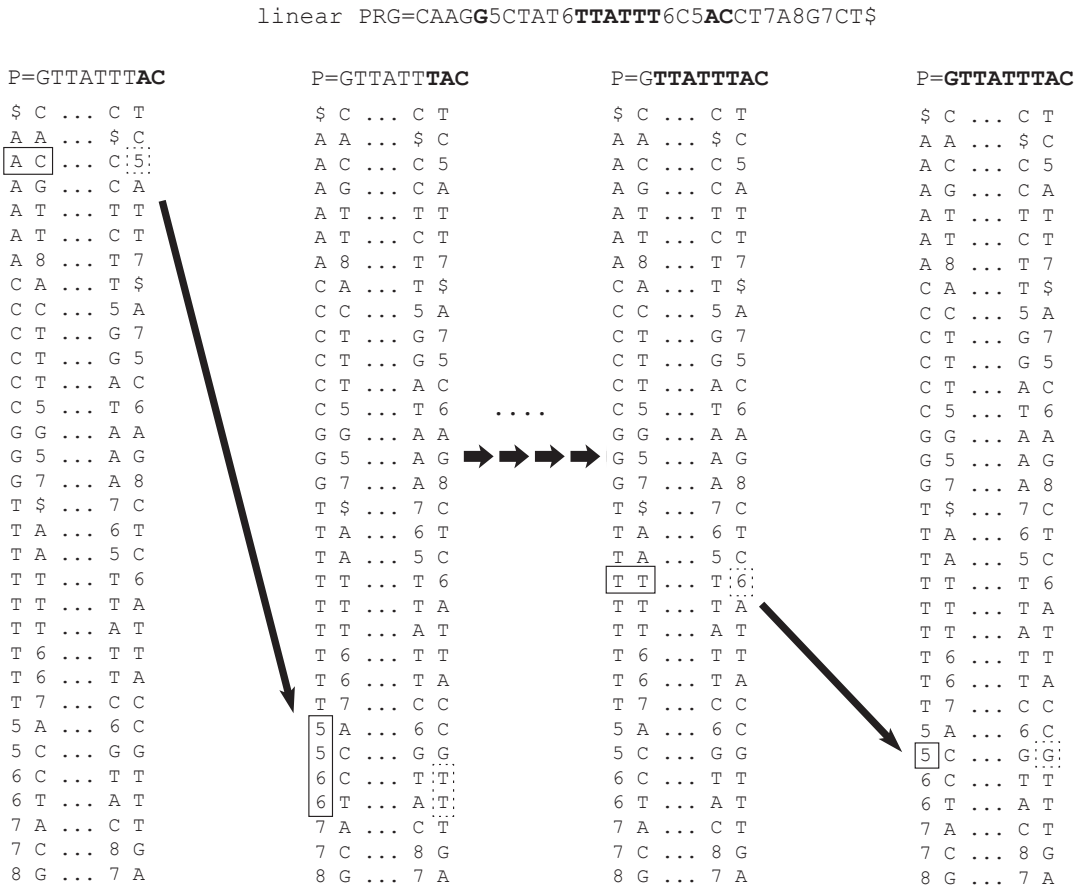


Figure 3.9: Backward search across the vBWT of the linear PRG in Figure 3.7. We start at the right-hand end of the read **GTTATTTAC**, with the character **C**, and as we extend we hit the character **5**, signalling the start or end of a variation site. We check the suffix array to get the coordinate in the linear PRG, and find it is the end. Therefore, the read must now continue into one of the alleles, signalled by the symbol **6**. Continuing in this manner (the shorter arrows signify multiple intermediate steps not shown) we are able to align across the site. Source: [Maciuca et al., 2016]

the markers, so they can be queried concurrently by adding the SA-interval of suffixes starting with all numbers marking that site (even and odd).

Algorithm 1 Variation-aware backward search

Input: pattern $P[1, m]$ and FM-index of PRG in integer alphabet
Output: list of SA intervals corresponding to matches of P

$l \leftarrow C(P[m])$ ▷ l for left
$r \leftarrow C(P[m] + 1)$ ▷ r for right
$i \leftarrow m$
$\texttt{SA_int} = \{[l, r)\}$ ▷ list of SA intervals
$\texttt{Extra_int} = \emptyset$ ▷ Extra intervals
while $i > 1$ and $\texttt{SA_int} \neq \emptyset$ **do**
    for all $[l, r) \in \texttt{SA_int}$ **do**
        $M \leftarrow \texttt{WT.range_search_2d}(l, r - 1, 5, |\Sigma|)$ ▷ find variation site markers
        for all $(idx, num) \in M$ **do** ▷ $idx \in [l, r), num \in [5, |\Sigma|]$
            if $num\%2 = 0$ **then**
                $\texttt{odd_num} = num - 1$
            else
                $\texttt{odd_num} = num$
            if $SA[C(\texttt{odd_num})] < SA[C(\texttt{odd_num}) + 1]$ **then**
                $\texttt{start_site} \leftarrow C(\texttt{odd_num})$, $\texttt{end_site} \leftarrow C(\texttt{odd_num}) + 1$
            else
                $\texttt{start_site} \leftarrow C(\texttt{odd_num}) + 1$, $\texttt{end_site} \leftarrow C(\texttt{odd_num})$
            if $num\%2 = 1$ and $SA[idx] = SA[\texttt{end_site}] + 1$ **then**
                $\texttt{Extra_int} = \texttt{Extra_int} \cup \{[C(num), C(num + 2)]\}$
            else
                $\texttt{Extra_int} = \texttt{Extra_int} \cup \{[C[\texttt{start_site}], C[\texttt{start_site}] + 1]\}$
    $i \leftarrow i - 1$
    $\texttt{SA_int} = \texttt{SA_int} \cup \texttt{Extra_int}$
    for all $[l, r) \in \texttt{SA_int}$ **do**
        $l = C(P[i]) + rank_{BWT}(P[i], l - 1) + 1$
        $r = C(P[i]) + rank_{BWT}(P[i], r) + 1$

If the variation marker found is even, the read is about to cross an allele boundary, which means its current suffix matches the beginning of an alternative allele and the read is about to walk out of a site, so the search cursor needs to jump to the start of site. The odd markers corresponding to that site can be found in the first column of the BWM, and then querying the suffix array decides which one marks the start of site. The SA-interval (size 1) for the BWM row starting with this odd marker

is recorded. Once the check for variation markers is finished and all candidate SA-intervals have been added, each interval can be extended with the next character in the read by using equations 1 and 2.

At the moment, `gramtools` only implements this exact matching algorithm, which does not account for read errors or variants missing from the PRG. Nevertheless, it is set up to support reads with mismatches in the future. Due to the symmetric positional structure of the variant markers in the linear PRG, the algorithm presented in this section can also perform variation-aware search in the forward direction. It would require the FM-index of the reverse string of the linear PRG (note this is different from the reverse complement). Performing backward search of the reverse of the read on the reverse of the linear PRG is equivalent to a forward search of the read in the original linear PRG. Bidirectional indices like the FMD-index [Li, 2012] or the bidirectional wavelet index [Schnattinger et al., 2012], efficiently combine the original and reverse FM-index to synchronise the two types of searches. In this way, forward and backward search can be alternated to extend a pattern match in both the right and left direction. This functionality would enable the search for super-maximal exact matches (SMEMs[6]) [Li, 2012] between the read and the linear PRG. In traditional mapping to a single reference, these SMEMs are used as seeds for inexact string matching, to produce alignments that account for read errors and novel variants [Li, 2013]. In the future, `gramtools` could follow a similar strategy, or a "lightweight alignment" approach that uses the SMEMs to score candidate regions from which the read may have originated without trying to determine its precise location. We discuss this further at the end of the chapter.

[6]refer back to Section 2.4.2

3.7.2 *k*-mer indexing

The algorithm presented in the previous section is bound to be slower than the typical backward search algorithm because of SA-interval splitting. Every time the search cursor reaches the edge of a variation site, the SA-interval splits due to skipping the non-DNA marker. Therefore, at each backward search iteration, a new SA-interval is generated for each variation site the current suffix crosses. As there are multiple SA-intervals to process per iteration, the search becomes slower.

This effect is most obvious in the first few iterations. When small suffixes are searched for, they have many matches in the PRG, so they overlap many variant site junctions. For example, in the first iteration, the first character in the read could be on the edge of a quarter of the total number of variants, thus generating an equal number of SA-intervals. As more characters are matched, the candidate hits narrow down and some of the SA-intervals generated in the previous iteration are discarded. After a few iterations, when the number of SA-intervals lowers, the search is much faster.

To avoid repeating the expensive computations from the first iterations, `gramtools` stores their results and uses them when quasimapping reads. `gramtools` has an option to precompute the index information for small k-mers in the PRG, and store it in a hash table. When a read is quasimapped, first the ending k-mer is looked up in the hash table, its SA-interval is retrieved and extended with the $(k+1)$-th character, thus starting the backward search described in the previous section.

3.8 Personalised reference inference

Every time a read reaches the beginning of an alternative allele (in other words, crossing either an even marker or an odd marker representing the start of a variation site), `gramtools` records a pair with the site and allele number in association with

the current SA-interval. When a read reaches the end of alternative alleles (in other words, crossing an odd marker representing the end of a variation site), `gramtools` records only the site number. Then it waits for the read to reach the other end of the site and match an entire allele before recording the allele number. If the read matching finishes before reaching the start of the variation site (i.e. in the middle of the allele(s)), all the alleles crossed up to that point are recorded.

For recording the crossed alleles, `gramtools` currently queries a mask over the linear PRG that encodes DNA symbols outside variation sites with a 0, site markers with a 0, and DNA symbols from alternative alleles with the respective allele's number within the site. This mask is an integer vector generated together with the linear PRG and we will refer to it as "allele mask". A much more efficient way to encode it would be with a bit mask over the PRG that represents DNA symbols as 0 and variation markers as 1. Then a simple *rank* query over this bit vector could determine the allele's position within the site.

If the PRG contains large variation sites, where alternative alleles can be bigger than the read length, a "site mask" is needed as well. This is a mask over the linear PRG that encodes DNA symbols outside variation sites with a 0, site markers with a 0, and DNA symbols from alternative alleles with the site number. This mask is queried to record the site number for reads that match entirely within an alternative allele and therefore haven't crossed any variation markers during the backward search. Again, a more efficient way of doing this is encoding the site mask as a bit mask over the PRG that represents DNA symbols as 0 and odd variation markers as 1. Then a simple *rank* query over this bit vector can determine whether the read lies within a variation site and if so, which one.

After quasimapping all reads, `gramtools` aggregates the information about the variants they crossed to calculate a coverage score for each alternative allele at each site. The scoring system is the following:

- if the read has a unique match, add +1 to each site/allele crossed (Fig. 3.10a)
- if the read has a unique "horizontal" match and multiple "vertical" matches, i.e. it quasimaps uniquely but starts or ends inside a site and the beginning/ending of the read sequence overlaps multiple alternative alleles, add +1 to each allele crossed (Fig. 3.10b)
- if the read has multiple "horizontal" matches that are unique "vertically", i.e. it quasimaps to multiple regions in the genome, but crosses a single allele at each site, add $+1/n_r$ to each allele crossed; n_r is the number of regions in the genome (variable or non-variable) matched (Fig. 3.10c)
- if the read has multiple "horizontal" matches out of which some are non-unique "vertically" as well, add $+1/n_r$ to each allele crossed; n_r is the number of "horizontal" matches (Fig. 3.10d)

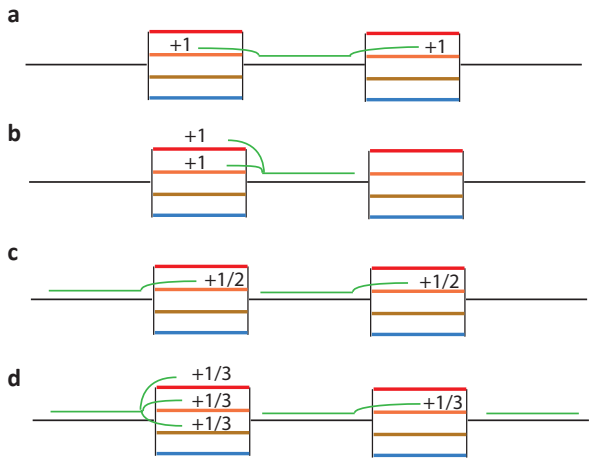


Figure 3.10: Toy example of a PRG and the different ways a read can quasimap to it and contribute to the coverage scoring of alternative alleles.

Finally, the linear PRG is traversed and at each variation site encountered, the allele with the highest coverage score is selected. Thus, `gramtools` composes a se-

quence from the path in the PRG with highest read support and this can be used as a personalised reference for the sample under analysis.

3.9 Remapping to graph-inferred reference and variant discovery

A second round of full mapping to the new, personalised reference is performed in order to detect variants that were not present in the PRG. This works well if the personalised reference is within a small edit distance from the reads and the novel variants are spread out in the genome (Fig. 3.11a). Since `gramtools` infers the new reference only from reads that match exactly in the PRG, novel variants within a smaller distance than the read length can create problems because no reads will quasimap in the region spanning the variants. If the respective region is diverse and overlaps a variation site in the PRG, `gramtools` will choose an arbitrary allele that can end up being very divergent from the true allele, resulting in a gap in sequence inference (Fig. 3.11b). Clustered novel variants in the non-variable region of the PRG will have the same fate as in standard methods based on a single reference.

Novel variants separated by a distance larger than the read length are less problematic because reads quasimapping around and between them should inform the inference of a good reference for variant calling (Fig. 3.11c). However, there is still a coverage loss in the region spanning the variants, which could affect inference in samples sequenced at low depth. Sequencing errors could also amplify the coveage loss. This will be discussed more at the end of the chapter.

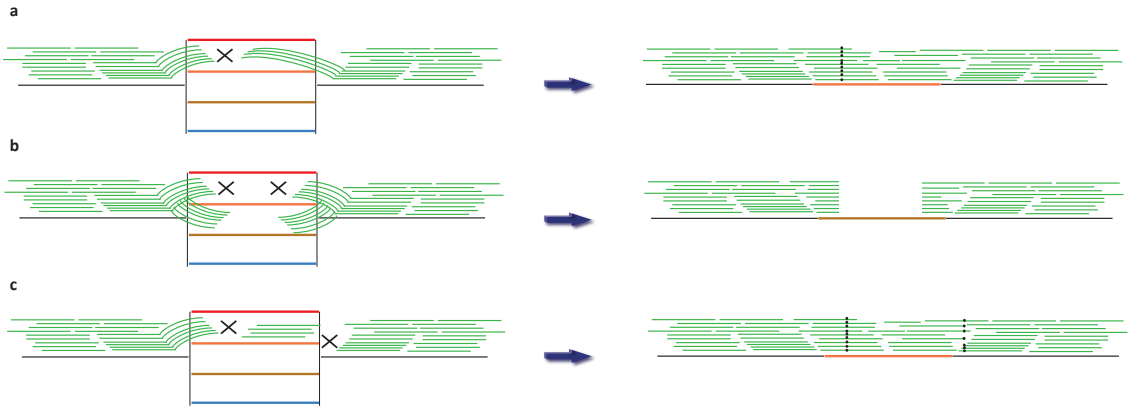


Figure 3.11: Schematic figures of how novel variants on top of the PRG (marked with an $\times$) can affect inference in the context of reads containing the variants failing to quasimap. On the left, toy examples of read mappings to the PRG; on the right, toy examples of resulting read pileups over the PRG-inferred reference. In a), there is a single SNP on top of an alternative allele. This produces a coverage loss, but the correct allele is likely to be selected based on quasimapped reads surrounding the SNP, enabling its discovery in the final step of full mapping. In b), there are 2 novel SNPs, separated by a distance smaller than the read length. No reads will quasimap in the region spanned by the SNPs within the alternative alleles. Hence, any of them could be selected arbitrarily based on reads quasimapping in the margins delimited by the SNPs. This can produce a gap in the pileup with respect to the inferred reference. In c), there are 2 novel SNPs separated by a distance larger than the read length. There will be a coverage loss in the region spanned by the SNPs, but some reads will quasimap in between, informing the choice of a close allele. The SNPs can then be discovered in the step of remapping to the personalised reference.

3.10 Implementation

3.10.1 Software design

A diagram of the flow of data through the `gramtools` pipeline is presented in Fig. 3.12. The pipeline starts with building the linear PRG and the associated mask vectors from a reference genome (in FASTA format) and previously discovered variants (in VCF format). We have an alternative script that builds the linear PRG from a multiple sequence alignment, available on request. The vBWT index for the linear PRG is constructed using the `SDSL` library [Gog et al., 2014]. Then k-mers are generated from the linear PRG for a specified k. Their index information (SA-intervals, sites and alleles overlapped) is precomputed by exact matching to the linear PRG. All these steps are part of the `--build` stage and their results can be cached and reused for many samples.

The `--quasimap` stage starts after a preprocessing step of read quality trimming; we use `seqtk` (`https://github.com/lh3/seqtk`). Trimmed reads are quasimapped to the PRG by querying the k-mer index and extending via the variation-aware backward search algorithm. At each variation site, alternative alleles are assigned a coverage score based on the reads overlapping them.

The `--infer` stage starts with choosing the allele with highest coverage score from each variation site. The alleles are concatenated with the interleaving non-variable sequences and a personalised reference is obtained. Now, reads can be remapped with `BWA-MEM` [Li, 2013] to this new reference and novel variants can be called via `SAMtools` [Li et al., 2009], or any other variant caller. The final output files are in SAM/BAM and VCF format with respect to the personalised reference.

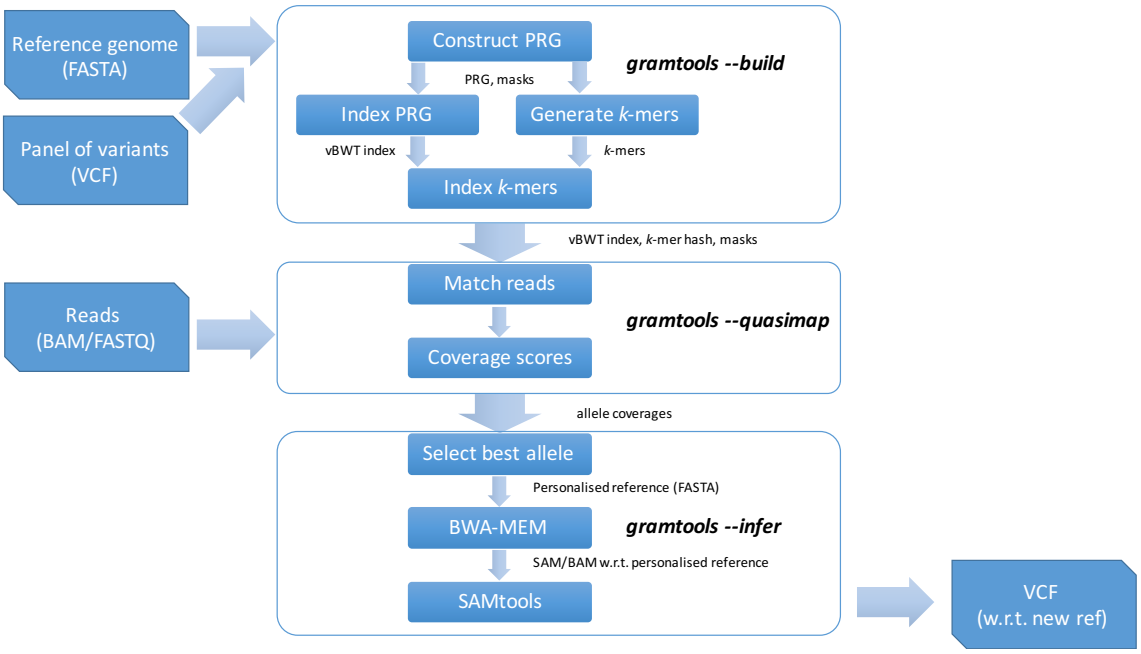


Figure 3.12: Diagram of the data flow through the `gramtools` pipeline.

3.10.2 Implementation details

The `gramtools` implementation is written in C++ and is freely available on GitHub at https://github.com/iqbal-lab-org/gramtools. It has the following dependencies:

- `HTSlib` - for parsing BAM/SAM/CRAM or FASTQ/FASTA file formats
- `zlib` - for reading compressed files
- `GoogleTest` - as a unit testing framework
- `Boost` - to encode a hash table of k-mers
- `SDSL` - to encode components of the vBWT index

The vBWT index uses the following `SDSL` data structures: a compressed suffix array based on a wavelet tree over the BWT (`csa_wt<>`); a balanced-shaped wavelet tree with integer alphabet (`wt_int<>`); fast uncompressed bit vectors (`bit_vector`).

The k-mer indexing step is multi-threaded using pthreads. The read quasimapping step is not currently multi-threaded, but it is an upcoming feature. Paired-end read information is not supported yet, but is straight-forward to implement since our approach preserves genomic coordinates.

3.11 Limitations

There are a number of limitations present in our current approach. Some are at the conceptual level, imposed by the underlying data structures and algorithms, and some are at the implementation level, imposed by our simplified encoding choices within `gramtools` . The latter are more straight-forward to overcome, though possibly at a performance cost. In any case, we discuss both types.

From the data structures perspective, our linearised representation of the PRG imposes restrictions on the size and type of variants that can be used as reference. Structural variants like copy-number variants, inversions and translocations cannot be naturally represented in a linear PRG. In theory, one could list the alternative sequences implied by these types of variants and represent them as a large variation site, but the variant size poses an additional problem. The region covered by such large variants is likely to contain other smaller variants in the population. Since our linear PRG format requires all overlapping variants to be represented by a single site whose coordinates are defined by the largest variants, structural variants can increase the size of the data structure considerably. A potential way to mitigate this is allowing nested sites in the linear PRG (Fig. 3.8f). As long as at each level of nesting, the inner site is fully enclosed in the outer site, our variation-aware backward search algorithm can be adapted to the new linear PRG format. It would require to store a stack recording identifiers for the sites crossed as a read goes in and out of different layers of nesting. It is unclear what cost this modification would induce on quasimapping

speed and it is worth further investigation.

From the algorithm perspective, the main limitation is that only reads that perfectly match the linear PRG contribute to the personalised reference inference. This means that reads containing sequencing errors or novel SNPs are lost. However, the error rates in Illumina data are very low. Illumina typically reports $\geq 75\%$ of bases having a quality score of $Q \geq 30$, which implies at most an error every 1000 bases. This would result in `gramtools` losing at most 10% of coverage with typical 100 bp reads. However, bases with lower quality scores are much more likely to be clustered towards the end of the read and be removed by the quality trimming preprocessing step.

As detailed in Section 3.9, the exact matching restriction can also create problems in samples containing variants that are not present in the PRG. Another way to look at it is that it requires the PRG to be rich enough such that it maximizes the proportion of reads mapping to it with at most 1 mismatch. In some applications, there might not be enough reference data available to fulfill this requirement.

In order to overcome the exact matching limitation, `gramtools` is set up to support bidirectional search in order to find SMEMs[7] between the reads and the PRG, which serve as seeds for full aligners like `BWA-MEM`. We do not believe that fully aligning the reads base-by-base in the traditional way is necessary, since we do not attempt to call variants directly from the PRG. The aim of the PRG mapping step is simply to find the alleles best supported by the reads and infer a reference close to the sample. We believe an approach similar to the alignment-free methods developed for RNA-Seq data (lightweight alignment/pseudoalignment/quasimapping) are best suited to this purpose, since they quickly determine the set of transcripts compatible with the reads by avoiding the full alignment computation [Patro et al., 2015], [Patro et al., 2017], [Srivastava et al., 2016], [Bray et al., 2016], [Patro et al., 2014].

[7] super maximal exact matches; refer back to Section 3.7.1 and Section 2.4.2

The allele coverage scoring can also be somewhat problematic in PRGs with long alternative alleles and samples with novel variants. This is due to reads contributing to the coverage score equally, regardless of the length of allele overlap. Reads containing novel variants are lost in the quasimapping process, but resulting gaps in the allele coverages won't be visible until the final stage of remapping to the inferred reference. Nevertheless, improving the coverage scoring is only worthwhile once reads that match the PRG imperfectly can be taken into account. The incorporation of lightweight read alignment should be coupled to the design of a more sophisticated coverage scoring system that accounts for the percentage of allele length that is covered by SMEMs.

Finally, one major limitation of `gramtools` is that it only supports haploid genomes at the moment. For multiploid genomes, the `--infer` stage would need to build a multiploid reference for the final variant discovery step. This is currently unavailable because haplotype phasing information contained in the reads is not used in allele scoring. Hence, each site in the PRG is genotyped independently and the phase of nearby variants is lost. For the final step of remapping and variant discovery to work, the personalised reference needs to be phased up to the read length. `gramtools` is set up to support such inference, since it keeps track of the sites and alleles crossed by the reads during quasimapping. The directed, acyclic structure of the PRG makes it a natural backbone for an HMM that infers the most likely multiploid path in the graph given a sample's reads [Dilthey et al., 2015].

Chapter 4

Gramtools performance and optimisations

4.1 Introduction

In the previous chapter, we described a data structure that encodes a population reference graph and associated algorithms for mapping sequencing reads along alternative variant paths in the graph. We presented an initial implementation of these concepts in a software called `gramtools`. In this chapter, we first design a set of experiments to measure the performance of `gramtools` and its computing resource requirements. Then, we introduce a number of algorithm optimisations to improve execution time and study the impact of some input parameter choices on performance. At the end, we profile the final version of the code and discuss future optimisations that could be incorporated.

4.2 Collaboration note

The analyses described in this chapter are solely the work of the author and all of the underlying `gramtools` code used to run them was designed/specified by her. However,

some specific sections of the code were written by Robyn Ffrancon, under the author's direction. The function that performs a search for variant markers in a SA-interval, for the optimisation described in Section 4.4.2, was written by Robyn Ffrancon. The code that generates k-mers neighbouring variants from the linear PRG, used for the analyses in Section 4.4.3 was written by Robyn Ffrancon.

4.3 Initial performance measures

4.3.1 Graph and index construction

This section contains performance measurements for the *–build* step of `gramtools`, specifically the construction of the population reference graph and the vBWT index. The graphs in this section will be used as references for benchmarking the *–quasimap* step in the rest of the chapter.

These are simple graphs, with mostly biallelic sites, constructed with a script that parses a VCF file and inserts the variants into the reference genome at their respective coordinates, enclosed by their numeric markers. Variants that are overlapping adjacent variants that have already been added to the linear PRG are discarded[1]. Variants occuring at consecutive genomic coordinates were merged into a single variant site, with all haplotypic combinations enumerated.

In the following subsections, we measure the time and amount of RAM required to build the linear PRG and its FM-index for *Plasmodium falciparum* and human genome variation.

[1] In Chapter 5, we will describe an alternative graph construction method that can handle variant regions of high complexity, where the VCF may contain many overlapping or densely clustered variants.

4.3.1.1 Whole-genome *Plasmodium falciparum*

We constructed a linear PRG from the 3D7 reference genome of *Plasmodium falciparum*, 22.9 Mb in size, and the whole-genome callset (44,439 variants) obtained by `Cortex` from a genetic cross - 7G8xGB4 [Miles et al., 2016]. The linear PRG was generated in 6.1 seconds and it contains 23,597,628 symbols over an alphabet of size 88,581.

Figure 4.1 shows the time and amount of memory required to build each component (suffix array, Burrows-Wheeler transform and wavelet tree) of the FM-index for this linear PRG. In total, the FM-index construction took 53 seconds, reaching peak RAM usage of 191 MB during the wavelet tree step.

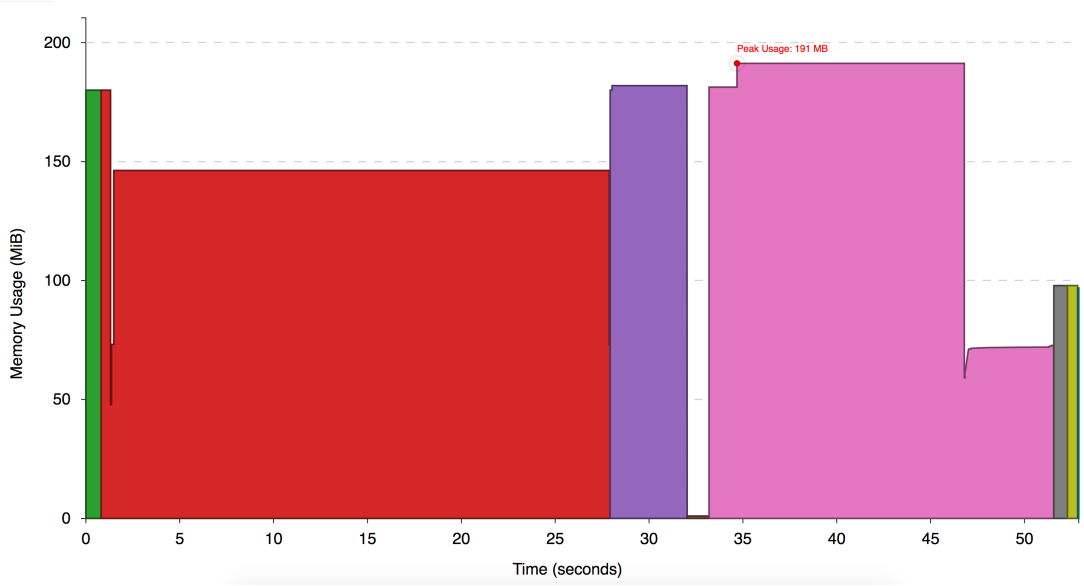


Figure 4.1: Time and memory requirements to construct the FM-index for the *Plasmodium falciparum* PRG. Coloured blocks represent parsing the linear PRG (green), constructing the suffix array (red), BWT (purple), wavelet tree (pink), sampling from the suffix array (grey), sampling from the inverse suffix array (olive).

As part of the *−build* step, `gramtools` also generates all k-mers in the PRG, for the suffix array intervals precalculation (see Chapter 3, Section 3.7.2). Since this particular PRG is sparse, we can make this computation more efficient by choosing to precalculate the SA-intervals only for k-mers overlapping variant sites, since we are

only interested in reads overlapping variants; the rest of reads do not contribute to allele coverages and inference of a new reference. For $k \leq 8$, the k-mers overlapping variants saturate all the k-mers in the PRG, so we initially chose $k = 9$, because the number of 9-mers overlapping variants (99,333) is considerably lower that the total number of 9-mers in the PRG (242,720). Unless otherwise specified, the quasimapping experiments based on this PRG are run with paramerer $k = 9$. We will further discuss in Section 4.4.3 the effect of this parameter choice. Generating all 9-mers overlapping variant sites in the current linear PRG took 4.6 seconds.

4.3.1.2 Whole-genome human

We constructed a linear PRG from the human reference genome GRC37, 3 GB in size, and the Phase 3 whole-genome callset from the 1000 Genomes Project [The 1000 Genomes Project Consortium, 2015]. We excluded variants that did not have specified sequence for the alternative alleles (mostly copy-number variants) and variants that had a frequency below 5%. The VCF filtering process resulted in a set of 7.4 million SNPs and 978,000 indels. The linear PRG contains 3,130,646,843 symbols over an alphabet of size 16,397,637. The graph was constructed from the VCF file in 12 minutes and contains 4,193,538 11-mers in total. Figure 4.2 shows the time and amount of memory required to build each component (suffix array, Burrows-Wheeler transform and wavelet tree) of the FM-index for the linear PRG. In total, the FM-index construction took 4h and 24 minutes, reaching peak RAM usage during the suffix array step, 25GB. Comparative figures for running `BWBBLE` [Huang et al., 2013] on the same input data set are given in Table 4.1, originally published in[2] [Maciuca et al., 2016].

The reasoning behind the exclusion of rare variants from the PRG is that graph quasimapping is followed by the *–infer* step of full mapping to the inferred person-

[2]copyright held by the author

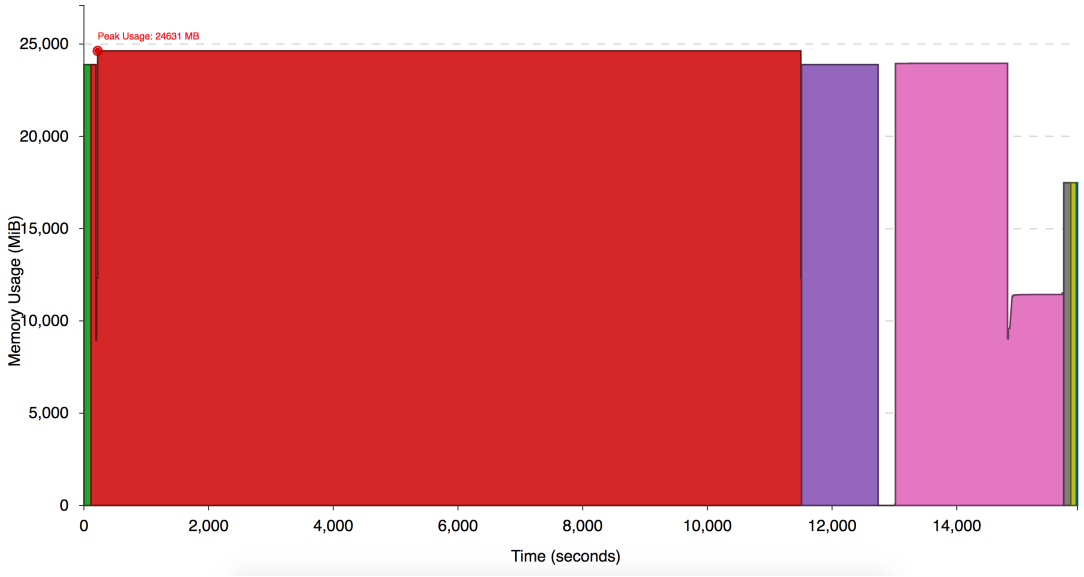


Figure 4.2: Time and memory requirements to construct the FM-index for the human PRG with common variants from 1000 Genomes Project. Coloured blocks represent parsing the linear PRG (green), constructing the suffix array (red), BWT (purple), wavelet tree (pink), sampling from the suffix array (grey), sampling from the inverse suffix array (olive).

Table 4.1: Comparison of `gramtools` and `BWBBLE` for FM-index construction costs and final data structure size for human PRG with common variants from 1000 Genomes Project.

Software	Peak Memory(GB)	Time (hrs/mins)	Final/in-use memory (GB)
`gramtools`	25	4h24m	17.5
`BWBBLE`	60	1h5m	12

alised reference. The aim of the *–quasimap* step is to enable the inference of a reference sequence from the graph that is close enough to the sample's reads such that standard variant calling tools work well. Therefore, only clustered variants affect the inference of a personalised reference; since rare SNPs are likely to be isolated and mapping across a single SNP works well, there is limited benefit in adding them to the PRG. On the other hand, other graph genome tools [Sirén, 2017], [Novak et al., 2015], [vg] attempt to call variants directly from the graph, so they aim to incorporate in the graph all the information available about genetic variants.

For the purpose of comparing with alternative tools, we measured the graph construction costs for the human genome with the full callset from 1000 Genomes Phase 3 VCF, without filtering for allele frequency. This resulted in a set of 81,443,073 SNPs and 3,299,133 indels. The linear PRG contains 4,566,662,020 symbols over an alphabet of size 158,548,838. The step of graph construction from the VCF file took 30 minutes to complete. Figure 4.3 shows the time and amount of memory required to build each component (suffix array, Burrows-Wheeler transform and wavelet tree) of the FM-index for the linear PRG. In total, the FM-index construction took 5h and 48 minutes on a single core, reaching peak RAM usage during the suffix array step, 27GB.

The `GCSA2` publication [Sirén, 2017] reports building a graph index for the same input data set in 14.1 hours on 32 cores, with peak RAM usage 52.3 GB. Moreover, it is using over 2 TB of I/O to fast disk, so they recommend using it on a system with at least 96 GB of memory, so that the written files remain in disk cache, allowing their fast load into memory. `gramtools` uses less RAM resources and has no significant I/O burden, but the comparison is not straightforward because the two approaches encode different types of graphs. Our vBWT structure only supports directed acyclic graphs, whereas `GCSA2` encodes a more general class of graphs. On the other hand, because `GCSA2` relies on a de Bruijn graph, it only supports pattern queries up to

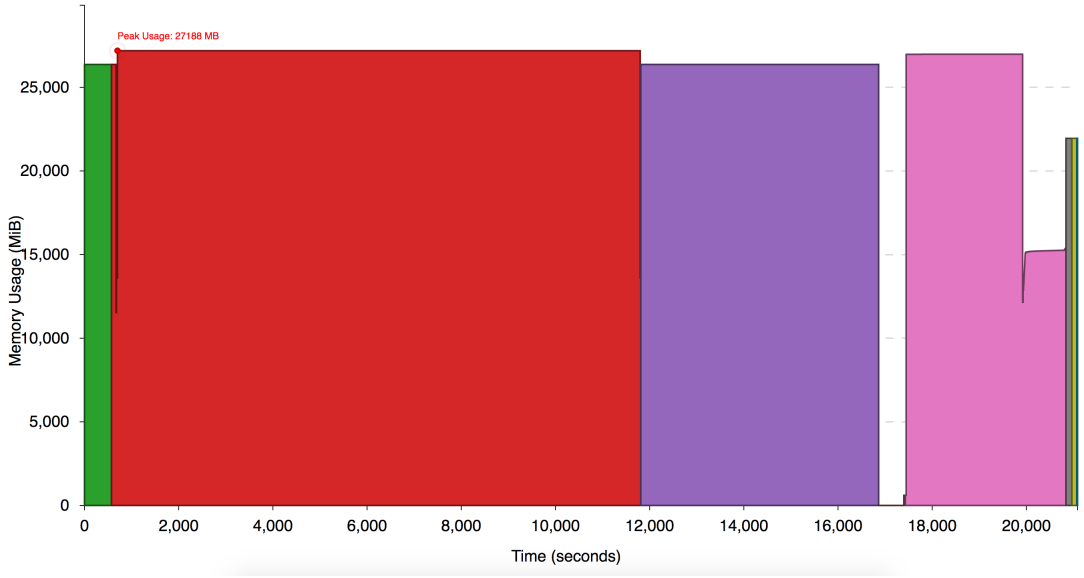


Figure 4.3: Time and memory requirements to construct the FM-index for the human PRG with full 1000 Genomes variant set. Coloured blocks represent parsing the linear PRG (green), constructing the suffix array (red), BWT (purple), wavelet tree (pink), sampling from the suffix array (grey), sampling from the inverse suffix array (olive).

a certain length k (the order of the de Bruijn graph), whereas `gramtools` supports pattern queries of any length. We will discuss this more at the end of the thesis.

We will see in the following section that the small amount of memory used by `gramtools` comes at the cost of slow runtimes. In the following sections we will measure the impact of input choices on the program's speed, introduce optimisations at the data structures and algorithms level, at the source code level and the build level, and measure the improvements.

4.3.2 Quasimapping runtime

The performance of the `gramtools` *–quasimap* step was measured on reads simulated from the PRG. We provide a script within `gramtools` that takes a linear PRG as input and returns a single genome corresponding to a path in the graph, together with reads simulated uniformly from this genome. We used this script to generate 10,000 error-free reads of 150 bp in length from an arbitrary path in the linear PRGs described

Table 4.2: Runtimes for quasimapping 10,000 simulated reads of length 150 bp to the *Plasmodium falciparum* PRG, with and without the *popcnt* CPU instruction enabled during `gramtools` build.

	Time (sec)	Speed (reads/sec)
with *popcnt* (baseline)	486	20.5
without *popcnt*	763	13

in sections 4.3.1.1 and 4.3.1.2, for *Plasmodium falciparum* and human genomes. We quasimapped these reads back to the linear PRGs (with `gramtools` version from commit a510cbd and $k = 9$, as described in Section 4.3.1) and measured runtimes (Table 4.2).

This experiment was run single-threaded on an Amazon web services i3.2xlarge virtual machine with 8vCPUs Intel(R) Xeon(R) CPU E5-2686 v4 @ 2.30GHz and 61 GiB memory (Machine A in Table 4.3). Unless otherwise specified, all experiments in this chapter are run on this machine. The system was running Debian 8.8 on Linux kernel 3.16.0. All code was compiled with gcc/g++ version 4.9.2.

Our CPU architecture supports the *popcnt* instruction (count number of bits equal to 1) available in the SSE4.2 instruction set, which increases the performance of the FM-index queries from `SDSL`. However, this instruction is only enabled when the GCC flags $-march = native$ and/or $-msse4.2$ are passed on during the `gramtools` build, so we can measure its impact on `gramtools` performance (Table 4.2). All remaining experiments in the rest of this chapter are run with the *popcnt* instruction activated.

Note on computer specifications. In some sections throughout the chapter, different machines were used for experiments. It will be mentioned when that is the case. Their specifications are given in Table 4.3. Our observations indicate that `gramtools` performs very similarly on Machines A and B.

For the human PRG, read quasimapping times could not be measured because the preceding step of calculating suffix array intervals for small k-mers was extremely slow and could not be completed. Even in the case of the small PRG with common variants

Table 4.3: Hardware specifications for machines used in experiments.

Label	Specification
Machine A	AWS i3.2xlarge virtual machine with 8vCPUs Intel(R) Xeon(R) CPU E5-2686 v4 @ 2.30GHz and 61 GiB memory
Machine B	AWS i3.8xlarge virtual machine with 32vCPUs Intel(R) Xeon(R) CPU E5-2686 v4 @2.30GHz and 244 GiB memory
Machine C	European Bioinformatics Institute hybrid cluster

only, `gramtools` took 3 minutes to process each 11-mer. At 4,193,538 11-mers, this is a prohibitively expensive computational step.

We profiled the quasimapping code with Linux `perf` tools, which reported that most of CPU time was spent in the `SDSL` functions rank_support_v5[3] (55.5%) and wt_int[4](6%). This is due to the large alphabet in our linear PRG, which makes the *rank* queries required by the backward search extension much slower than in a standard FM-index over DNA alphabet. In addition, at each step in the backward search operation, a range search is performed over the wavelet tree (line 8 in Algorithm 1), in order to find symbols marking variant sites in the BWT. The range search implies repeated *rank* queries over a large alphabet as well. In the following sections, we propose alternative ways of computing the ranks and the range search, which reduce the impact of the extended alphabet.

4.4 Quasimapping speed optimizations

4.4.1 Ranks storage

In a standard FM-index over the 4-letter DNA alphabet, a sample of the rows in the Occ matrix (where $O(i, c)$ is the number of occurences of symbol c in $BWT[1, i]$,

[3] https://github.com/simongog/sdsl-lite/blob/master/include/sdsl/rank_support_v5.hpp

[4] https://github.com/simongog/sdsl-lite/blob/master/include/sdsl/wt_int.hpp

see last paragraph of Chapter 3, Section 3.4.5) is stored for fast *rank* queries during backward search. This requires $4n\lceil \log_2 n \rceil b$ bits, where n is the genome size and b is the proportion of rows sampled. The value of $O(i, c)$ for a row that was not stored is computed on the fly from the closest stored position, so there is a trade-off between the memory requirement of storing the *Occ* matrix and its access time for a *rank* query. The FM-index based on an integer alphabet wavelet tree (*wt_int*) in `SDSL` does not store any ranks, since it was designed for large alphabets that make the *Occ* matrix too large for storage ($|\Sigma| n \lceil \log_2 n \rceil b$ bits). Therefore, in our initial `gramtools` implementation, ranks are computed on the fly from the wavelet tree at each backward search extension (lines 25-26 in Algorithm 1). Despite the fact that the wavelet tree enables efficient *rank* computation that scales with the $\log_2$ of the alphabet size $|\Sigma|$ (see Chapter 3, Section 3.4.6), results in section 4.3.2 showed that the `gramtools` quasimapping time is longer than desirable and that more than half of time is spent computing ranks.

We added to `gramtools` a function that precomputes the *Occ* matrix for the A,C,G,T letters only. In spite of the large alphabet of our linear PRG, only the ranks of DNA letters are needed for variation-aware backward search ($P[i]$ in lines 25-26 in Algorithm 1). The intuition behind this is the following: ranks are used to jump from a position in the last column of the Burrows-Wheeler matrix (BWM) to one in the first column (see Chapter 3, Section 3.4.5), while ensuring both point to the same character c in the original linear PRG string; in other words, the *rank* query identifies the correct c from the run of c's in the first column, the one that corresponds to the c in the last column. This ensures that the symbols matched via backward search in the BWM are adjacent in the linear PRG. However, in the case of non-DNA symbols that mark boundaries between alternative alleles, there is no need for this adjacency requirement. That is because when a variant marker is found during the backward search, entire portions of the linear PRG (corresponding to non-matching alleles) are

Table 4.4: Runtimes for quasimapping 10,000 simulated reads of length 150 bp to the *Plasmodium falciparum* PRG with the version of `gramtools` code that precomputes and stores the *rank* matrix *Occ* and with the baseline version that calculates ranks from the wavelet tree instead. The one-time precalculation step consisted of computing the SA-intervals for 9-mers overlapping variant sites in the PRG using 7 threads.

`gramtools` version	Qmap time (sec)	Qmap speed (reads/sec)	Precalc time (min)
with stored ranks	345	29	190
without stored ranks (baseline)	486	20.5	215

skipped. A variant site marker from the last column of the BWM is mapped in the first column to either the entire run of markers from that specific site or just to the marker signalling the beginning of site. Therefore, ranks of variant markers are never needed in the variation-aware backward search.

Storing the entire *Occ* matrix for our linear PRG requires $4n'\lceil \log_2 n' \rceil$ bits, where n' is the length of the linear PRG. For the whole-genome PRG of *Plasmodium falciparum* (described in Section 4.3.1.1), this amounts to 57.5 Mb, so there is no need to sample the rows. We modified the backward search function in `gramtools` to access this matrix instead of using the `SDSL` *rank* query over the wavelet tree and we performed the same experiment described in Section 4.3.2. Resulting runtimes are shown in Table 4.4, together with baseline runtimes from Section 4.3.2 for comparison. To generate these results, we ran the `gramtools` version from commit 82834b2 on Machine A.

4.4.2 BWT bit mask

After *rank* queries, the most computationally expensive component of `gramtools` was scanning the BWT for symbols representing variant markers. At each iteration in the backward search, before extending to the next character, we need to check whether the current matched read substring is preceded by a variant marker anywhere in the linear PRG. A scan for symbols larger than 4 (*range_search_2d*, line 8 in Algorithm 1)

must be performed within the range given by the current SA-interval. With a wavelet tree this range search is done in $O(d \log(|\Sigma|/d))$ time (where d is the size of the output) and implies repeated *rank* queries over the tree nodes [Mäkinen et al., 2015].

Alternatively, the range search can be done with a bit mask over the BWT instead of the wavelet tree. The BWT can be encoded in a bit vector, where 0 represents DNA symbols $(1-4)$ and 1 represents variant markers (≥ 5). Then the range search for non-DNA symbols can be done with *rank* and select queries over this bit vector, which are much faster than the ones over a wavelet tree because of the reduced alphabet size. The `SDSL` *bit_vector* encoding supports *rank* and select queries in constant time and uses $64\lceil \frac{n'}{64} + 1 \rceil$ bits, where n' is the length of the linear PRG. The pseudocode for this alternative range search is given in Algorithm 2.

Algorithm 2 Range search for variant markers on BWT

Input: suffix array interval borders sa_int_start and sa_int_end, BWT string bwt
Output: array $found_markers$ of pairs $(marker_idx, marker)$, where $marker_idx$ is the BWT position where a variant marker with symbol $marker$ is found; $sa_int_start \leq marker_idx < sa_int_end$ and $bwt[marker_idx] = marker$

$markers_before_start = rank_{bitmask}(sa_int_start - 1)$
$offset = 1$
$marker_idx = select_{bitmask}(markers_before_start + offset)$
while $marker_idx < sa_int_end$ **do**
    $found_markers \leftarrow found_markers \cup \{(marker_idx, bwt[marker_idx])\}$
    $offset++$
    $marker_idx = select_{bitmask}(markers_before_start + offset)$

For the whole-genome PRG of *Plasmodium falciparum* (described in Section 4.3.1.1), the BWT bit mask and its *rank* and select support structures require an additional 4MB. We modified the backward search function in `gramtools` to do a range search over this bit mask (as described in Algorithm 2) instead of using the `SDSL` *range_search_2d* function over the wavelet tree and we performed the same experiment described in Section 4.3.2 (on Machine A). Resulting runtimes are shown in Table 4.5, together with baseline runtimes from Section 4.3.2 for comparison. Note that the observed differences are solely due to the change in the range search func-

Table 4.5: Runtimes for quasimapping 10,000 simulated reads of length 150 bp to the *Plasmodium falciparum* PRG with the version of `gramtools` code that performs the range search on a bit mask over BWT and with the baseline version that performs the range search over the wavelet tree instead. The one-time precalculation step consisted of computing the SA-intervals for 9-mers overlapping variant sites in the PRG using 7 threads.

`gramtools` version	Qmap time (sec)	Qmap speed (reads/sec)	Precalc time (min)
with bit mask range search	330	30.3	201
with WT range search (baseline)	486	20.5	215

tion – the *rank* storage optimisation described in the previous section is not included on this code branch (commit 768a785). We will report runtimes for the 2 merged optimisations in Section 4.5.

4.4.3 *k*-mer indexing

In this section, we investigate the performance impact of the choice of parameter k indicating the k-mer size for which suffix array intervals should be precomputed and stored before read quasimapping. As mentioned in Section 3.7.2, calculating the SA-intervals for the first symbols matched accounts for a significant proportion of the read quasimapping time. This is due to small k-mers matching many regions of the PRG and generating an additional SA-interval for each variant they overlap. Although expensive, precomputing the SA-intervals for all k-mers in the PRG (or only k-mers overlapping variants) needs to be done only once if its results are stored. However, its exact cost and quasimapping benefit depend on the value of k.

As k gets larger, the total number of k-mers increases, so we expect the precalculation time to increase and the data structure containing the suffix array intervals to become larger, requiring more memory. On the other hand, with each backward search iteration, the number of matching hits in the linear PRG should narrow down and the number of suffix array intervals should decrease. This means that with larger precalculated k-mers , quasimapping starts from less suffix array intervals and is con-

sequently faster. We seek to find the optimal value of k that reconciles these trade-offs. Note that the experiments in this section have been run with the baseline version of `gramtools` (from commit a510cbd), that does not include the two optimisations previously discussed (ranks storage and range search over BWT bit mask). We will report performance measures for the code version with the two merged optimisations and the optimal k parameter in Section 4.5.

4.4.3.1 *Plasmodium falciparum* genome

We performed the same experiment as in Section 4.3.2, quasimapping 10,000 simulated reads to the *Plasmodium falciparum* linear PRG, with varying size for the precalculated k-mers between $2 - 20$. Runtimes for the precalculation step and the quasimapping step are given in Table 4.6. As expected, the overall precalculation time increases with k-mer size and number of k-mers , whereas quasimapping time decreases. In the precalculation step, multiple instances of `gramtools` were run in parallel for different subsets of k-mers . Because of the computation length, this was done on the European Bioinformatics Institute computer cluster (Machine C), with variable numbers of cores being assigned to each job (third column in Table 4.6). At the end, results for each k-mer subset were aggregated and the read quasimapping step was run single-threaded on Machine B.

It is worth noting that for this experiment, we precalculated SA-intervals for both k-mers overlapping variants and k-mers neighbouring variants, i.e. located within 150 bp (read length) of a variant site. This ensured that all reads contributing to allele coverage counts were found in the PRG, across all k's. For $k = 2 - 8$, this set saturates all k-mers in the PRG. For $k = 9$, although this set is larger than the set of k-mers overlapping variants used in other sections of this thesis, it does not impact those results since most reads were quasimapped anyway. To generate this new set of k-mers , we ran the `gramtools` version from commit 9d41880, which includes an

Table 4.6: Impact of the k-mer size used for SA-intervals precalculation on quasimapping 10,000 simulated reads of length 150 bp to the *Plasmodium falciparum* PRG. In the first column, number of k-mers overlapping and neighbouring (within a read length) variant sites in the PRG. In the second column, elapsed runtimes for precalculating the SA-intervals for k-mers from first column, after running on a computer cluster (Machine C). This calculation was parallelized across a variable number of cores, specified in the third column. In the last column, read quasimapping times after the k-mer precalculation, single-threaded on a single computer (Machine B). All times are reported in seconds.

	Number of k-mers	Precalc time	# cores	Qmapping time
k=2	16	1.48	16	4,747
k=3	64	4.8	25	2,612
k=4	256	10.8	25	1,611
k=5	1,024	32.8	25	1,116
k=6	4,060	120	25	838
k=7	14,738	441	25	670
k=8	43,545	1,391.2	25	558
k=9	214,579	415.1	430	477
k=10	594,921	1,085.7	496	410
k=11	1,297,697	2,496.8	500	354
k=12	2,300,645	4,679.3	495	301
k=13	3,472,011	7,376.5	497	259
k=14	4,639,825	10,155.8	494	219
k=15	5,668,328	13,999.4	498	185
k=16	6,486,590	15,470.5	499	154
k=17	7,095,346	18,744.6	497	127
k=18	7,533,296	18,623.6	499	104
k=19	7,847,918	19,818	497	84
k=20	8,081,606	20,989.1	490	68

option to generate k-mers within a given distance from a variant site.

Next, we investigated how the number of SA-intervals generated at each k-mer size is responsible for the runtimes in Table 4.6. We quasimapped the 10,000 simulated reads again, but with no k-mer precalculation this time (setting $k = 1$ in the `gramtools` command). We modified `gramtools` to report the number of SA-intervals generated as each successive symbol in the read is matched. Figure 4.4 shows the median, minimum and maximum number of SA-intervals across the 10,000 reads at each backward search iteration. Strikingly, despite the fact that the median drops to

1 at $k = 13$, meaning that most 13-mers are crossing no more than 1 variant, extreme outliers are present far along the backward search iterations: over 1000 generated SA-intervals can be observed up to $k = 43$. Moreover, the number of SA-intervals peaks at 60,149 for $k = 14$ – this is extremely high for a PRG with an alphabet size of 88,581, indicating that there is a 14-mer that overlaps the majority of variant sites. This is consistent with the well-known fact that *Plasmodium falciparum* has a very repetitive genome, with expansions and contractions of short tandem repeats being a main source of variation [Gardner et al., 2002], [Tan et al., 2010], [Miles et al., 2016]. Consequently, `gramtools` spends a disproportionate amount of time on a reduced number of repetitive k-mers that overlap many distinct variant sites, generating many SA-intervals that need to be extended. To control for genome content, in the next section we will run a similar experiment using the human genome as a base for the PRG, which is much less repetitive.

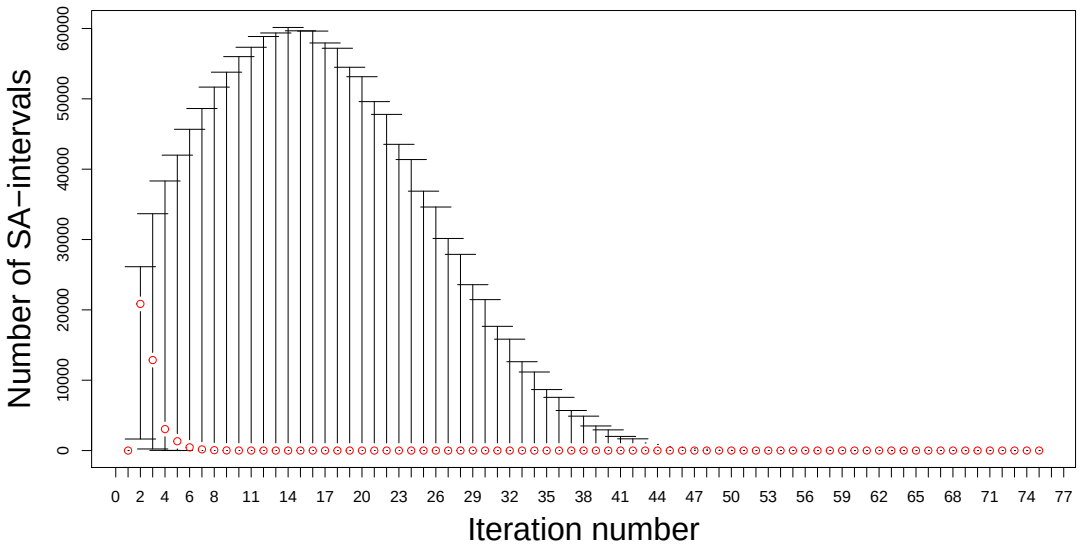


Figure 4.4: Median (in red) and range of the number of suffix array intervals generated per backward search iteration (or per character matched) during quasimapping simulated reads to the *Plasmodium falciparum* PRG (across 10,000 reads of 150-bp length). Only the first 75 iterations are shown.

4.4.3.2 Controlling for genome content: *P.falciparum*-sized PRG on human genome sequence

In this section, we study the effect of k-mer size on the number of SA-intervals generated during backward search with an experiment based on the human genome. We built a linear PRG for a 23 MB region (same size as the *Plasmodium falciparum* genome) of chromosome 21 of the human genome. This region represents nearly half of the chromosome and contains 44,490 variants with allele frequency above 0.23 (*Plasmodium falciparum* PRG has 44,439). The resulting linear PRG contains 24,193,756 symbols over an alphabet of size 87,604. The graph construction step took 5 seconds to complete and generating all 9-mers overlapping variant sites (211,694) in the graph took an additional 4.8 seconds. We will sometimes refer to this PRG as "human control PRG".

As before, we simulated 10,000 150-bp reads from a path in the PRG and quasimapped them, reporting the number of SA-intervals generated after each character match. A summary in the form of median, minimum and maximum number of SA-intervals generated at each iteration is in Fig. 4.5. Similar to the *Plasmodium falciparum* PRG, the median drops to 1 after 11 iterations, but the ranges are much less extreme. The maximum number of generated SA-intervals, 17,580, is observed after the second iteration, but decreases rapidly in the following iterations, reaching a value under 1000 at $k = 17$. As the backward search has less SA-intervals to process during each iteration, the speed of quasimapping is considerably higher in this experiment, 1,250 reads/sec compared to 20.5 reads/sec in the previous *P. falciparum* experiment (Table 4.7). There is a drawback though: a less repetitive genome induces a higher k-mer precalculation time due to the increased number of k-mers (Table 4.7). However, this is a one-off cost.

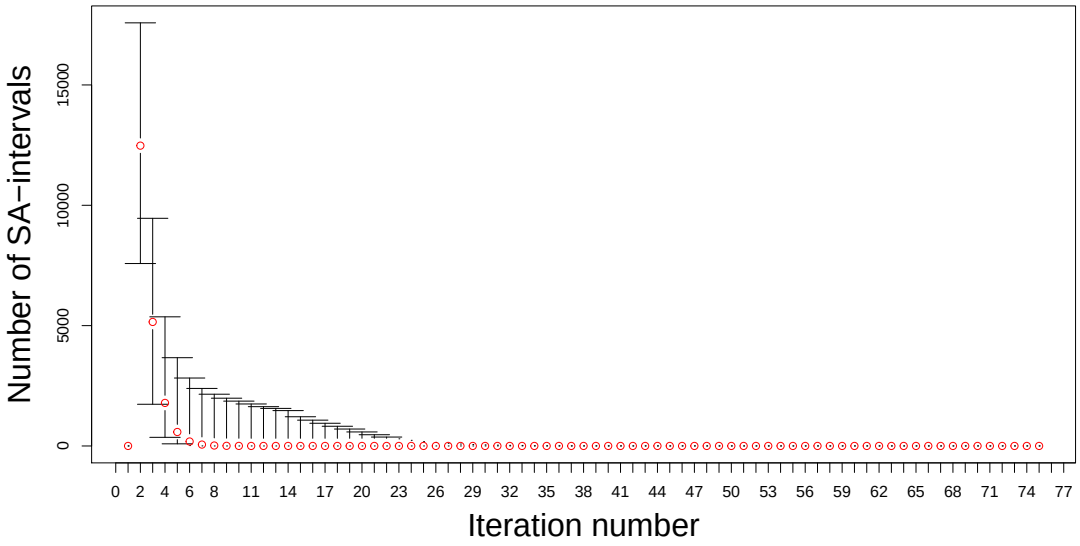


Figure 4.5: Median (in red) and range of the number of suffix array intervals generated per backward search iteration (or per character matched) during quasimapping simulated reads to the human control PRG (across 10,000 reads of 150-bp length). Only the first 75 iterations are shown.

Table 4.7: Runtimes for quasimapping 10,000 simulated reads of length 150 bp to the human control PRG compared with runtimes for the same experiment on *Plasmodium falciparum* PRG from section 4.3.2. The one-time precalculation step consisted of computing the SA-intervals for 9-mers overlapping variant sites in the PRGs using 7 threads.

Species	Qmapping time (sec)	Qmapping speed (reads/sec)	Precalc time (min)	# of k-mers (k=9)
human control	8	1,250	387	211,694
P. falciparum	486	20.5	215	99,333

4.5 Final performance measures

We made final benchmarks quasimapping 10,000 reads with a `gramtools` branch containing both the *rank* and range search optimisations, choosing an optimal parameter k informed by the results in Section 4.4.3. The quasimapping experiments in this section were run single-threaded with the `gramtools` version at commit c2a327a, on Machine B.

4.5.1 Whole-genome *Plasmodium falciparum*

In the case of the *Plasmodium falciparum* PRG, we chose $k = 17$ for precalculated k-mers , since the maximum number of SA-intervals it generates is past the peak in Fig. 4.4. Quasimapping runtimes are shown in Table 4.8, together with baseline runtimes from Section 4.3.2 for comparison. Note that we had to rerun the experiment described in Section 4.3.2 on the current machine (B) for consistency, hence the slightly different figures from the ones in the top row of Table 4.2.

Table 4.8: Runtimes for quasimapping 10,000 simulated reads of length 150 bp to the *Plasmodium falciparum* PRG, with `gramtools` merged optimisations version and $k = 17$ and `gramtools` baseline version and $k = 9$ respectively.

	Time (sec)	Speed (reads/sec)
`gramtools` optim + k17	27	370
`gramtools` baseline + k9	472	21

4.5.2 Human control genome

In the case of the human control PRG, we chose $k = 11$ for precalculated k-mers , since it is the smallest k for which the median number of SA-intervals generated drops to 1 (Fig. 4.5). However, 11-mers overlapping variants couldn't be generated because of a software bug, so all 11-mers in the PRG were used instead for precalculation (3,224,759 11-mers in total). The precalculation step was completed in 1.5 hours on

a computer cluster (Machine C), across 500 cores.

Indexing all k-mers in the PRG is a conservative approach that induces marginally overestimated quasimapping times due to unnecessarily quasimapping reads that do not overlap variants and do not contribute to inference. Quasimapping runtimes are shown in Table 4.9, together with baseline runtimes from Section 4.4.3.2 for comparison. Note that we had to rerun the experiment described in Section 4.4.3.2 on the current machine for consistency (B), hence the slightly different figures from the ones in the top row of Table 4.7.

Table 4.9: Runtimes for quasimapping 10,000 simulated reads of length 150 bp to the human control PRG, with `gramtools` merged optimisations version and $k = 11$ and `gramtools` baseline version and $k = 9$ respectively.

	Time (sec)	Speed (reads/sec)
`gramtools` optim + k11	0.91	>10,000
`gramtools` baseline + k9	7.6	1,315

4.6 Impact of graph structure on performance

In this section, we investigate how the choice of parameters controlling the reference graph structure affects the `gramtools` performance. As discussed in the previous chapter (Section 3.6), the same set of variants can be encoded by different graphs, by either assigning them to separate variation sites (Fig. 4.6b), or collating them into a single larger graph site that enumerates all their haplotypic combinations (Fig. 4.6c). In theory, the former option would induce slower quasimapping times due to a larger alphabet and more variant markers that need to be crossed while aligning the reads. However, the latter option is only viable if the variants are close enough so that the length of the resulting haplotypes does not increase the length of the linear PRG too much. There is also a limit on the number of variants that can be collated into a single graph site, because of the combinatorial explosion of the number of possible haplotypes, which can also increase the length of the linear PRG too much. This poses

the question whether augmenting the linear PRG with long haplotypes increases its length enough to cancel out the speed-up produced by the reduced alphabet size. We present some experiments designed to tackle this question.

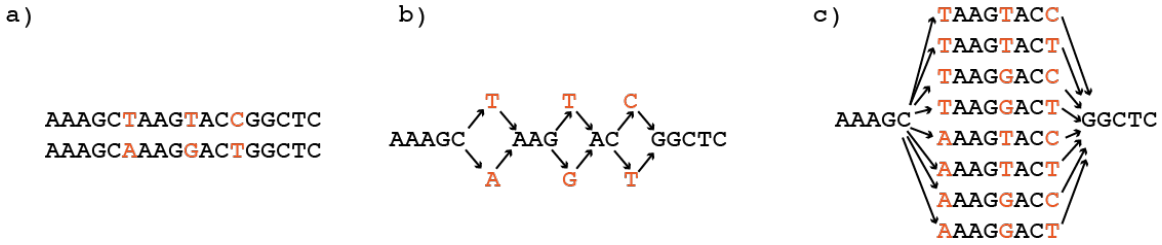


Figure 4.6: PRG graph structure options. The sequences shown in **a)** could be represented either as 3 separate mutations, shown in **b)**, or enumerated as 8 small haplotypes, shown in **c)**. `gramtools` supports both options.

We used as a primary reference sequence the same 24 MB region of chromosome 21 of the human genome that was used in Section 4.4.3.2. We made this choice in order to avoid confounding from the unusual repetitiveness of the *Plasmodium falciparum* genome. We simulated SNPs on the primary reference in the following way: after every 5000 bp, we took a block of size 120bp and simulated a SNP every 10bp within that block. This resulted in 4,695 blocks, with 12 evenly spaced SNPs per block, totalling 56,340 SNPs. We built 5 different PRGs encoding this set of variants, with the following structures:

- 1 SNP per graph site; this resulted in 12 graph sites per block, each with 2 alternative sequences of length 1
- 2 SNPs per graph site; this resulted in 6 graph sites per block, each with 4 alternative sequences of length 11
- 3 SNPs per graph site; this resulted in 4 graph sites per block, each with 8 alternative sequences of length 21
- 4 SNPs per graph site; this resulted in 3 graph sites per block, each with 16 alternative sequences of length 31

Table 4.10: Length and alphabet size of 5 equivalent PRGs encoding the human control genome and 56,340 simulated SNPs that occur in clusters of 12, separated by 9 bases. Each PRG has a different structure, from the simplest that represents 1 SNP per graph site (1 SNP collated), to the most complex that represents 6 SNPs per graph site (6 SNPs collated) as long haplotypes.

PRG type	Length of linear PRG	Alphabet size
1 SNP collated	24,225,361	112,684
2 SNPs collated	25,070,461	56,344
3 SNPs collated	26,929,681	37,564
4 SNPs collated	30,788,971	28,174
6 SNPs collated	54,780,421	18,784

- 6 SNPs per graph site; this resulted in 2 graph sites per block, each with 64 alternative sequences of length 51

Table 4.10 summarizes these graph structures, showing the total length and alphabet size for each linear PRG. Even with the relatively small distance between SNPs that we have chosen (9 bp), collating more than 3 SNPs into a single graph variation site results in a considerable increase in the size of the reference data structure, because of the number of additional haplotypes. In the extreme case of collating 6 SNPs, the size of the resulting linear PRG is more than double the size of the PRG representing the SNPs in separate variation sites.

Next, we simulated 155,000 reads of length 150 bp (equivalent of 1x depth of coverage) from an arbitrary graph path and quasimapped them to each of the 5 linear PRGs, recording runtimes. We used $k = 11$ for the size of precalculated k-mers , resulting in 930,730 k-mers overlapping variant sites. Table 4.11 shows precalculation times for these k-mers and subsequent read quasimapping times, obtained with `gramtools` version at commit 10ce376 (baseline, no optimisations), on Machine B. As expected, read quasimapping becomes faster as the number of collated SNPs increases, due to the reduction in alphabet size. The biggest increase is observed when going from no collation to combining every 2 adjacent SNPs, when both the quasimapping time and k-mer precalculation time decrease by $\sim$30%. Collating more than 2

Table 4.11: Runtimes for quasimapping 155,000 reads of length 150 bp to 5 equivalent, but distinctly structured PRGs encoding the human control genome and 56,340 simulated SNPs that occur in clusters of 12. The one-time precalculation step consisted of computing the SA-intervals for all 11-mers overlapping variant sites in the PRG using 24 threads.

PRG type	Qmapping time (sec)	Qmapping speed (reads/sec)	Precalc time (hours)
1 SNP collated	1385	112	10.06
2 SNPs collated	1022	151	7.07
3 SNPs collated	973	159	7.35
4 SNPs collated	905	171	8.76
6 SNPs collated	880	176	19.95

SNPs continues to decrease quasimapping time, but gradually increases the k-mer precalculation time. This time keeps increasing to the point where it surpasses the initial time to precalculate k-mers for the PRG representing each SNP as a separate graph site – nearly 20 hours for the PRG that collates 6 SNPs versus 10 hours for the initial PRG. This confirms our initial assumption that there is a limit to the number of SNPs that can be collated before the additional length of the linear PRG overrides any performance gains implied by the reduced alphabet size. In this case, the optimal choice is collating 4 SNPs, as it implies the fastest mapping speed without an additional time burden in the k-mer precalculation step.

In a real data scenario, it is much more difficult to define a hard rule about combining nearby variants because the distances between them will vary and because regions of high variant density won't be distributed evenly across the genome. Therefore, it may be advantageous to collate more than 4 variants if they are closer than 10 bp or if there are overall less regions with dense variants than in our experiment. In this respect, our simulation is somewhat extreme as all SNPs occur in clusters, whereas real variants tend to be more dispersed across the genome (we note that this is also the reason why we see much lower quasimapping speeds in this section than in Section 4.5.2, despite having a comparable number of variants encoded in the PRG). Moreover, some variant-dense regions will be larger than others, so the optimal num-

ber of variants to collate might vary from region to region. More experiments based on real data are needed to clarify these trade-offs, though we sketch the following approach: identify the blocks of the same length as the reads, that contain multiple variants; calculate an upper bound for the length of the linear PRG as the length of the PRG obtained by collating all variants within each block; choose a different number of variants to collate for each block, informed by the distances between variants and the upper bound. An alternative method, depending on the availability of haplotype information, would be to collate a large number of variants, but only include haplotypes that had been observed before.

4.7 Scaling

The main reason our variation-aware backward search is slower than the normal backward search in an FM-index is that with the search of each successive symbol in the read, multiple SA-intervals need to be extended with the respective symbol. Let us focus first on the operations required to extend a single SA-interval of size 1, i.e. representing a unique match in the PRG of the substring that has already been searched. Extending such SA-interval with a DNA character is a constant time operation[5], as in the standard backward search. If a non-DNA character is encountered, an additional constant time operation needs to be performed in order to skip it and then extend with a DNA character. Let the read length be $|R|$ and the length of its smallest substring with a unique match be k. Thus, the number of operations required to match the rest of the read starting from this unique substring is at most $|R| - k + 2v$, where v is the number of variant sites crossed during the search. Then, if we let v_{max} be the maximum number of variants within a window of length $|R|$ across all genomic windows of this length, the computational complexity of this procedure is $O(|R| - k + 2v_{max})$.

[5]provided ranks have been stored

Now, suppose SA-intervals have been precalculated for k-mers at some k and the goal is to extend them to the end of the reads and measure the number of operations required by read quasimapping. The number of SA-intervals a k-mer generates is a function of the number of its "horizontal repeats"[6] in the PRG and the number of variants overlapped by these repeats. "Horizontal" repeats that do not cross any non-DNA characters are grouped together in a single SA-interval. During further character extensions, for every one of these "horizontal repeats" that crosses a non-DNA character, the original SA-interval splits and an additional SA-interval is created. Whenever "vertical repeats" are encountered, they are grouped together in a single SA-interval, which cannot split during subsequent extensions.

To sum up, for any indexed k-mer , among the precalculated SA-intervals, there is one SA-interval representing matches (either "horizontal" or "vertical") that haven't crossed any non-DNA characters; each of the remaining SA-intervals represents a "horizontal match" that crosses at least one non-DNA character. These remaining SA-intervals are either of size 1, or if not, they represent "vertical repeats" and cannot split further, so they remain "horizontally unique" – hence, extending each of them to the end of the read has complexity $O(|R| - k + 2v_{max})$. Now, the original SA-interval representing matches that haven't crossed any non-DNA characters can split into multiple SA-intervals during subsequent character extensions. Let this SA-interval be $[l, r]$. In the worst case scenario, each of the $r - l + 1$ matches it represents is neighbouring[7] a variant and will create an additional SA-interval in a subsequent character matching iteration. In that case, the total number of SA-intervals that need to be processed to extend such k-mer to the end of the read is equal to the number of "horizontal matches" the k-mer has in the PRG. Therefore, if we let h_{max} to be the maximum number of "horizontal matches" across all indexed[8] k-

[6] refer back to Section 3.8

[7] within a distance equal to $|R| - k$ from a non-DNA character

[8] across all k-mers at size k used for precalculation

mers , the computational complexity of read quasimapping after k-mer indexing is $O(h_{max}(|R| - k + 2v_{max})$ in the worst case scenario.

It is worth noting that most of the time this upper bound would not be reached, due to many k-mers being repeated in non-variable regions of the genome. In those cases, the actual maximum number of SA-interval extensions performed during read quasimapping is h_var_{max}, the maximum number of "horizontal matches" that are neighbouring[9] a variant across all k-mers in the PRG[10], which can be much smaller than h_{max}. In Figs. 4.4 and 4.5, h_var_{max} corresponds to the peaks at $k = 14$ and $k = 2$, respectively, if no k-mer precalculation was done.

In practice, h_{max} and v_{max} depend on the size of the genome, the number of variants in the PRG and the spatial distribution of these variants in the genome, although the specific dependency relationship will vary between species with different genome contents. However, certain PRG structures can be used to limit the impact of these variables. As seen in the previous section, v_{max} can be controlled by collating nearby variants into single sites. Choices made during graph construction might also control, to some extent, for the number of variant sites that cross repetitive regions, such that the upper bound for number of SA-intervals generated, h_{max}, is rarely reached.

4.8 Conclusions

In this chapter, we evaluated the computational requirements of `gramtools` for the *Plasmodium falciparum* and human genomes. We showed that our population reference graph implementation scales to the whole human genome, requiring 27 GB to index the reference genome and 80 million variants, a lower memory usage than `GCSA2` and `BWBBLE`. However, the read quasimapping step of `gramtools` is readily usable only

[9] within a distance equal to $|R| - k$ from a non-DNA character
[10] at any k that is larger than the one used for precalculation

for megabase-sized genomes at the moment. The optimised version quasimaps at a rate of 370 reads/second against a whole-genome PRG for *Plasmodium falciparum*. At this rate, a sample with 50x depth of coverage, would be processed in 5.7 hours on a single core.

It is yet unclear whether `gramtools` quasimapping will be viable for the whole human genome. The main bottleneck is the k-mer precalculation step and memory management issues that prevent its split over a very large number of cores. Past these issues, it is hard to estimate how `gramtools` will scale based on the optimisations presented in this chapter and the results on the human control PRG, since it does not depend only on genome size and alphabet size, but also on the genetic architecture and the repeat content. This will be discussed further below.

There are a number of improvements that could be made in the future to enhance the performance of `gramtools`. Currently, the k-mer precalculation step is the most inefficient, so optimisation efforts should be directed there first. A main shortcoming is that k-mers are processed independently, when there is index information that can be reused due to the sequence relationships between k-mers . Substring sharing between k-mers can be exploited in the following way: if SA-intervals for all *k-1*-mers are stored, the SA-interval for any k-mer can be obtained by simply extending the SA-interval of its *k-1*-suffix with its first character; the process can continue through backward induction to obtain *k-1*-mers from *k-2*-mers; and so on until *1*-mers A,C,G,T, which have a single SA-interval each. In practice, k-mers should be processed together, in sorted order, and once the SA-intervals for a suffix have been calculated, that information should be stored and never be recalculated for any of the k-mers sharing the respective suffix. We expect this to provide a significant boost to the speed of k-mer precalculation.

Another way to improve the k-mer precalculation step is to change the data structure encoding the hash table of k-mers and their suffix array intervals. Currently,

`gramtools` uses the `Boost` *unordered_map*, which performs poorly on data sets of the order of tens of millions of elements (the human PRG with common variants has 65 million 13-mers). Many other more efficient hash function implementations are available [Limasset et al., 2017].

One drawback of the variation-aware backward search implementation is that whenever an odd variant marker is encountered, the suffix array is queried to determine whether the marker represents the start or end of the variant site (line 14 in Algorithm 1). This is expensive if the position queried was not sampled during the suffix array construction, since it needs to be inferred from the nearest sampled position. Currently, `gramtools` uses the default `SDSL` sampling strategy, but `SDSL` offers the option to specify the sampling strategy, so we could ensure that all positions representing odd variant markers are sampled in the suffix array. This would speed up quasimapping, but also the k-mer precalculation step, since variant markers are crossed most often in the first few backward search iterations.

In any reference graph approach, there is an inherent coupling between graph structure and read mapping. We have shown on simulated data that the `gramtools` quasimapping performance depends on how it is chosen to represent variants as alternative sequences in the graph. This suggests that on real data, efforts to optimise the graph structure will need to be tailored to the species analysed and based on the variant distribution across the genome. More experiments are needed to understand how genetic architecture impacts `gramtools` scaling to different species.

The repeat content of the genome in question also affects quasimapping efficiency, as it was proven by our comparisons between the PRG of the unusually repetitive *Plasmodium falciparum* and the human control PRG. It is likely that quasimapping to the *Plasmodium falciparum* PRG is slowed down by outlier k-mers that are extremely repetitive and overlap many variants, inducing many candidate hits for reads ending in such k-mers . One way to mitigate this is finding the least repetitive k-mer in the

read and starting the pattern match from there, with backward search from the start of the k-mer and forward search from its end. This assumes the bidirectional search functionality, which can be naturally supported by `gramtools`, as discussed at the end of the previous chapter.

Chapter 5

Malarial surface antigens application: *DBLMSP* and *DBLMSP2*

5.1 Introduction

Malaria is the deadliest parasitic disease affecting humans. It is caused by infection with parasites from the *Plasmodium* genus, with *Plasmodium falciparum* being the most virulent species, responsible for the majority of lethal cases [WHO, 2016]. *Plasmodium falciparum* has a complex life cycle, reproducing asexually in the human host and sexually in the mosquito vector. It is transmitted by female *Anopheles* mosquitoes during a blood meal, when sporozoites[1] are injected into the human skin. These travel to the liver and invade hepatocytes. After development in the liver, merozoites[2] are released into the bloodstream, where they undergo repeated cycles of erythrocyte[3] invasion, asexual multiplication inside the RBCs, release, and then

[1] from Greek, *sporos*=seeds
[2] blood stage forms of the parasite
[3] red blood cells or RBCs

reinvasion. At this stage, clinical symptoms of malaria (fever, chills, etc.) are manifested, often induced by an excessive inflammatory response. Some merozoites exit the clonal expansion cycle and differentiate into gametocytes[4], which can be ingested by a mosquito during another blood meal. Sexual division occurs in the mosquito's gut and haploid sporozoites are formed, which then travel to the sallivary glands. From there, they can be transmitted to another human during the next blood meal, continuing the life cycle. The information in this paragraph was taken from [Cox, 2010], [Cowman et al., 2012].

Despite decades of research efforts, a long-lasting vaccine with broad efficiency has not been found [Schwartz et al., 2012]. The only licensed vaccine is RTS,S/AS01, which targets the pre-erythrocytic stage. It has had moderate efficacy against clinical malaria (36% in children after 4 years), but its effect diminishes over time [RTS, 2015]. The theoretical viability of a vaccine is supported by the fact that adults living in endemic areas present less infection episodes and lower densities of parasitemia, indicating naturally acquired immunity [Doolan et al., 2009]. However, this immune response is inefficient and incomplete, since it first protects against severe symptoms during childhood, necessitating repeated exposure to acquire the ability to control parasite growth [Barry et al., 2011]. This slow acquisition of partial protection suggests an allele-specific immune response. Indeed, two advanced blood stage vaccines, FMP2.1/AS01 and 'Combination B', only had efficacy to infections with parasites carrying the same allele as the vaccine antigen, *AMA1* and *MSP2*, respectively [Thera et al., 2011], [Ouattara et al., 2013], [Genton et al., 2002].

Many proteins displayed on the merozoite surface have high levels of diversity in the underlying gene. Since these proteins are briefly exposed to host antibodies in between erythrocyte invasions, this diversity is likely part of a mechanism for immune evasion [Gomes et al., 2016]. Antigenic variation has been proposed as an

[4]precursors of male and felame gametes

explanation for the slow development of adaptive immunity and the limited efficacy of current vaccines [Takala and Plowe, 2009], [Barry et al., 2009], [Thera and Plowe, 2012], [Stanisic et al., 2013]. Leading vaccine candidates have extensive sequence diversity [Barry et al., 2009], [Soulama et al., 2011], but vaccines have often been developed on a single allele, increasing the risk of immune selection pressure on the non-vaccine-like allele [Genton et al., 2002], [Takala et al., 2009]. Moreover, decisions about which allele to include in the formulation have often been made based on which strain was available in the laboratory, resulting in vaccines based on a low-prevalence antigenic variant that is probably not representative of target parasite populations (Fig. 5.1 from[5] [Barry et al., 2009]) [Takala et al., 2009].

GENE	VACCINE*	HAPLOTYPE	PREVALENCE
csp	3D7	1	0.01
trap	T9/96	59	0.01
lsa1	3D7	1	0.23
ama1	3D7, FVO	2, 3	0.06, 0.06
eba175	FVO	2	0.13
msp1	3D7, FVO, FUP	1, 2, 3	0.16, 0.28, 0.26
msp3	FC27	6	0.10
msp4	3D7	1	0.13
glurp	FVO	8	0.06
pfs48/45	NF54	11	0.09

*Names of laboratory isolates used for vaccine development.
doi:10.1371/journal.pone.0008497.t004

Figure 5.1: Worldwide prevalence of *P. falciparum* antigen haplotypes that are components of malaria vaccines. Source: [Barry et al., 2009]

In order to tackle these issues, there has been increasing demand for comprehensive population genetic studies that measure the extent of antigenic diversity and the frequency of different haplotypes in multiple geographic locations [Ouattara et al., 2015], [Barry et al., 2009], [Barry and Arnott, 2014]. Such studies require large sample sizes in order to capture the extent of diversity, but the recent advances in high throughput genotyping and sequencing technologies have made them feasible. Their results, together with immuno-epidemiological studies that assess the correlation between antibody response and malaria incidence [Fowkes et al., 2010], are expected to

[5]article distributed under the CC BY license, which allows reuse with appropriate citation

inform the design of a multivalent vaccine that generates a broad antibody response and protects against a wide range of parasite strains [Osier et al., 2014], [Barry and Arnott, 2014].

Indeed, initial small studies (< 100 samples) on genomic sequencing of *P. falciparum* have revealed many signatures of balancing selection, mostly in genes expressed at the merozoite stage of the parasite, with divergent alleles being preserved within populations at low to medium frequencies [Amambua-Ngwa et al., 2012], [Ochola et al., 2010], [Tetteh et al., 2009]. This set of genes are now prioritised for functional study and vaccine candidacy assessment. Some of the highest balancing selection signals were found in the *DBLMSP* and *DBLMSP2* genes (Fig. 5.2 from[6] [Amambua-Ngwa et al., 2012]), which will be the focus of this chapter.

However, describing the genetic architecture of regions with a high degree of polymorphism presents analytical challenges with short-read genomic sequencing data. A previous study has shown that the complex genetic variation in several well-studied merozoite surface antigens (including *DBLMSP* and *DBLMSP2*) is inaccessible to standard variant calling techniques based on mapping sequencing reads to a single reference genome, 3D7 [Gardner et al., 2002], due to frequent alignment failure [Miles et al., 2016]. In this chapter, we demonstrate how to solve this problem by applying `gramtools` to an extended set of references. We will perform a population genetic study on the *DBLMSP* and *DBLMSP2* genes, and characterise their polymorphisms in detail using whole-genome sequencing data from 14 populations around the world.

5.2 *DBLMSP* and *DBLMSP2* allelic dimorphism

DBLMSP and *DBLMSP2* (also known as *MSP3.4* and *MSP3.8*) encode proteins secreted on the surface of the merozoite and emerged as the most polymorphic genes on chromosome 10 in an analysis of sequencing data from experimental genetic crosses

[6]article distributed under the CC BY license, which allows reuse with appropriate citation

Gene ID	Chr	Product description	N	% CDS	S	$\pi \times 10^{-3}$	dN/dS	Tajima's D	Fu & Li's D*	Fu & Li's F*	Peak Expression
PF10_0355	10	merozoite surface protein	1845	80	111	25.9	0.78	3.31	1.96	2.95	late schizogony
MAL7P1.229	7	cytoadherence linked asexual protein	4064	99	56	5.6	0.47	2.79	0.59	1.66	late schizogony
PFA0700c	1	*Plasmodium* exported protein (hyp10)	333	100	18	22.7	6.62	2.76	1.24	1.97	late ring
PF10_0051	10	ADP/ATP carrier protein, putative	995	97	14	5.8	0.32	2.60	0.58	1.49	late ring
PFL2555w	12	*Plasmodium* exported protein (PHISTa)	797	96	11	5.6	0.30	2.49	0.85	1.52	early ring
PFD0980w	4	holo-(acyl-carrier protein) synthase, putative	1785	100	15	3.3	0.44	2.27	1.11	1.78	merozoite
PF11_0014	11	PfMC-2TM_11.1, Maurer's cleft protein	621	89	30	18.3	1.07	2.16	0.79	1.49	late ring
PFA0065w	1	Pfmc-2TM, Maurer's cleft protein	565	100	21	13.6	0.38	2.04	1.70	2.04	late ring
PFC0110w	3	cytoadherence linked asexual protein 3.2	3749	100	144	13.8	0.11	2.02	1.42	1.92	early schizogony
PF07_0124	7	conserved, unknown function	2381	99	14	2.2	0.23	1.97	0.58	1.24	late schizogony
PF07_0042	7	conserved, unknown function	6282	100	103	5.5	0.52	1.97	1.41	1.91	gametocyte
PF11_0344	11	apical membrane antigen 1	1844	100	78	14.1	7.25	1.95	1.52	1.97	late schizogony
PFB0950w	2	conserved *Plasmodium falciparum* protein family	718	100	20	9.7	0.36	1.93	1.31	1.72	merozoite
PF08_0002	8	surface-associated interspersed gene 8.2	5786	94	181	10.6	1.26	1.85	1.55	1.99	gametocyte
PFB0080c	2	*Plasmodium* exported protein (PHISTb)	1133	100	20	6.1	0.92	1.84	0.92	1.40	early ring
PF10_0015	10	acyl-CoA binding protein, isoform 1	273	100	15	19.6	0.35	1.83	1.11	1.60	merozoite
PF07_0004	7	*Plasmodium* exported protein	1472	100	45	10.2	0.56	1.82	0.38	1.06	merozoite
PF07_0085	7	ferrodoxin reductase-like protein	1929	100	10	1.9	0.04	1.82	0.78	1.33	early trophozoite
PF10_0348	10	Duffy binding-like merozoite surface protein	1599	98	32	6.8	1.33	1.81	1.86	2.10	late schizogony
MAL8P1.32	8	nucleoside transporter 2	1758	100	12	2.4	1.39	1.70	−0.14	0.56	early trophozoite
PFA0180w	1	ATP-dependent RNA helicase, putative	3768	100	21	1.2	0.54	1.65	0.22	0.82	early trophozoite
PFE0560c	5	MORN repeat protein, putative	3849	90	20	1.8	0.46	1.63	0.95	1.40	late trophozoite
MAL13P1.105	13	ser/thr protein phosphatise 2A subunit	2512	100	10	1.4	0.03	1.63	0.11	0.68	late trophozoite
PFB0935w	2	cytoadherence-linked asexual protein 2	4264	100	58	4.4	0.16	1.61	1.09	1.49	late schizogony
PFA0665w	1	DBL containing protein, unknown function	6991	80	194	9.5	0.46	1.59	1.35	1.73	null

N, number of aligned nucleotide positions analysed; % CDS, percentage of the complete gene coding sequence analysed; S, number of polymorphic sites analysed per gene; π, pairwise nucleotide diversity index; peak expression, as determined by previous microarray transcriptome analyses; dN/dS, Nei & Gojobori ratio of pairwise nucleotide diversity at nonsynonymous sites compared with synonymous sites. Results for all 2853 genes with 3 or more SNPs are given in Table S2.
doi:10.1371/journal.pgen.1002992.t001

Figure 5.2: 25 genes with $\geq$10 SNPs showing highest values of Tajimas D index in a genome-wide analysis of sequences of 65 Gambian *P. falciparum* isolates. Top balancing selection signal is in PF10_0355, an alternative name for the *DBLMSP2* gene. PF10_0348 is gene *DBLMSP*. Source: [Amambua-Ngwa et al., 2012]

(Fig. 5.3 from[7] [Miles et al., 2016]). They belong to the *msp3* gene family [Rice et al., 2014], consisting of 8 paralogous genes tandemly arrayed across 32 kb, which have been proposed as candidates for a multicomponent vaccine [Daher et al., 2010]. Unlike other members of the *msp3* family, these two genes contain a Duffy binding-like (DBL) domain [Singh et al., 2009], [Wickramarachchi et al., 2009], which is known to mediate interactions between the parasite and receptors on the erythrocyte surface [Hodder et al., 2012]. DBL domains are also present in the *EBA175* and *EBA140* ligands [Maier et al., 2003], which facilitate erythrocyte invasion, and in members of the hyper-variable *var* gene family [Kraemer and Smith, 2006], which are exposed on the surface of the infected erythrocyte. However, the function of the DBL domain in *DBLMSP* and *DBLMSP2* is not completely characterised. It has been shown *in vitro* that the products of these two genes bind directly to IgM via their DBL domains [Crosnier et al., 2016]. It has been suggested that this binding masks the parasite from the host adaptive immune system since, in its presence, immunoreactivity to IgG from malaria-exposed adults is reduced [Crosnier et al., 2016].

We performed an initial analysis of genetic variation of the 2 genes using Illumina short-read sequencing data from 434 samples from The Gambia, Ghana, Guinea and Cambodia (Fig. 5.4 from[8] [Crosnier et al., 2016]). We generated variant calls with the `Cortex` *de novo* assembler, which builds a de Bruijn graph[9] of the sequencing reads in the sample, without using a reference genome. `Cortex` has high specificity and it can access complex variation, but with limited sensitivity. Phylogenetic trees (Fig. 5.4 (a)) of a few representative haplotypes confirmed a known allelic dimorphism within the DBL domains [Amambua-Ngwa et al., 2012], [Tetteh et al., 2009], [Ochola et al., 2010]. Haplotypes cluster into two major allelic classes that are deeply diverged (∼65% nucleotide similarity), where the diversity between classes is orders

[7] article distributed under the CC BY license, which allows reuse with appropriate citation
[8] copyright held by the author
[9] refer back to Section 2.4.1

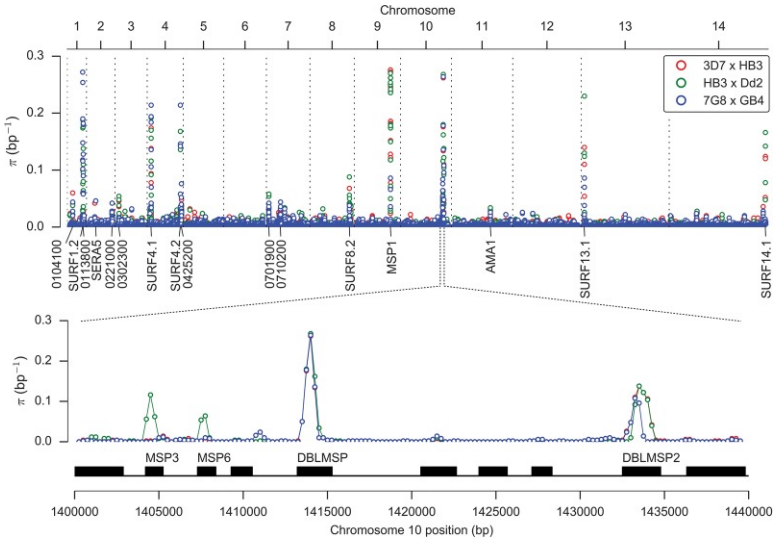


Figure 5.3: Variation in nucleotide diversity over the core genome of *P. falciparum*. The peak of nucleotide diversity on Chromosome 10 is expanded to show four distinct peaks due to genes encoding merozoite surface antigens *MSP3*, *MSP6*, *DBLMSP*, and *DBLMSP2*. All labeled loci (with the exception of *AMA1*) are sites of complex variation where read mapping often fails due to highly diverged alleles from the reference sequence. For these loci, *de novo* assembly of sequence reads is required to determine the nonreference alleles. Source: [Miles et al., 2016]

of magnitude greater than the diversity within classes. Figures 5.4 (b) and (c) show a schematic representation of an amino acid multiple sequence alignment (MSA) of a few representative haplotypes, with the same colour indicating identical sequence. The location of the dimorphism can be observed as in the middle of the DBL domain for *DBLMSP*, and at the beginning of the DBL domain for *DBLMSP2*. The MSAs suggest that recombination occurs in the two genes, but it is limited, or selected against, within the dimporhic region. We included in the MSA the *DBLMSP* sequence of *Plasmodium rechenowi*, the outgroup species of *Plasmodium falciparum*, which split several million years ago, to show that it is very close to the 3D7-like allelic class. This indicates that the dimorphism might be ancient and predate the split between *Plasmodium falciparum* and *Plasmodium reichenowi*, although more *Plasmodium reichenowi* samples need to be sequenced to confirm.

The phenomenon of allelic dimorphism has been observed in other genes encoding

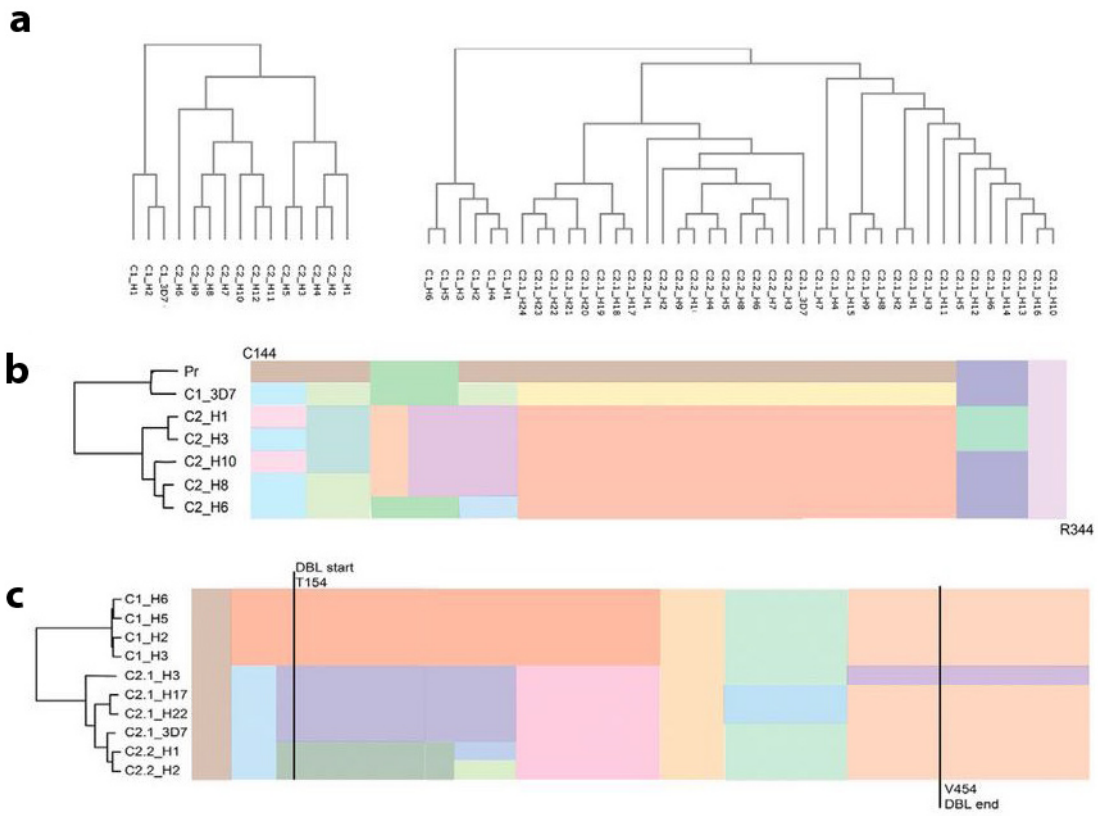


Figure 5.4: Genome sequencing of 434 *P. falciparum* isolates reveals *DBLMSP* and *DBLMSP2* are dimorphic and divergent. **(a)** Cluster dendrogram of the 15 *DBLMSP* haplotypes (left) and 41 *DBLMSP2* haplotypes (right) identified in the study. Both trees were obtained using `ClustalW` [Sievers et al., 2011]. **(b)** A schematic representation of the amino acid multiple sequence alignment of six representative exemplar haplotypes for the DBL domain of *DBLMSP* between residues Cys144 and Arg344, and the *P. reichenowi* ortholog. Note the long non-recombining region has two alternate allelic forms (orange and yellow), of which the yellow is relatively close to the *P. reichenowi* sequence (shown in brown). A phylogenetic tree indicating the relationships of the sequences built by clustering haplotypes based on a distance matrix (where distance corresponds to the number of SNP and indel differences) is shown on the left. The alleles similar to 3D7 are labelled class 1 and the ones belonging to the dissimilar clade are labelled class 2. **(c)** A schematic representation of the amino acid multiple sequence alignment of 10 representative exemplar haplotypes for the DBL domain of *DBLMSP2* between residues Glu102 and Gly550. Note the long non-recombining region has two alternate allelic forms (orange for class 1 and purple/dark green for subclasses 2.1/2.2, respectively). A phylogenetic tree indicating the relationships of the sequences built by clustering haplotypes based on a distance matrix (where distance corresponds to the number of SNP and indel differences) is shown on the left. Source: [Crosnier et al., 2016]

merozoite surface antigens, such as *MSP1* [Miller et al., 1993], *MSP2* [Ferreira and Hartl, 2007], *MSP3* [McColl and Anders, 1997], *MSP6* [Pearce et al., 2004], suggesting it underlies a broader mechanism used for immune escape, possibly driven by a dimorphic feature of the host immune system. However, there has been no evidence yet that the dimorphic alleles differ in functional properties [Crosnier et al., 2016], [Binks and Conway, 1999]. Several hypotheses have been proposed for the origin of the dimorphism: adaptation to local host populations, introgression from a related species, paralogous genes losing one copy via deletion, population bottleneck from an ancient diverse *Plasmodium falciparum* containing multimorphic alleles and complex modes of balancing selection [Roy et al., 2008]. But population genetic data has yielded conflicting conclusions [Hartl et al., 2002], [Hartl, 2004], since the high levels of polymorphism in surface antigens do not appear consistent with the low levels of diversity in the rest of the genome [Sundararaman et al., 2016]. Moreover, while these hypotheses seemed plausible for particular instances of the dimorphism, each of them has failed to offer a general explanation for this pattern across unrelated loci [Roy et al., 2008]. Recent publication of a draft genome for *Plasmodium gaboni*, an outgroup of both *Plasmodium reichenowi* and *Plasmodium falciparum*, has allowed estimations for times of divergence between dimorphic alleles in several merozoite surface proteins. The estimated origins of dimorphism at these loci range from after the *Plasmodium reichenowi*/*Plasmodium falciparum* split (*MSP6*) to before the *Plasmodium gaboni*/*Plasmodium reichenowi* split (*MSP1*), suggesting recurrent emergence [Roy, 2015]. This illustrates the challenge in developing a general evolutionary model for allelic dimorphism, because the loci displaying this pattern may have had very different histories. However, past studies have had limited resolution because of small sample sizes and the scarcity of whole-genome data available, but also because of computational challenges in the analysis of such polymorphic loci.

Accessing variation in dimorphic regions is difficult with standard methods be-

cause they rely on a single reference sequence, so the power of variant discovery on the divergent haplotype is low. In order to circumvent this, we developed a reference graph resource that incorporates their known variants acquired from multiple sources of information, we quasimapped reads to this graph with `gramtools` and extracted a personalised reference per sample to serve as substrate for calling novel polymorphisms not present in the graph (Fig. 5.5(a)-(d)). The details of each step in this process are explained in the next sections. We apply this method to *DBLMSP* and *DBLMSP2* in order to get a full picture of the variation in the two genes.

5.3 PRG construction from Pf3k reference panel

The Pf3k (*Plasmodium falciparum* 3000) project is an international consortium aiming to provide a high-resolution view into the natural genome variation of *Plasmodium falciparum*. The project has so far generated high-quality short-read genome sequencing data from 2512 field samples from major malaria-endemic countries around the world. In addition, 12 reference-quality whole-genome assemblies were generated from long-read sequencing of cultured isolates. Variant calls and genotypes from the entire data set are publicly available at (https://www.malariagen.net/projects/pf3k). The raw sequencing reads are deposited in the European Nucelotide Archive (ENA) and accession numbers are listed in the metadata file on the project's website.

Our approach uses the Pf3k resource to build a graph reference genome for chromosome 10 of *P. falciparum*, with incorporated *DBLMSP* variation in 14 populations across Africa and Asia. The geographical distribution of the samples is shown in Table 5.1. We constructed a PRG representing 5 whole-genome assemblies from samples cultured in the laboratory (Dd2, HB3, IT, SN01, GA01) and the union of variants discovered in 2512 field samples by two independent pipelines – `GATK`, based on mapping reads to the primary reference genome and `Cortex`, based on *de novo* assembly

Country	Number of samples
Bangladesh	50
Cambodia	570
DR of the Congo	113
Ghana	617
Guinea	100
Laos	85
Malawi	369
Mali	96
Myanmar	60
Nigeria	5
Senegal	137
Thailand	148
The Gambia	65
Vietnam	97
Total	2512

Table 5.1: Pf3k samples and their country of provenance.

of reads (Fig. 5.5 (a) and (b)). See Methods Section B.1.1 for details on the input variants for the reference graph. We wrote a custom script for graph construction[10], due to the complexity of the region under study and the necessity to integrate call sets from different pipelines that represent variants in different formats.

Figure 5.6 shows the density of variable sites present in the reference graph along the gene and the number of alternative sequences they contain. The graph contains 84 variable sites across the *DBLMSP* gene, most of them corresponding to SNPs or short indels. However, there are 6 sites with tightly clustered variation, which contain a large number of long alternative sequences. The most diverse of them is in the DBL domain of the gene, with an average of 1 SNP every 3 bases. It is represented in the reference graph as a site containing 3,383 sequences of length 535bp, although only a subset of them have actually been observed in the Pf3k data set.

This is due to the fact that `Cortex` outputs alternative alleles in the form of long, assembled haplotypes in regions with high density of variants and we add artificial recombinants of such haplotypes to their site in the graph in order to make the rep-

[10]that is different from the one described in Section 4.3.1

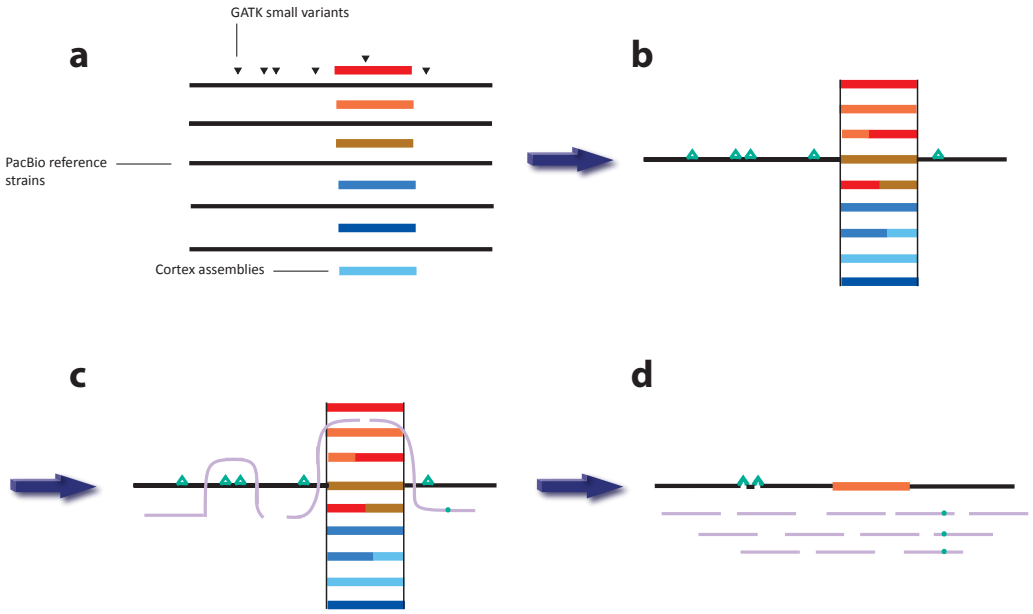


Figure 5.5: Schematic showing workflow for variant detection from a reference graph. **(a)** Firstly, current information about the variation catalogue, pooled from PacBio assemblies, long variant calls from `Cortex` and SNPs or short indels called by `GATK`, is integrated via multiple sequence alignments at variable loci. The figure depicts the DBL domain locus and surrounding SNPs. Colours represent different haplotypes, with the red spectrum representing the 3D7-like dimorphic class and the blue spectrum representing the other class. **(b)** Secondly, a variant site in the reference graph is created by listing observed alternative sequences from the alignment and their mosaics, representing recombinants, onto distinct edges. Adjacent variant sites are joined in the graph by an edge representing the non-variable sequence in between them. In the DBL domain, recombinants are allowed within dimorphic class (red/red, blue/blue) but not between classes (red/blue) – refer to Methods, Section B.2 for details. This graph will act as a generative model for sequences. **(c)** Thirdly, reads are aligned to the graph and coverage counts on alternative alleles are collected. **(d)** Finally, a sequence path consisting of alleles with highest read support is extracted from the reference graph and used as a personalised reference for the sample under analysis. Novel variants that were not present in the catalogue used to construct the graph can be called on top of this personalised reference with standard tools.

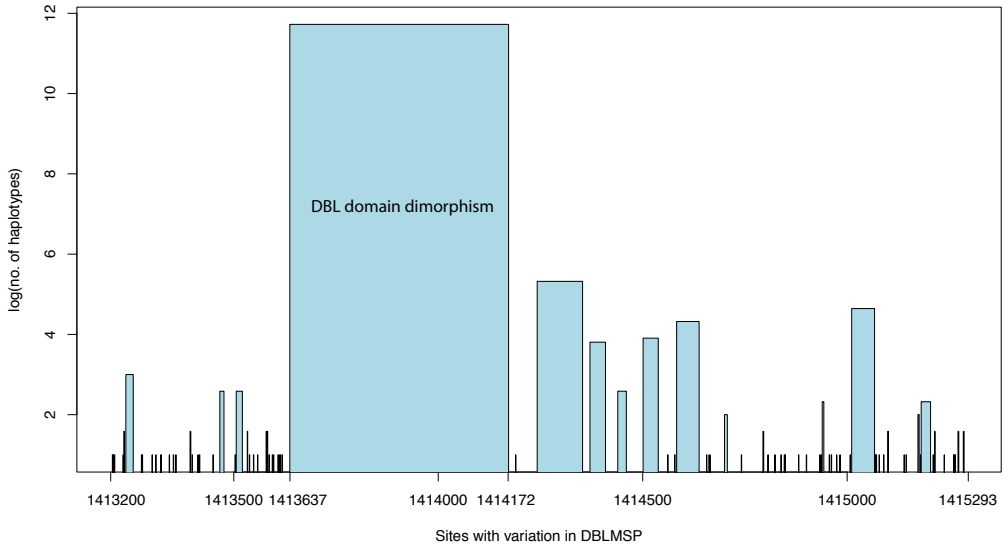


Figure 5.6: Bar plot outlining the reference graph structure. Each bar represents a variable site within the *DBLMSP* gene, plotted at its genomic coordinates, with height proportional to the logarithm (base 2) of the number of alternative sequences and width proportional to their length. The *log* scale is used for the y-axis because the site containing the dimorphic region of the DBL domain consists of 3,383 alternative haplotypes.

resentation more general (mosaics in Fig. 5.5 (b)). At variable sites outside the DBL domain, a shared *21*-mer between all alternative sequences is treated as a recombination breakpoint and all resulting haplotypic combinations are added to the graph. Within the DBL domain, such extended sharing does not exist and we impose the same rule to subsets of alleles belonging to the same dimorphic class (see Methods, Section B.2 for details).

A compact vBWT index of this reference graph was built by `gramtools` in order to enable quasimapping reads, as described in Chapter 3. See Methods, end of Section B.2, for the space requirements.

5.4 *DBLMSP* sequence inference for clonal samples

`Gramtools` was used to quasimap Illumina reads from 5 laboratory samples and 1316 field samples from the Pf3k data set to the reference graph described in the previous section. The mean read length across these samples is 93 bp and the mean depth of coverage is 75. We currently look only at this subset of samples because they were classified as unmixed by collaborators from the Pf3k project, using the software `DEploid` [Zhu et al., 2017]. Hence, the data should come from a haploid organism[11]. In this section, we evaluate the benefit of using the personalised references inferred by `gramtools` over the canonical 3D7 reference as a substrate for variant calling and sequence inference for the *DBLMSP* gene.

[11] `gramtools` only supports haploid inference at the moment, as discussed in Section 3.11

7G8	GB4	KE01	KH02	GN01
0	1	0	2	147

Table 5.2: Edit distance between PacBio assembly and graph-inferred reference in the *DBLMSP* gene validates the `gramtools` inference in 4/5 samples.

5.4.1 PacBio assemblies validate the personalised references inferred by `gramtools` in *DBLMSP*

We validated the `gramtools`-based sequence inference on 5 laboratory strains (7G8, KE01, KH02, GB4, GN01) sequenced both with the PacBio long-read technology and the Illumina short-read technology. Note that these samples were excluded from the reference graph construction process. High-quality whole-genome assemblies were generated by collaborators from the PacBio data as part of the Pf3k project. We considered these assemblies as truth and we aligned them with the *DBLMSP* sequence inferred by `gramtools` from Illumina data.

Table 5.2 shows the edit distance between the *DBLMSP* personalised reference inferred from the graph and the sequence extracted from the whole-genome assembly. The personalised reference is at most 2 SNPs apart from the PacBio sequence in 4/5 samples. Resulting pileups from mapping reads from these 4 samples with `BWA-MEM` to the graph-inferred references are displayed in Fig. 5.8, right. They show even coverage across the gene and the read consensus agrees with the PacBio assemblies, confirming that the small differences between the graph-inferred references and the true sequences can be detected by standard variant calling tools.

The 5th sample, GN01, is 147 nucleotides apart from the PacBio sequence and appears to be a recombinant between the 2 dimorphic classes. This causes a pileup gap when mapping to both 3D7 and the personalised reference (Fig. 5.7(a)). Aligning the sequence extracted from the PacBio assembly to 3D7 (Fig. 5.7(b)) shows that its DBL domain is nearly identical to 3D7 on the first ∼200 bp (class 1), but highly divergent on the following ∼400 bp (class 2). Because there are no paths in the

reference graph supporting class 1/class 2 recombinants[12] (see Methods, Section B.2), we would expect a personalised reference belonging to class 1 to yield a pileup gap of ∼400 bp and a personalised reference belonging to class 2 to yield a pileup gap of ∼200 bp. Indeed, `gramtools` inferred a class 1 reference that leads to a pileup with a gap of ∼400 bp in the divergent region (Fig. 5.7 (a), bottom). This sample is further discussed below, in Section 5.4.2.2.

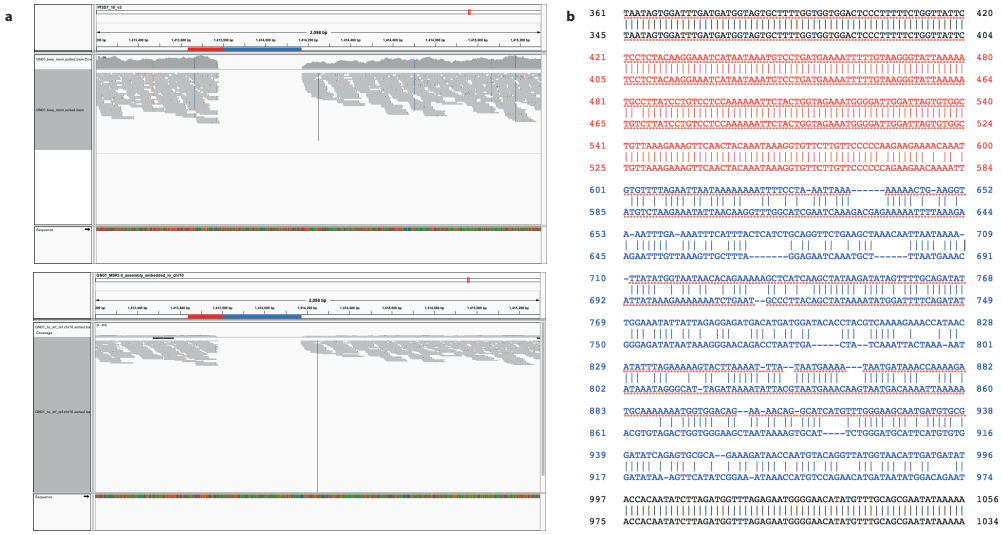


Figure 5.7: Validation strain GN01 has an inter-class recombinant in the dimorphic region of the DBL domain. **(a)** IGV screenshots of pileups with respect to the 3D7 reference (top) and the personalised reference (bottom) show a gap in the DBL domain, suggesting that the GN01 sequence is divergent from 3D7, but also from the haplotypes in the reference graph. Note that the appearance of reduced depth of coverage in the bottom pileup is a consequence of the IGV scaling. In fact, the raw values are the same. **(b)** Pairwise alignment between 3D7 and the GN01 sequence obtained from the PacBio assembly shows a section similar to 3D7 (red) and a divergent section (blue) in the dimorphic region.

We investigated the benefit of using graph-inferred personalised references over 3D7 for standard mapping. We compared pileups of reads mapped to each type of reference with `BWA-MEM`. In all validation samples, the pileups with respect to the graph-inferred reference illustrate an improvement from 3D7 in terms of edit distance between the read consensus and reference (Figures 5.7 (a) and 5.8). All 5 samples show

[12] red/blue recombinants in Fig. 5.5 (b)

more SNPs in the pileups with respect to 3D7 than with respect to the personalised reference. In the following section, we expand this comparison between references to Pf3k field samples.

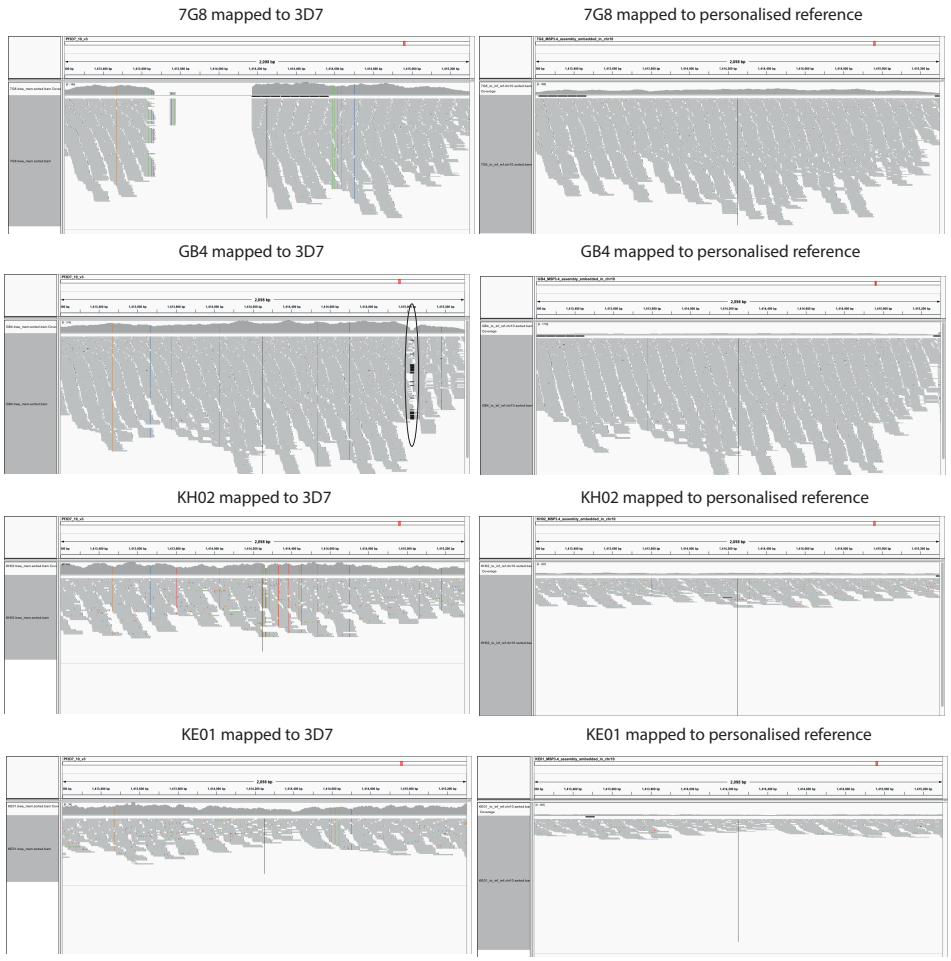


Figure 5.8: IGV screenshots of read pileups in the *DBLMSP* region for 4/5 validation samples. For 7G8, the pileup with respect to 3D7 has a gap of 476 nucleotides in the DBL domain, whereas the pileup with respect to the graph-inferred reference shows even coverage and 9/9 SNPs removed. For GB4, the pileup with respect to 3D7 presents a dip in coverage between coordinates 1415015-1415030 (circled), which gets smoothed out in the pileup with respect to the graph-inferred reference; in addition 7/8 SNPs are removed. For KH02 and KE01, the pileups with respect to 3D7 do not show any gaps, but show clustered SNPs which could produce alignment problems; the pileup with respect to the graph-inferred reference shows 18/20 SNPs removed for KH02 and 6/6 for KE01. Note that in these last 2 samples, the appearance of reduced depth of coverage in the pileups on the right compared to the ones on the left, is a consequence of the IGV scaling. In fact, the raw values are the same. Enlarged versions of each pileup can be found in Appendix Figs. A.3 - A.10.

5.4.2 Measures across pileups in field samples

To quantify the edit distance between a sample and the reference used for its sequence analysis, we measured the number of gaps and mismatches between the read consensus and the reference. Refer to Methods, Section B.3 for how we defined gaps and mismatches within a pileup. For 1316 clonal field samples, we mapped reads with `BWA-MEM` to both 3D7 and the graph-inferred reference and calculated these measures in the DBLMSP region from the resulting pileup files, as follows.

5.4.2.1 Initial measures show read pileups are closer to the personalised references than to 3D7

Figure 5.9 (left) shows overlaid histograms of the number of gaps in the read consensus from the 3D7 reference and the graph-inferred references. The bimodal distribution is consistent with the known allelic dimorphism in the *DBLMSP* gene. 465 samples show a reduction in the number of gaps when mapped to the graph-inferred reference and on average, that reduction is 264 gaps. 820 samples show no reduction because there are no gaps with respect to the 3D7 reference to begin with. Overall, the graph-inferred reference acts at least as well as 3D7 in 1295 samples.

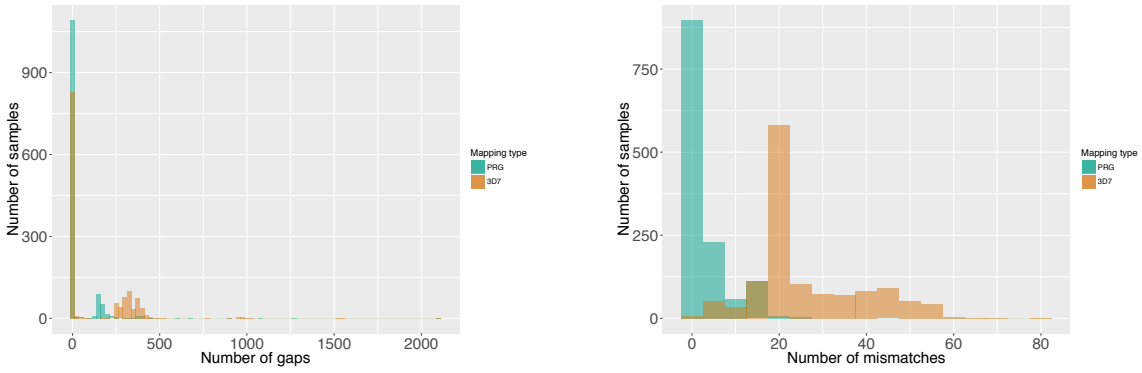


Figure 5.9: Histograms of measures calculated from pileups with respect to the 3D7 reference (orange) and the personalised references (green): number of gaps (left) and number of mismatches (right). Both gaps and mismatches are reduced when mapping to the personalised reference compared to 3D7.

Figure 5.9 (right) shows overlaid histograms of the number of mismatches in the read consensus from the 3D7 reference and the graph-inferred references, again with a bimodal distribution. 1299 samples show a reduction in the number of mismatches when mapped to the graph-inferred reference and on average, that reduction is 22.9 mismatches. The graph-inferred references have at most 24 new variants on top[13], usually isolated and easy to detect by standard variant calling tools.

5.4.2.2 Discovery of 2 novel haplotypes

Interestingly, for 229 samples, we still see gaps in read pileups even after mapping to the graph-inferred references. The positions of these gaps along the gene are plotted in Fig. 5.10 (a). A consistent gap within the DBL domain is observed across many samples, with average size 194 bp. A possible cause is that this subset of samples contains variation that is not captured in the reference graph. By mapping raw contigs[14] assembled by `Cortex` to the personalised references inferred from the graph, we identified 2 samples that had a contig spanning their pileup gap (see Methods, Section B.4 for details).

We confirmed that the alleles derived from the 2 novel contigs were not present in the reference graph and added them to the large DBL locus. Then the entire analysis described in the previous section was rerun on the new reference graph, including read quasimapping, personalised reference inference and pileup measures calculation. Results are presented in the next section.

[13] *DBLMSP* gene length is 2094 bp
[14] refer to Section 2.4.3

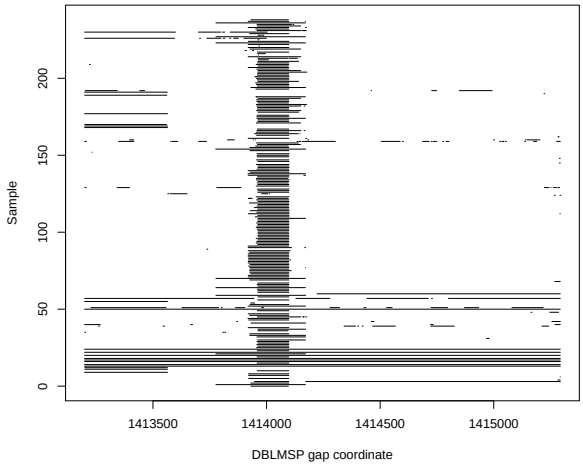


Figure 5.10: Genomic coordinates of regions without read support when mapping to the personalised references in samples where the DBLMSP sequence cannot be fully resolved. The majority of problematic samples (including the GN01 strain used for validation) contains a gap in the DBL domain between coordinates 1413958 and 1414100. We identified 2 samples where `Cortex` had assembled contigs spanning this gap.

5.4.2.3 Updating the reference graph with 2 novel haplotypes further improves sequence inference

Figure 5.11 (left) shows overlaid histograms of the number of gaps between the read consensus, the 3D7 reference and the personalised references inferred from the updated graph respectively. Pileups with respect to graph-inferred references are at least as good as the ones with respect to 3D7 in all but 3 samples. 480 samples show a reduction in the number of gaps when mapped to the graph-inferred reference as opposed to 3D7 and on average, that reduction is 318.6 gaps. There are 97 samples that are left with gaps and on average, they have 161.5 gaps. See Appendix Fig. A.1 for the distribution of these gaps along the gene. The consistent gap in Fig. 5.10 is eliminated.

Figure 5.11 (right) shows overlaid histograms of the number of mismatches in the read consensus from the 3D7 reference and the references inferred from the updated

graph. 1310 samples show a reduction in the number of mismatches when mapped to the graph-inferred reference and on average, that reduction is 23.5 mismatches. The graph-inferred references have at most 19 new variants on top, usually isolated and easy to detect by standard variant calling tools.

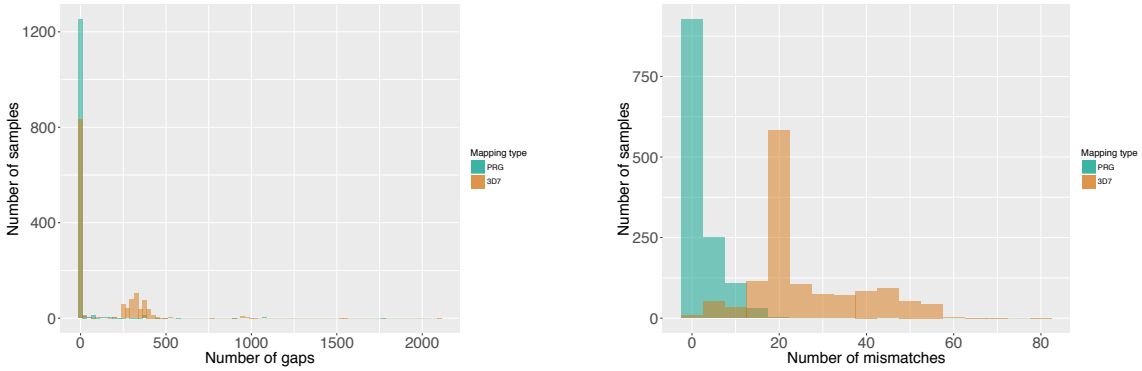


Figure 5.11: Histograms of gaps and mismatches calculated from pileups with respect to the 3D7 reference and the personalised references inferred from the graph updated with 2 additional haplotypes discovered from raw `Cortex` contigs. Inference is further improved, with the personalised references closing the gaps produced by mapping to 3D7 in 132 additional samples compared to the ones resolved by the previous reference graph in Fig. 5.9.

In summary, we started with 1316 samples, 820 of which had *DBLMSP* sequence fully resolved by mapping to 3D7. After mapping to the initial reference graph, 1079 samples were fully resolved, increasing to 1211 after updating the reference graph with 2 novel alleles derived from `Cortex` contigs, that initially went undiscovered.

5.5 Population frequencies

We can now build a map of the diversity across the gene's DBL domain within and between populations from 1241 samples with fully accessible sequence (i.e. without gaps in the DBL domain). We see the 2 known forms at common frequencies in all 14 countries, which is consistent with previous claims of balancing selection (Fig. 5.12). However, there is striking variation between populations in the diversity and type of

alleles found. African countries show similar allele frequency patterns, with 2 dominant alleles from the 3D7-like type (allele5 and allele0) and 1 dominant allele from the divergent type (allele2). By contrast, countries in SE Asia are more heterogeneous, with multiple alleles maintained at similar lower frequencies, despite being less diverse than Africa in terms of the number of distinct alleles found. In particular, there seems to be no clear dominant allele from the divergent class in Cambodia, Vietnam and Thailand. We can also observe high-frequency alleles that are specific to one country, such as allele1 in The Gambia and allele26 in Thailand.

In total, 108 alleles were observed in the DBL domain, 57 from class 1/3D7-like and 51 from class 2, with a frequency of class 1 alleles of 65% and a frequency of class 2 alleles of 35%. These alleles render a total nucleotide diversity of $\pi = 0.086$, with contributions from within class 1 $\pi_{11} = 0.0015$, from within class 2 $\pi_{22} = 0.0045$ and between the 2 classes $\pi_{12} = 0.08$.

5.6 Impact of size of reference panel on inference

In this section, we explore how the `gramtools` inference of personalised references changes with the number of samples used in the reference graph construction. Using the same method as in Section 5.3, we built graphs from 2, 10, 50 and 100 samples from Ghana ('training' samples). Then we took a separate set of 140 samples from Ghana as well ('validation' samples) and quasimapped them to each of these graphs, inferred personalised references and calculated gaps and mismatches metrics from the respective pileups.

For the gaps metric (Fig. 5.13, bottom), the biggest improvement is observed when going from a single reference sequence to a graph of 2 samples, where each sample has one version of the dimorphism. This was expected since gaps in pileups are produced by sequences too divergent from the reference, which belong to the second dimorphic

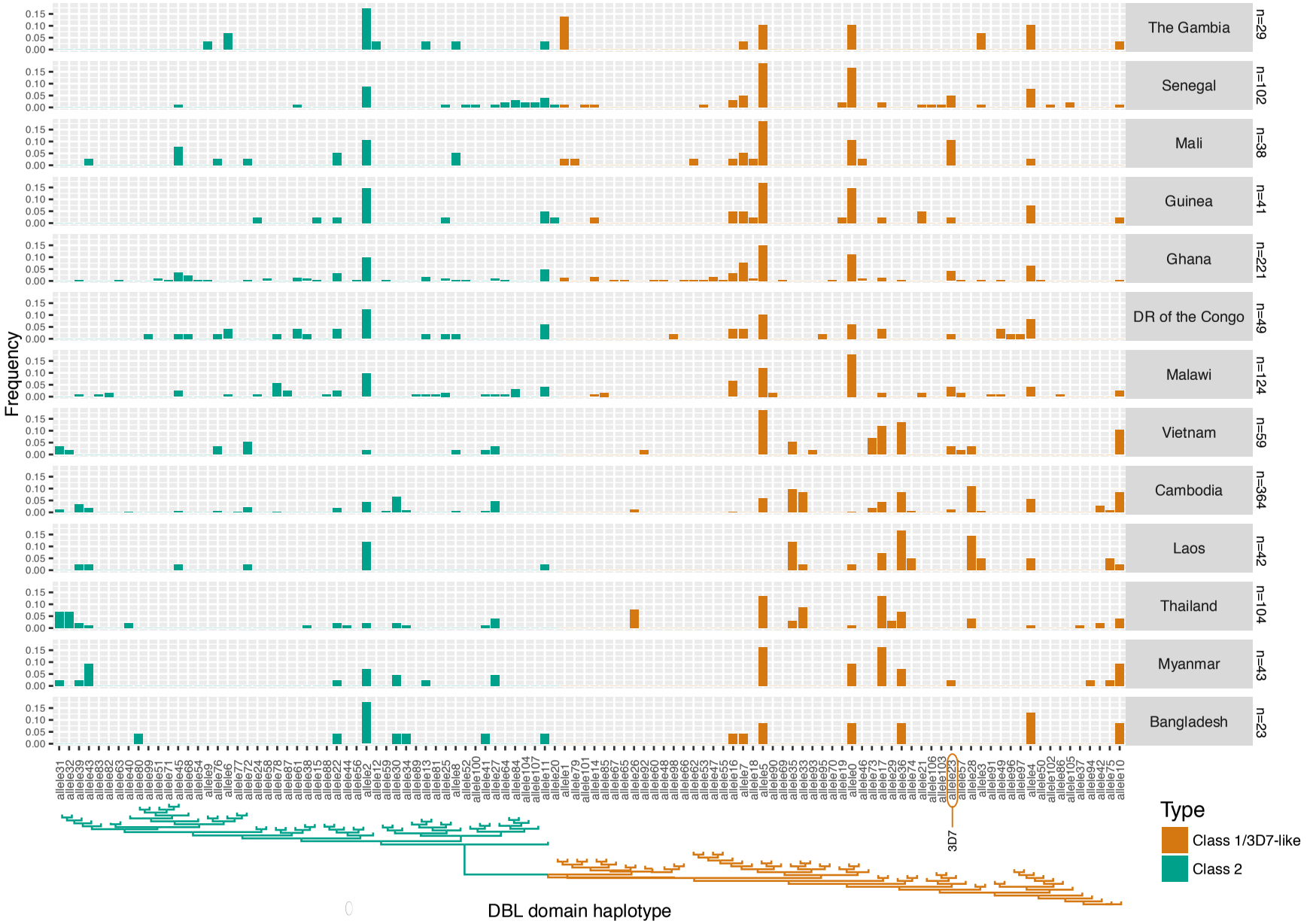


Figure 5.12: Population level frequencies of haplotypes in the DBL domain of the gene. The phylogenetic tree on the bottom shows they come from two deeply diverged lineages that are preserved across populations, suggesting they are under balancing selection. `ClustalW` was used to build the phylogenetic tree.

class. Once a representative from this class is present in the graph and can be used as a reference, many samples from the same class are expected to be within a small edit distance because of the low within-class diversity. In the initial mapping to 3D7, 81 samples show no pileup gaps, increasing to 99 samples when mapping to the graph of 2 samples. Using the graph of 10 samples closes the gaps in a further 3 samples, but no further gaps are closed when using larger graphs of 50 or 100 samples. However, in some of the remaining samples with open gaps, the number of gaps changes slightly with the size of the graph, causing the graphs of 10, 50 and 100 samples to perform slightly worse than the graph of 2 samples in terms of mean number of gaps.

Increasing the number of samples in the reference graph from 1, to 2, to 10, to 50, shows a consistent improvement in terms of the number of mismatches between the read consensus and the personalised reference (Fig. 5.13, top). However, the graph built from 100 samples performs slightly worse than the one built from 50 samples.

We suspect the reason larger graphs can perform slightly worse is that increasing the number of alternative alleles increases the probability of choosing a reference that is not the closest to the sample analysed in cases when it contains novel variation. This is particularly relevant to large graph sites with many long alleles, like the DBL site in our current graph. As described in Section 3.9, if a sample has a variant that is not present in the graph, reads overlapping that variant do not quasimap exactly and therefore do not contribute to coverages counted on the alternative alleles of variants that are nearby the novel one (within a smaller distance than the read length). As a consequence of this reduced information, sometimes the choice between those alternative alleles is made arbitrarily. In a larger graph, there are more options, but many of them have not actually been observed (see Section 5.3), so it is possible, by chance, to choose one that is more distant from the sample than the alternatives from the same graph or a smaller one.

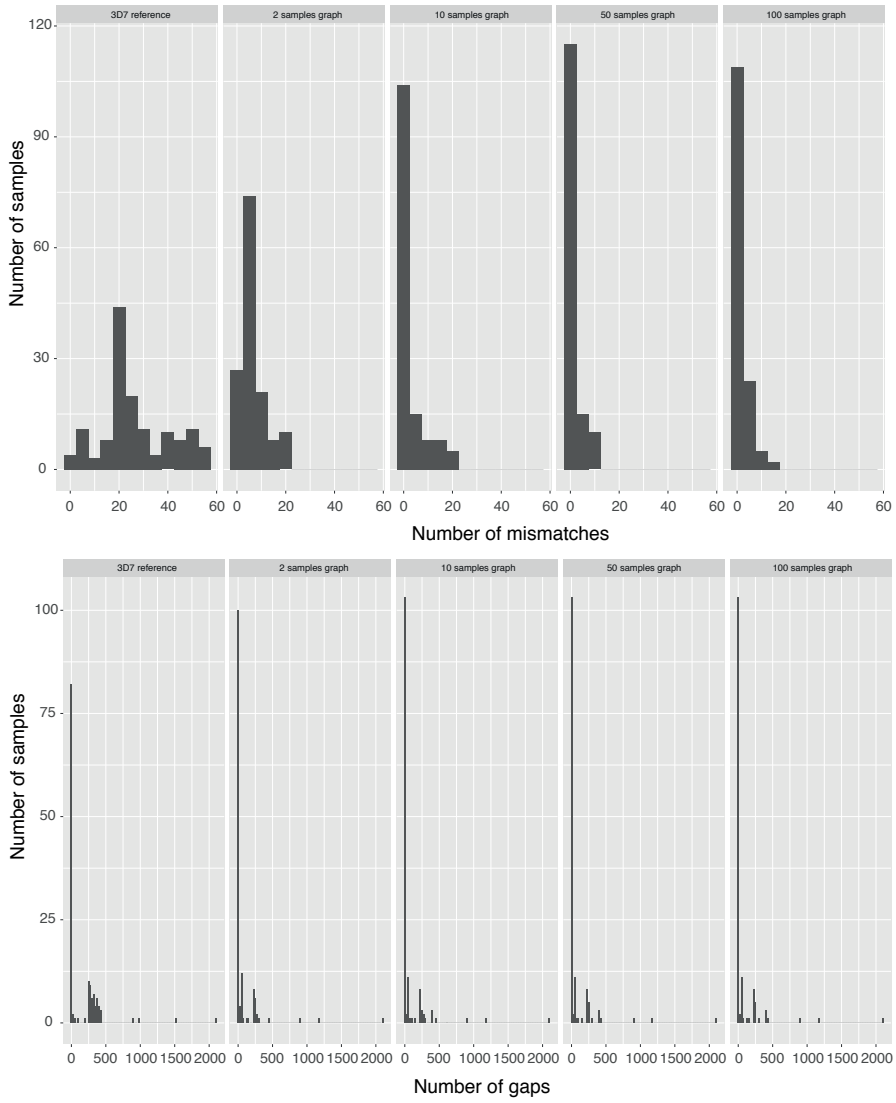


Figure 5.13: Measures calculated from pileups with respect to personalised references inferred from graphs built out of 1, 2, 10, 50 and 100 samples respectively. The top plot shows the number of mismatches gradually decreases as more samples are added to the reference graph, with one exception: the graph consisting of 100 samples performs worse than the one consisting of 50 samples. Mean number of mismatches in each plot, from left to right, is 26.7, 6.3, 2.9, 1.5 and 1.9 respectively. The bottom plot shows the biggest improvement in number of gaps is observed when extending 3D7 with one extra sample containing a class 2 haplotype in the DBL domain, as expected from the dimorphic structure. Mean number of gaps in each plot, from left to right, is 154, 70, 72.2, 71.7 and 71.7 respectively.

5.7 *DBLMSP* and *DBLMSP2* joint analysis

5.7.1 Motivation

One of the novel *DBLMSP* haplotypes discovered in the DBL domain in Section 5.4.2.2 contains a 278 bp sequence identical to a segment of the DBL domain of the *DBLMSP2* gene[15]. The same repeated sequence is also contained in the validation strain GN01 and the PacBio assembly confirms it is not an artifact (see Methods, Fig. B.4). Such long tract of identical sequence between the two genes is likely to have occurred through non-allelic gene conversion. This opened up the possibility of these two related genes to have co-evolved, which would imply that subsets of their alleles have extensive homology. In that case, it can be difficult to place reads to the correct gene, especially if only one of them is represented with variants in the reference structure. In our current reference graph representation, which includes variants from *DBLMSP* only, there is a risk of *DBLMSP* absorbing reads from a non-reference, paralogous *DBLMSP2* allele simply because the respective *DBLMSP2* allele is not included in the graph. Hence, having variants of both genes represented in the reference graph should inform the inference process for both of them because it reduces this risk. In the remainder of this chapter, we present a sequence analysis of *DBLMSP* and *DBLMSP2* based on a reference graph that incorporates variants in both genes and try to understand their evolutionary relationships.

5.7.2 PRG construction

We constructed a reference graph containing variation present in *DBLMSP* and *DBLMSP2* from the same reference panel described in Section 5.3: 5 PacBio whole-genome assemblies (Dd2, HB3, IT, SN01, GA01) and variants discovered by `GATK` and `Cortex` in 2512 field samples from the Pf3k project. In the *DBLMSP* region, we also

[15] 3D7 version

added the 2 novel haplotypes discovered in Section 5.4.2.2 and the novel sequence extracted from the PacBio assembly of sample GN01[16] (refer to Methods, Section B.1.2 for details). Throughout this section, we will refer to this reference graph as 'joint graph', or 'joint PRG'; the reference graph in the Section 5.3, constructed only from *DBLMSP* variation, will sometimes be referred as 'single gene graph' or 'single PRG'.

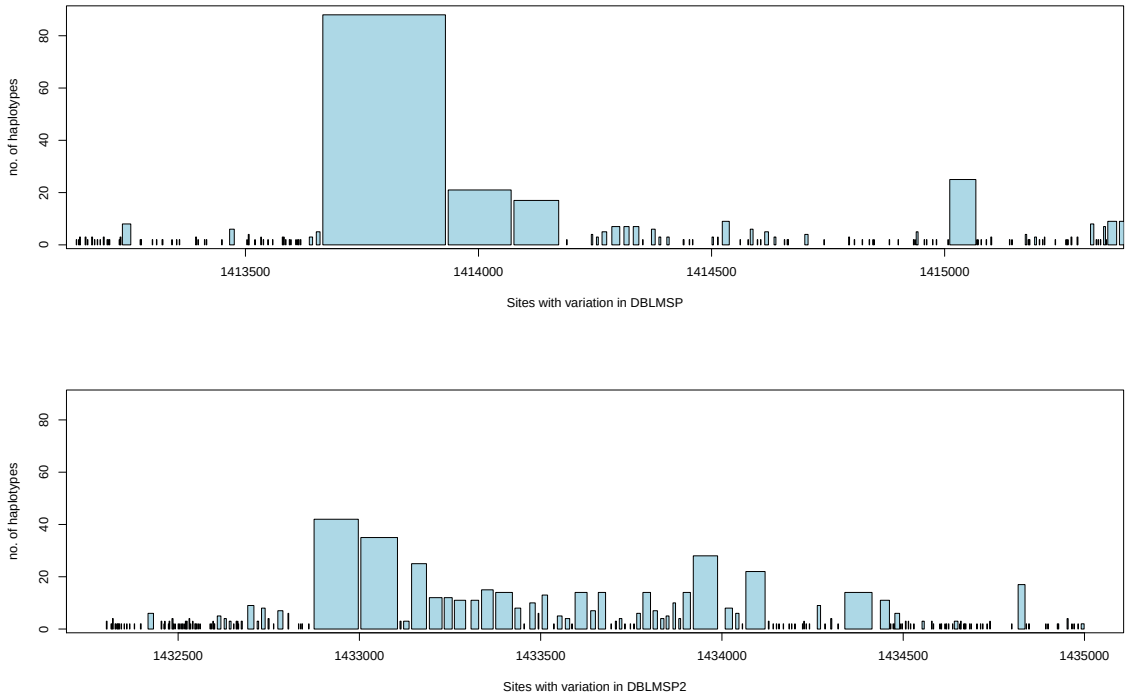


Figure 5.14: Bar plot outlining the reference graph structure in *DBLMSP* (top) and *DBLMSP2* (bottom). Each bar represents a variable site plotted at its genomic coordinates, with height proportional to the number of alternative sequences and width proportional to their length.

The new reference graph contains 107 variable sites across *DBLMSP*, 130 variable sites across *DBLMSP2* and 76 variable sites surrounding the two genes; in total 313 sites. The DBL domain of *DBLMSP2* is comprised of 34 variable loci containing up to 42 alternative sequences. The DBL domain of *DBLMSP* is mainly comprised of 3 variable loci containing 88, 21 and 17 alternative sequences, respectively.

[16]see Section 5.7

Note that the graph structure in *DBLMSP* is different from the one in Section 5.3, Fig. 5.6, where the DBL domain was represented by a single larger site[17], due to a different parametrisation of the graph construction script (see Methods, Section B.2). The current graph presents a higher granularity of variant sites because we chose to represent recombination more naturally than previously. Instead of adding mosaic sequences, large variant sites were broken down at regions of sequence similarity and graph paths were created to represent allelic combinations between adjacent sites (see Methods, Section B.2, Fig. B.1). This choice should limit the issues caused by large graph sites with many alternative alleles, such as missing recombinants, as in Section 5.4.1, or leading to a very divergent allele in samples with novel variation, as in Section 5.6.

A compact vBWT index of this reference graph was built by `gramtools` in order to enable quasimapping reads. See Methods, end of Section B.2, for the space requirements.

5.7.3 Sequence inference for clonal samples

We used `gramtools` to analyse the same set of samples[18] described in Section 5.4 with the new reference graph for both genes.

5.7.3.1 PacBio assemblies validate the personalised references inferred by `gramtools` in *DBLMSP* and *DBLMSP2*

We validated the new personalised references against the PacBio assemblies of the 5 laboratory strains[19] used in Section 5.4. Table 5.3 shows the edit distance between the personalised reference inferred from the graph and the sequence extracted from the whole-genome assembly for both genes. For *DBLMSP*, the result is similar to

[17] containing 3,383 alternative sequences
[18] 5 laboratory samples and 1316 clonal samples from field
[19] 7G8, KE01, KH02, GB4, GN01

	7G8	GB4	KE01	KH02	GN01
DBLMSP	1	1	0	1	0
DBLMSP2	24	1	4	13	12

Table 5.3: Edit distance between PacBio assembly and graph-inferred reference in *DBLMSP* and *DBLMSP2*.

the one obtained through the single gene graph analysis (Section 5.4.1, Table 5.2). Note that sample GN01 is resolved with the current graph, while previously the edit distance was 147.

In *DBLMSP2*, 4/5 samples are resolved, being within 4 variants from the true sequence. Samples KH02 and GN01 contain a 12 bp deletion, hence the larger edit distance in the table. Apart from GN01, all samples contain a 1 bp deletion. It is possible that these 2 deletions occur due to errors in the graph construction rather than due to undiscovered variants missing from the reference graph. Pileups reveal that the larger edit distance in 7G8 is due to a gap of 76bp with no read support, apart from a small number of multimapped reads. Failure to resolve this stretch of sequence could be due either to variants missing from the graph or to a repeat somewhere else in the genome that absorbs the rest of reads. If it is the former reason, extracting the corresponding sequence from the PacBio assembly and adding it to the graph should fix the issue, as in the case of *DBLMSP* in sample GN01.

5.7.3.2 Read pileups are closer to the personalised references than to 3D7 in *DBLMSP* and *DBLMSP2*

In this section, we quantify the benefit of mapping to personalised references inferred from the graph of both genes over the 3D7 reference. We use the same measures as in Section 5.4.2.1, the number of mismatches and gaps between the read consensus and the reference for 1316 clonal samples.

As expected, the results for *DBLMSP* are consistent to the ones in Section 5.4.2.3 (Fig. 5.11). The *DBLMSP* sequence is fully resolved in all but 83 samples that are

left with gaps in the pileups with respect to their personalised references.

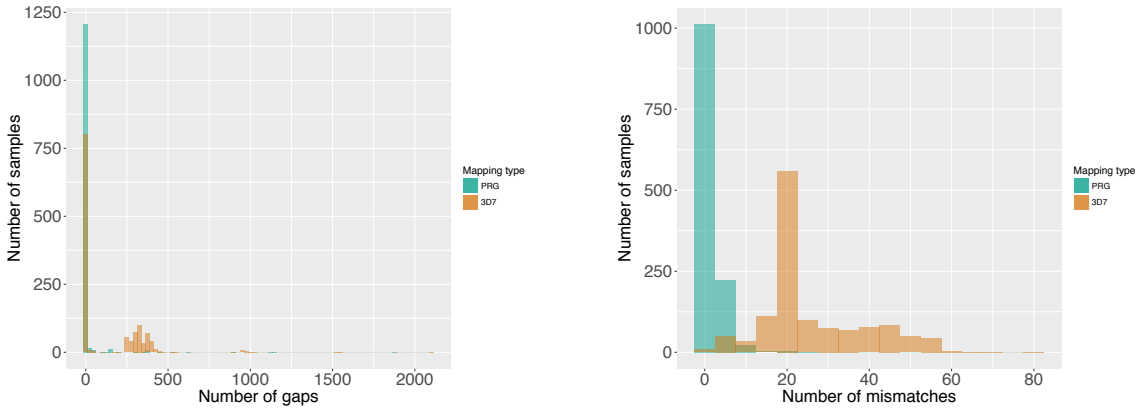


Figure 5.15: Histograms of measures calculated from pileups in *DBLMSP* with respect to the 3D7 reference and the personalised references inferred from the joint graph: number of gaps (left) and number of mismatches (right). Results are consistent with the ones obtained from the single gene graph.

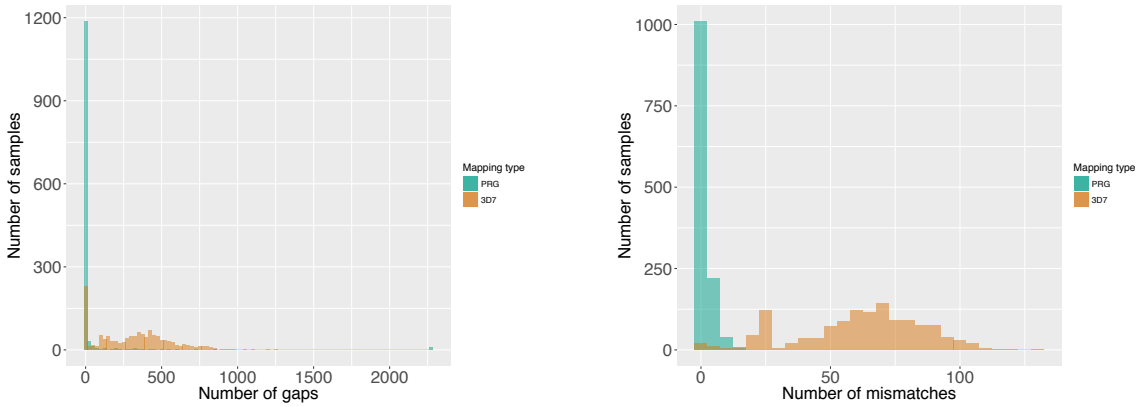


Figure 5.16: Histograms of measures calculated from pileups in *DBLMSP2* with respect to the 3D7 reference and the personalised references inferred from the joint graph: number of gaps (left) and number of mismatches (right). In the majority of samples, both gaps and mismatches are reduced to 0 when mapping to the personalised references.

Figure 5.16 shows the number of gaps between the read consensus, the 3D7 reference and the *DBLMSP2* references inferred from the joint graph respectively. 1051 samples show a reduction in the number of gaps when mapped to the personalised reference and on average, that reduction is 364.5 gaps. 209 samples show no reduction because there are no gaps with respect to the 3D7 reference to begin with.

The *DBLMSP2* sequence is fully resolved in all but 129 samples that are left with gaps in the pileups with respect to their personalised references. The average size of these gaps is 236.9 bp and their distribution along the gene is illustrated in Appendix Fig. A.2. A consistent gap in the DBL domain can be observed in a fraction of samples. That specific sequence might be identified by taking a similar approach to the one described in Section 5.4.2.2, i.e. mapping raw assembly contigs to the personalised references and checking if any of them cover the gap.

5.7.4 Gene conversion between *DBLMSP* and *DBLMSP2*

For the remainder of this chapter, we will perform a preliminary population genetic analysis for *DBLMSP* and *DBLMSP2*, based on the alleles inferred by `gramtools`. We only use the 1092 samples with fully resolved sequence (no gaps in pileups) in both genes. Despite extended homology between the 2 genes in some regions (see Motivation, Section 5.7.1), there is enough overall dissimilarity (54%-60% nucleotide identity in the PacBio assemblies, driven by the 200 bp difference in gene length) to be confident that the sequence inferred by `gramtools` is assigned to the correct gene in this set of samples.

We built a phylogenetic tree from alleles over the dimorphic region of both genes (Fig. 5.17). There can be observed 3 major allelic classes in between the two genes: one that is specific to *DBLMSP* (left, orange), one that is specific to *DBLMSP2* (bottom, pink) and one that contains alleles from both genes (top, green). We label the *DBLMSP* gene-specific alleles as class 1, the overlapping alleles from both genes as class 2 and the *DBLMSP2* gene-specific alleles as class 3. As before, the 3D7 allele belongs to class 1 in *DBLMSP*, whereas in *DBLMSP2*, it belongs to class 2. We note that in the outgroup species *Plasmodium reichenowi*, the *DBLMSP* sequence is very similar to one of the class 1 alleles (93% nucleotide identity) and the *DBLMSP2* sequence is very similar to one of the class 3 alleles (93% nucleotide identity), indicating

	DBLMSP Class 1	*DBLMSP2* Class 3	Both Class 2
DBLMSP Class 1	0.02	0.67	0.67
DBLMSP2 Class 3		0.06	0.23
Both Class 2			0.08

Table 5.4: Average pairwise distance from sequences within and between the 3 main clades in the phylogenetic tree in Fig. 5.17, calculated with the `dist.dna` function from the *ape* package in R.

that these allelic classes have been maintained over long periods of time.

The non-gene-specific clade has class 2 alleles from *DBLMSP* interleaved with alleles from *DBLMSP2*. The multiple sequence alignment of a few alleles from this class from both genes (Fig. 5.18) shows long stretches of identical sequence and SNPs shared by alleles from both genes, suggesting they have been transferred by non-allelic gene conversion. Table 5.4 shows the mean nucleotide distance within and between the three main clades in the phylogenetic tree. Interestingly, class 1 alleles in *DBLMSP* are just as distant from *DBLMSP* class 2 alleles as they are from all alleles in *DBLMSP2* (class 2 and class 3). This suggests that class 2 and class 3 alleles have close evolutionary relationships, whereas class 1 alleles have a completely different history. We suspected class 1 alleles might have been acquired in *DBLMSP* from some other region of the genome, so we searched for them in the whole-genome PacBio assemblies, but no hits outside *DBLMSP* were returned.

5.7.5 Population frequencies

5.7.5.1 Allele frequencies

We built a global diversity map for the dimorphic block of the DBL domains of the two genes. Fig. 5.19 shows the total frequency of each dimorphic allelic class and Figs. 5.20 and 5.21 show individual allele frequencies per country.

The two dimorphic forms of *DBLMSP2* are observed at common frequencies in all 14 populations, indicating this gene is under balancing selection too (Fig. 5.21).

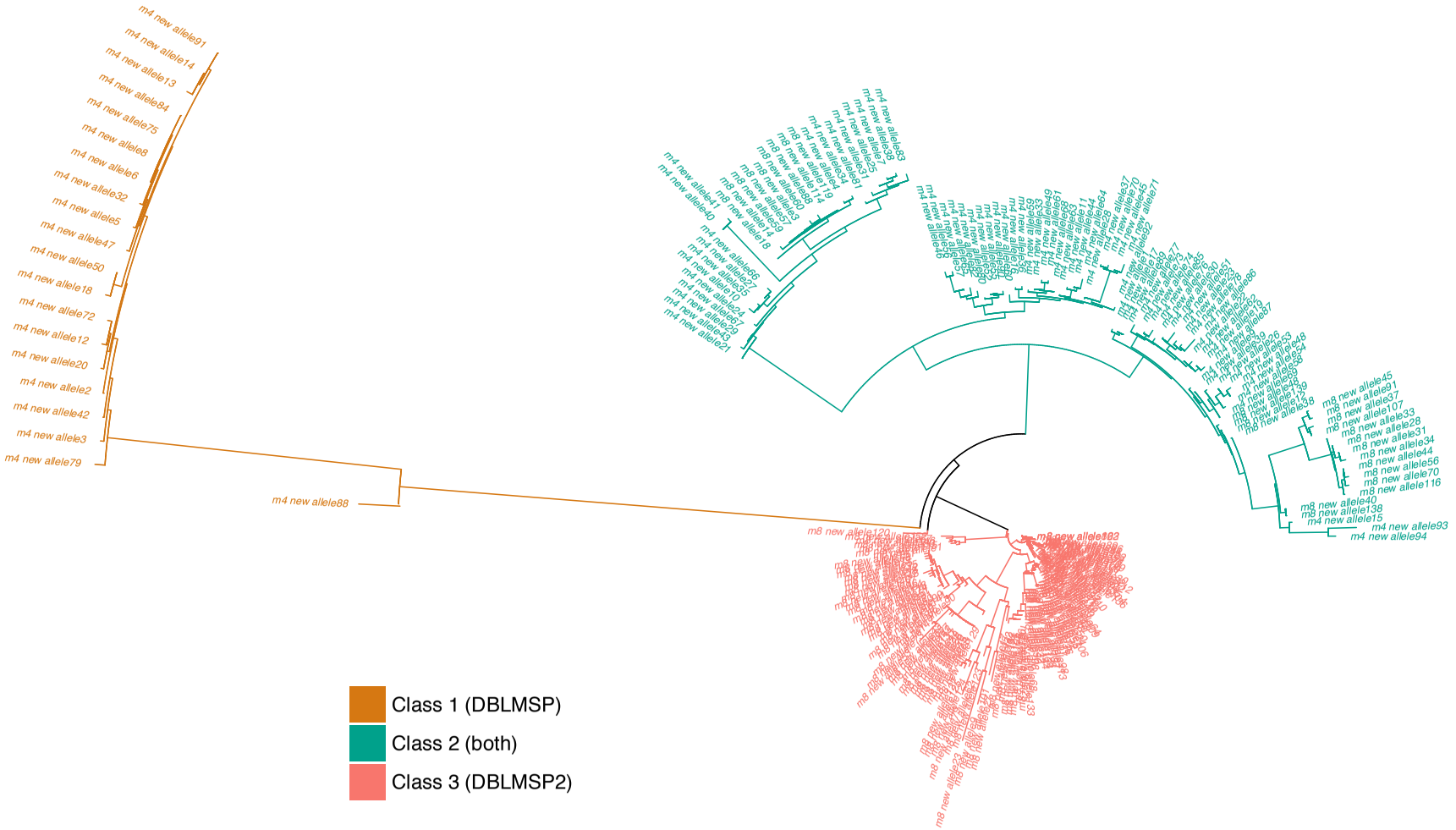


Figure 5.17: Phylogenetic tree of all fully resolved alleles from the dimorphic region of the DBL domains of *DBLMSP* and *DBLMSP2*, obtained from 1092 samples. Alleles with the "m4" prefix come from *DBLMSP* and alleles with the "m8" prefix come from *DBLMSP2*. We used the `bionj` function from the *ape* package in R [Paradis et al., 2004] to build the tree.

```
m8_new_allele45       aatgtcagaaatttcttaacagttaataaaggtgtccttgttcccccaagaagaaaacaa
m8_new_allele91       aatgtcagaaatttcttaacagttaataaaggtgtccttgttcccccaagaagaaaacaa
m8_new_allele40       aatgtcagaaatttcttaacagttaataaaggtgtccttgttcccccaagaagaaaacaa
m8_new_allele138      aatgtcagaaatttcttaacagttaataaaggtgtccttgttcccccaagaagaaaacaa
m8_new_allele139      aatgtcagaaatttcttaacagttaataaaggtgtccttgttcccccaagaagaaaacaa
m8_new_allele48       aatgtcagaaatttcttaacagttaataaaggtgtccttgttcccccaagaagaaaacaa
m8_new_allele12       aatgtcagaaatttcttaacagttaataaaggtgtccttgttcccccaagaagaaaacaa
m8_new_allele38       aatgtcagaaatttcttaacagttaataaaggtgtccttgttcccccaagaagaaaacaa
m4_new_allele15       ------------------------------------------------agaagaaaacaa
m4_new_allele71       ------------------------------------------------agaagaaaacaa
m4_new_allele70       ------------------------------------------------agaagaaaacaa
m4_new_allele37       ------------------------------------------------agaagaaaacaa
m4_new_allele45       ------------------------------------------------agaagaaaacaa
                                                                      ************

m8_new_allele45       atgtgttttagaattaatataaacaattttcctaaattaaaaaaaactgaaggtaaattt
m8_new_allele91       atgtgttttagaattaatataaacaattttcctaaattaaaaaaaactgaaggtaaattt
m8_new_allele40       atgtgttttagaattaatataaacaattttcctaaattaaaaaaaactgaaggtaaattt
m8_new_allele138      atgtgttttagaattaatataaacaattttcctaaattaaaaaaaactgaaggtaaattt
m8_new_allele139      atgtgttttagaattaatataaacaattttcctaaattaaaaaaaactgaaggtaaattt
m8_new_allele48       atgtgttttagaattaatataaacaattttcctaaattaaaaaaaactgaaggtaaattt
m8_new_allele12       atgtgttttagaattaatataaacaattttcctaaattaaaaaaaactgaaggtaaattt
m8_new_allele38       atgtgttttagaattaatataaacaattttcctaaattaaaaaaaactgaaggtaaattt
m4_new_allele15       atgtgttttagaattaatataaacaattttcctgaattaaaaaaaactgaaggtaaattt
m4_new_allele71       atgtgttttagaattaatataaacaattttcctgaattaaaaaaaactgaaggtaaattt
m4_new_allele70       atgtgttttagaattaatataaacaattttcctaaattaaaaaaaactgaaggtaaattt
m4_new_allele37       atgtgttttagaattaatataaacaattttcctgaattaaaaaaaactgaaggtaaattt
m4_new_allele45       atgtgttttagaattaatataaacaattttcctgaattaaaaaaaactgaaggtaaattt
                      ********************************* ************************

m8_new_allele45       gaaaatttcatttactcatctgcaggttctgaagctaaacaattaataaaattatatggt
m8_new_allele91       gaaaatttcatttactcatctgcaggttctgaagctaaacaattaataaaattatatggt
m8_new_allele40       gaaaatttcatttactcatctgcaggttctgaagctaaacaattaataaaattatatggt
m8_new_allele138      gaaaatttcatttactcatctgcaggttctgaagctaaacaattaataaaattatatggt
m8_new_allele139      gaaaatttcatttactcatctgcaggttctgaagctaaacaattaataaaattatatggt
m8_new_allele48       gaaaatttcatttactcatctgcaggttctgaagctaaacaattaataaaattatatggt
m8_new_allele12       gaaaatttcatttactcatctgcaggttctgaagctaaacaattaataaaattatatggt
m8_new_allele38       gaaaatttcatttactcatctgcaggttctgaagctaaacaattaataaaattatatggt
m4_new_allele15       gaaaatttcatttactcatctgcaggttctgaagctaaacaattaataaaattatatggt
m4_new_allele71       gaaaatttcatttactcatctgcaggttctgaagctaaacaattaataaaattatatggt
m4_new_allele70       gaaaatttcatttactcatctgcaggttctgaagctaaacaattaataaaattatatggt
m4_new_allele37       gaaaatttcatttactcatctgcaggttctgaagctaaacaattaataaaattatatggt
m4_new_allele45       gaaaatttcatttactcatctgcaggttctgaagctaaacaattaataaaattatatggt
                      ************************************************************

m8_new_allele45       aataacacagaaaaagctcatcaagctataagatatagttttgcagatattggaaatatt
m8_new_allele91       aataacacagaaaaagctcatcaagctataagatatagttttgcagatattggaaatatt
m8_new_allele40       aataacacagaaaaagctcatcaagctataagatatagttttgcagatattggaaatatt
m8_new_allele138      aataacacagaaaaagctcatcaagctataagatatagttttgcagatattggaaatatt
m8_new_allele139      aataacacagaaaaagctcatcaagctataagatatagttttgcagatattggaaatatt
m8_new_allele48       aataacacagaaaaagctcatcaagctataagatatagttttgcagatattggaaatatt
m8_new_allele12       aataacacagaaaaagctcatcaagctataagatatagttttgcagatattggaaatatt
m8_new_allele38       aataacacagaaaaagctcatcaagctataagatatagttttgcagatattggaaatatt
m4_new_allele15       aataacacagaaaaagctcatcaagcaatgaaatatagttttgcagatattggaaatatt
m4_new_allele71       aataacacagaaaaagctcttcaagcaatgaaatatagttttgcagatattggaaatatt
m4_new_allele70       aataacacagaaaaagctcatcaagctataaaatatagttttgcagatattggaaatatt
m4_new_allele37       aataacacagaaaaagctcatcaagctataaaatatagttttgcagatattggaaatatt
m4_new_allele45       aataacacagaaaaagctcttcaagctataaaatatagttttgcagatattggaaatatt
                      ******************* ****** ** * ****************************
```

Figure 5.18: Multiple sequence alignment of a few Class 2 alleles from both *DBLMSP* and *DBLMSP2* (belonging to green clade in Fig. 5.17). Alleles with the "m4" prefix come from *DBLMSP* and alleles with the "m8" prefix come from *DBLMSP2*. The alignment was obtained with `ClustalW`.

Overall, there are 4 dominant alleles from class 3 (alleles 4,8,2,6) and one from class 2 (allele3). As before, African countries have a clear dominant allele from class 2 (allele3), whereas in Asia multiple alleles are maintained at similar frequencies. There is much more heterogeneity between populations than in *DBLMSP*, with high variability in the levels of diversity, the type of alleles present and their relative frequencies. For instance, the frequency of class 2 alleles in Senegal is only 5%, whereas in Laos is 47%. The unusually low frequency of class 2 alleles in Senegal (see Fig. 5.19) could be explained by recent intervention strategies that have reduced the levels of diversity in parasite populations. We cannot observe any obvious continent-specific allele patterns, but there are some alleles that are country-specific (alleles 1,10,11,5 in Gambia, allele28 in Thailand).

Note that the allele frequency plot for *DBLMSP* is different from the one in section 5.6 (Fig. 5.12) because it represents a smaller locus. The alleles in Fig. 5.12 represent the entire DBL domain, whereas the alleles in Fig. 5.20 represent only its dimorphic region. As a result, Fig. 5.20 illustrates the dimorphism more clearly, with the dominant alleles from each class (allele2 and allele4) being placed at the extremes of the phylogenetic tree. Similarity can be observed between the African countries in terms of the dominant alleles and their relative frequencies (alleles 2,3,6,14 and 4,9). Asian countries are more heterogeneous, especially in class 2 alleles, for which there is no dominant type in Thailand, Cambodia and Vietnam, but multiple alleles are maintained at 5% frequencies. We observe continent-specific alleles (allele18 at $\sim$10% frequency in all Asian countries but absent in Africa, allele8 much rarer in Africa than in Asia, allele17 absent in Asia), but also country-specific alleles (allele13 in Guinea, allele50 in Laos, alleles 1,10,7 in Gambia).

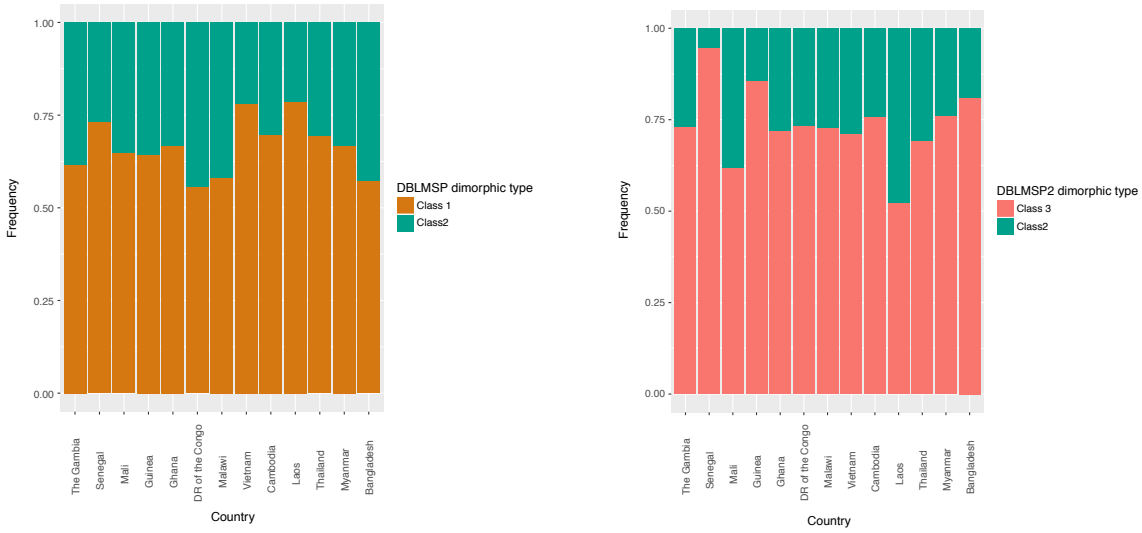


Figure 5.19: Within-country frequencies of dimorphic classes in *DBLMSP* (right) and *DBLMSP2* (left). Overall frequencies: 68% class 1 and 32% class 2 in *DBLMSP*, and 74% class 3 and 26% class 2 in *DBLMSP2*.

5.7.5.2 Linkage disequilibrium between allelic classes of *DBLMSP* and *DBLMSP2*

In population genetics, two loci are said to be in linkage disequilibrium (LD) if the frequency of their allelic combinations deviates from what would be expected if the loci were independent and associated randomly. We observe such pattern for the DBL domains of the *DBLMSP* and *DBLMSP2* genes. Overall counts for each dimorphic class combination of the two genes are presented in Table 5.5. We performed Pearson's chi-squared test of independence on this table, obtaining a p-value of $1.819e-05$, demonstrating that the allelic class present in *DBLMSP* is strongly associated with the allelic class present in *DBLMSP2*. The residuals (Table 5.6) suggest that the source of this result is an overrepresentation of class 2/class 2 combinations to the detriment of class 1/class 2 and class 2/class 3 combinations.

We confirmed this result by performing a meta-analysis of the association between the allelic class present in *DBLMSP* and *DBLMSP2* within each country. Results are shown in Fig. 5.22. We obtained a p-value of $1e-04$ and an odds ratio $OR = 2.03$,

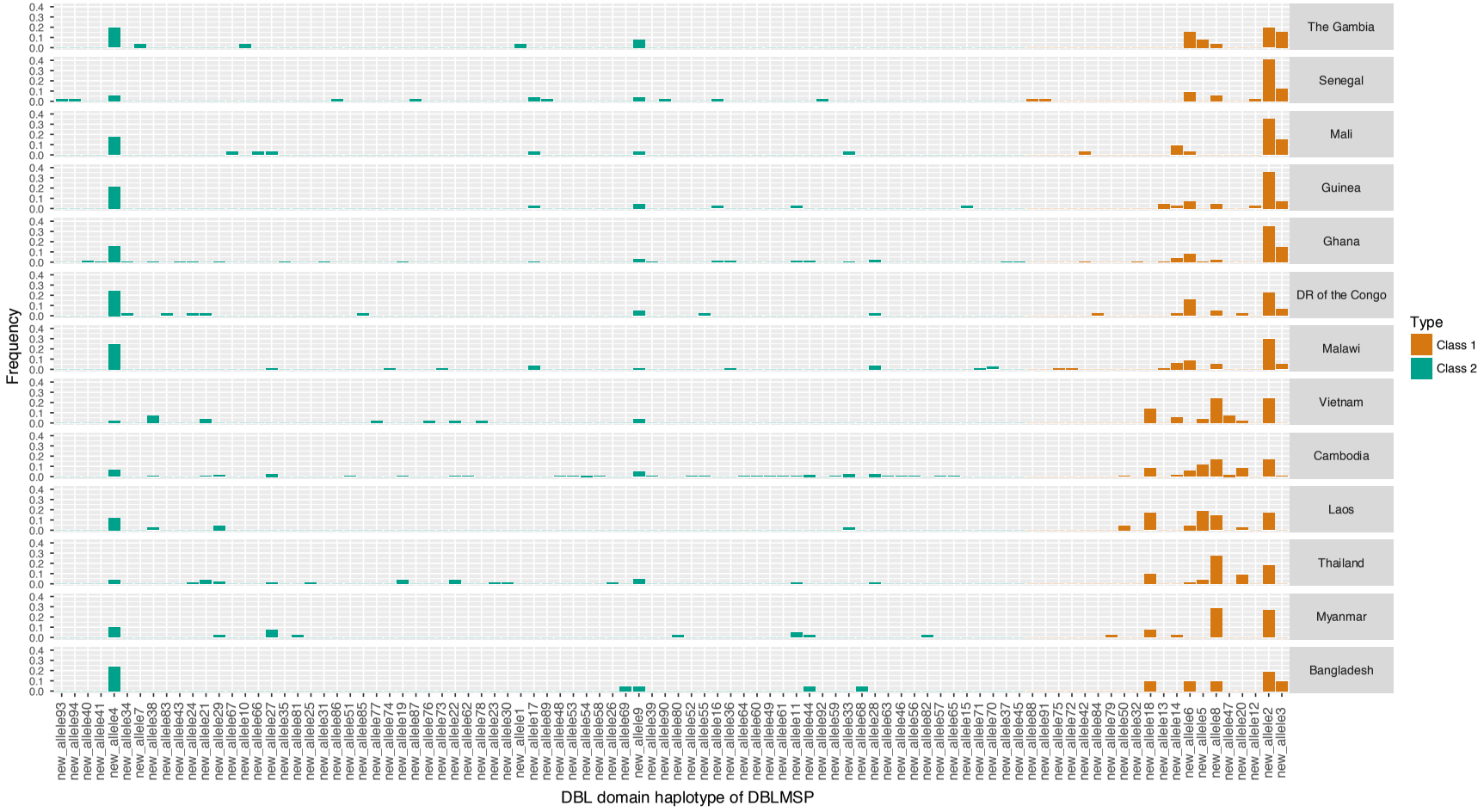


Figure 5.20: Population level frequencies of haplotypes in the dimorphic region of the DBL domain of *DBLMSP*. Alleles at the bottom are sorted in the order imposed by their phylogenetic tree.



Figure 5.21: Population level frequencies of haplotypes in the dimorphic region of the DBL domain of *DBLMSP2*. Alleles at the bottom are sorted in the order imposed by their phylogenetic tree.

DBLMSP2 / *DBLMSP*	Class 3	Class 2	Total
Class 1	577	163	740
Class 2	231	121	352
Total	808	284	1092

Table 5.5: Contingency table for dimorphic class combinations of *DBLMSP* and *DBLMSP2*. If the two loci were not associated, the number of samples containing the class 2/class 2 combination would be expected to be close to the product of individual class 2 frequencies, i.e. $352 \cdot 284/1092 = 91.5$. Instead, 121 samples containing the class 2/class 2 combination are observed.

DBLMSP2 / *DBLMSP*	Class 3	Class 2
Class 1	1.258743	-2.123165
Class 2	-1.825079	3.078423

Table 5.6: Residuals table after applying Pearson's chi-square test for association between the dimorphic class genotype from each gene, showing the class 2/class 2 combination counts deviate the most from the expectation under independence.

meaning that the odds of a class 2 allele presence in *DBLMSP2* are twice as high when the *DBLMSP* allele belongs to class 2 than when it belongs to class 1. This is somewhat surprising since loci are often kept under balancing selection because of a heterozygote fitness advantage[20], so we were expecting to see favoured combinations of alleles from different clades in the phylogenetic tree (class 1/class 3, class 1/class 2 or class 2/class 3). Parasites carrying a class 2 allele in both *DBLMSP* and *DBLMSP2* could give rise to non-allelic homologous recombination/gene conversion events because of potential misalignment of the two genes due to high sequence similarity. Nevertheless, the most frequent combination is the one consisting of gene specific alleles (class 1/class 3).

Figure 5.23 shows finer resolution LD at the allelic level (e.g. allele 4/allele 3, allele 20/allele 4, allele 8/allele 2), but with different patterns across countries (Figs. 5.24 - 5.25, A.11 - A.20). We expected to observe less LD in countries with high transmission intensities, because of recombination occurring more frequently. We observe no

[20] see Section 2.1

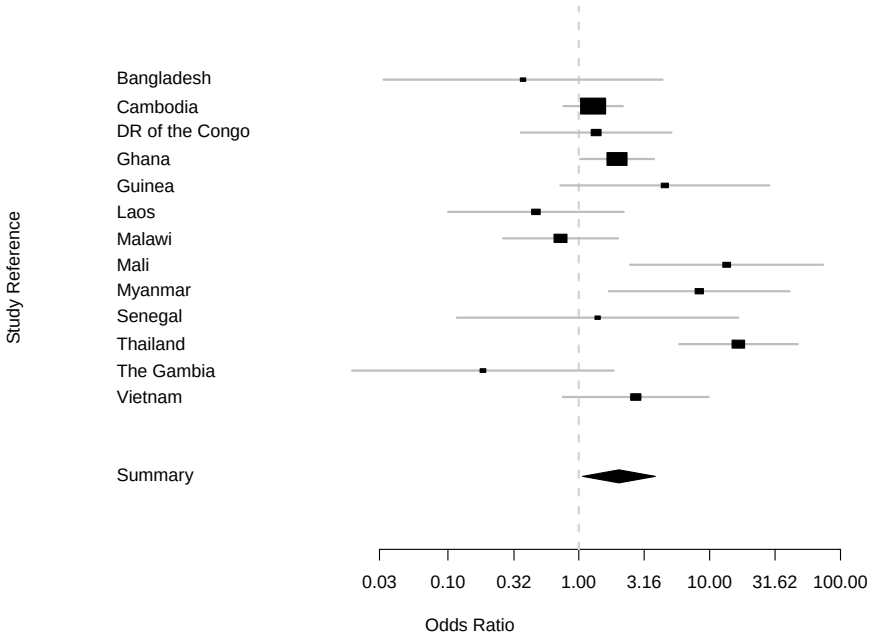


Figure 5.22: Meta-analysis of linkage disequilibrium between the dimorphic status of *DBLMSP* and *DBLMSP2* per country. $OR = 2.03$, with a 95% confidence interval $CI = (1.07, 3.84)$. The random effects model `meta.DSL` in the R package *rmeta* was used for this analysis.

such pattern. We used two measures as proxy for transmission rates within populations: estimated parasite rates in 2015 from the Malaria Atlas Project [Bhatt et al., 2015] and percentage of non-clonal infections from the Pf3k project. We tested for association with the r^2 metric of LD, but neither transmission measure was significantly correlated (p-values 0.2 and 0.6 respectively). This could indicate that natural selection is acting in the region of the *DBLMSP* genes, but we avoid to draw a conclusion because of the limited power of this analysis.

It is worth noting that antibody response to the 3D7 versions of the DBL domains of *DBLMSP* (class 1) [Chiu et al., 2015] and *DBLMSP2* (class 2) [Tetteh et al., 2013] has been correlated to lower incidence of clinical malaria. A potential explanation for the rarity of strains containing both dimorphic classes of either gene is that their antigens could be recognised by hosts exposed to either version of the dimorphism, reducing the strain's fitness.

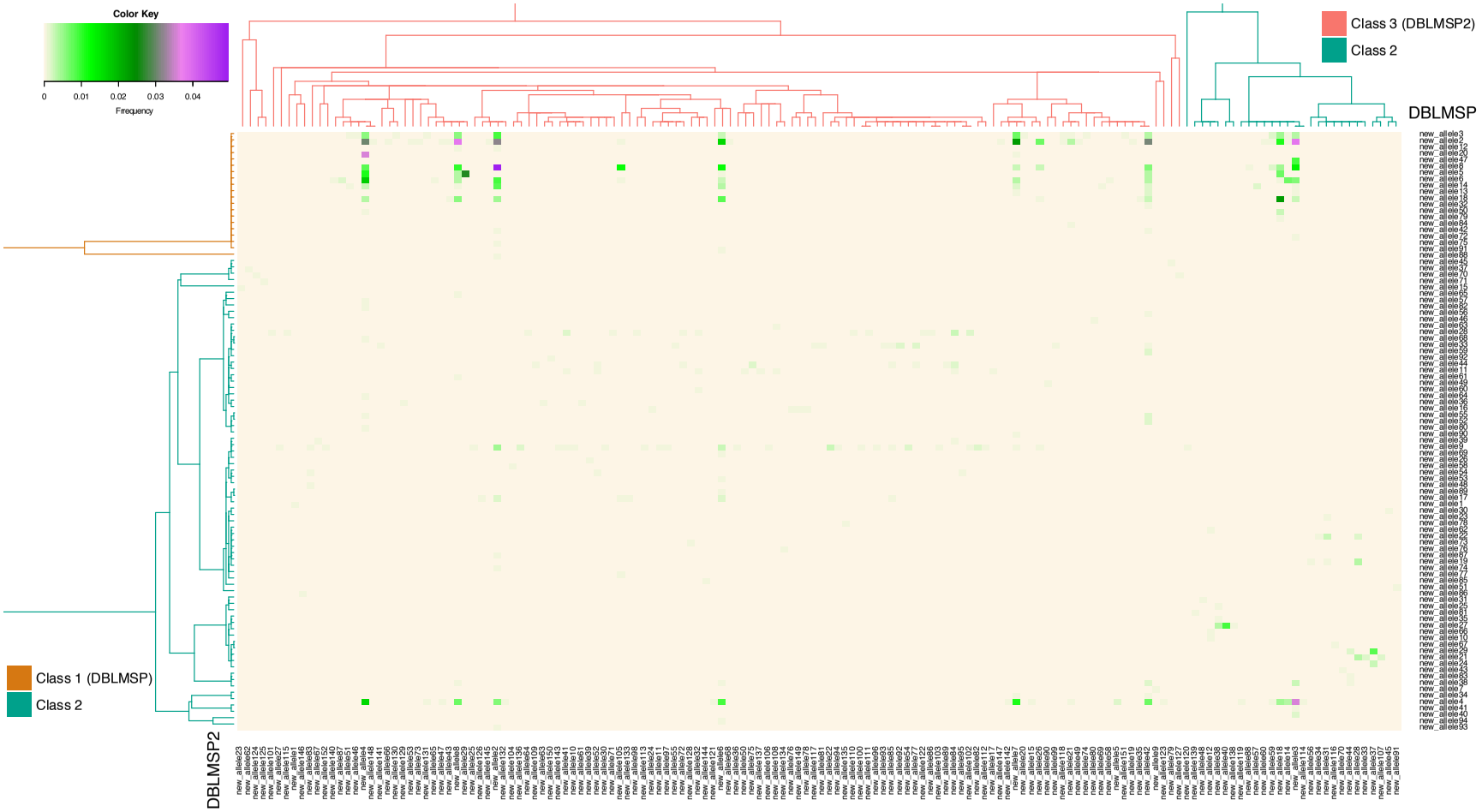


Figure 5.23: Joint frequency heatmap of alleles present in *DBLMSP* and *DBLMSP2*. Note that rows and columns are sorted by dendrograms built from the genetic distance between alleles and not from the distance calculated from their frequency patterns.

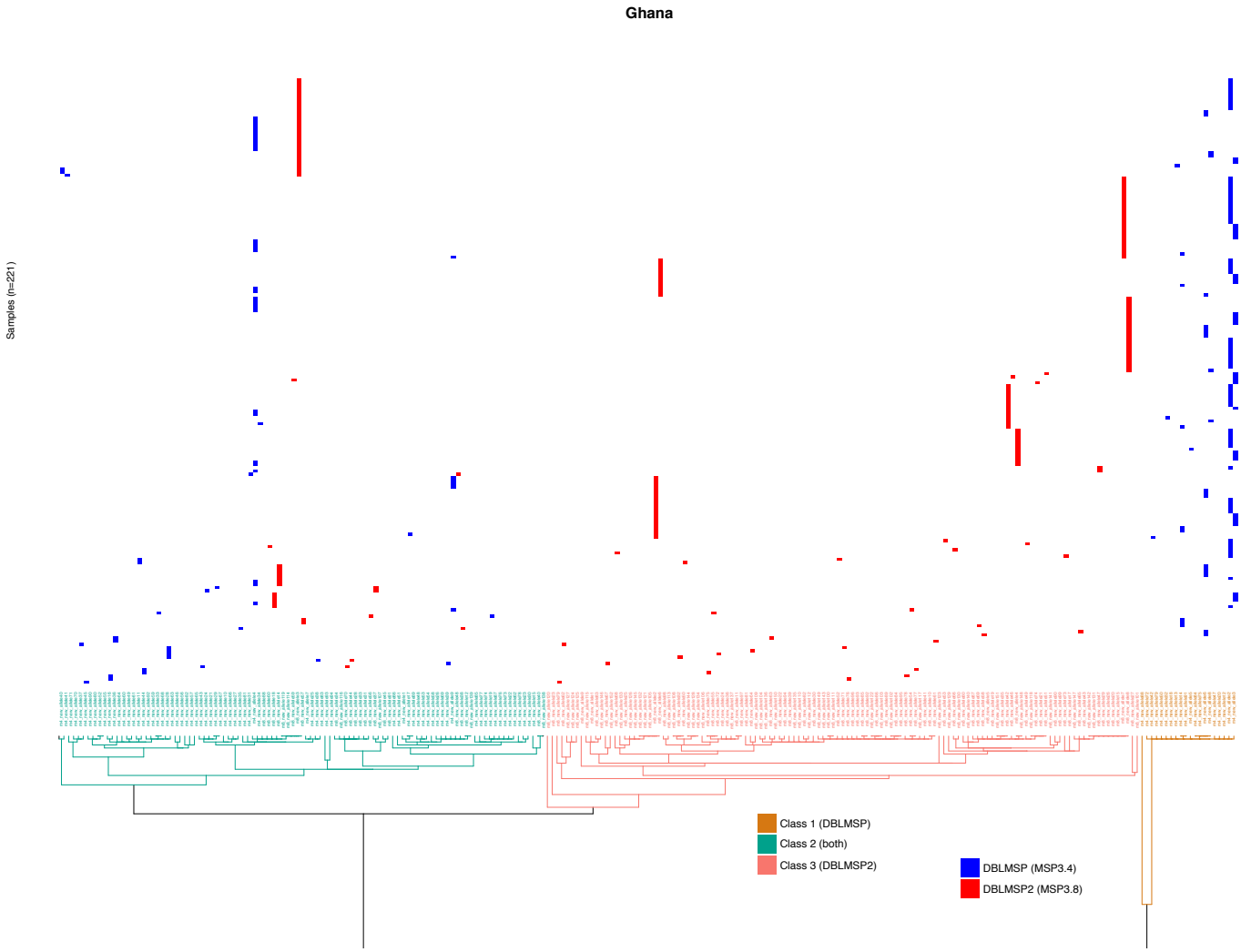


Figure 5.24: LD patterns in Ghana. Each row represents a sample, with a red dot in the column corresponding to its *DBLMSP* allele and a blue dot in the column corresponding to its *DBLMSP2* allele. Samples are sorted according to genotype similarity, whereas alleles are sorted based on their distance in the phylogenetic tree.

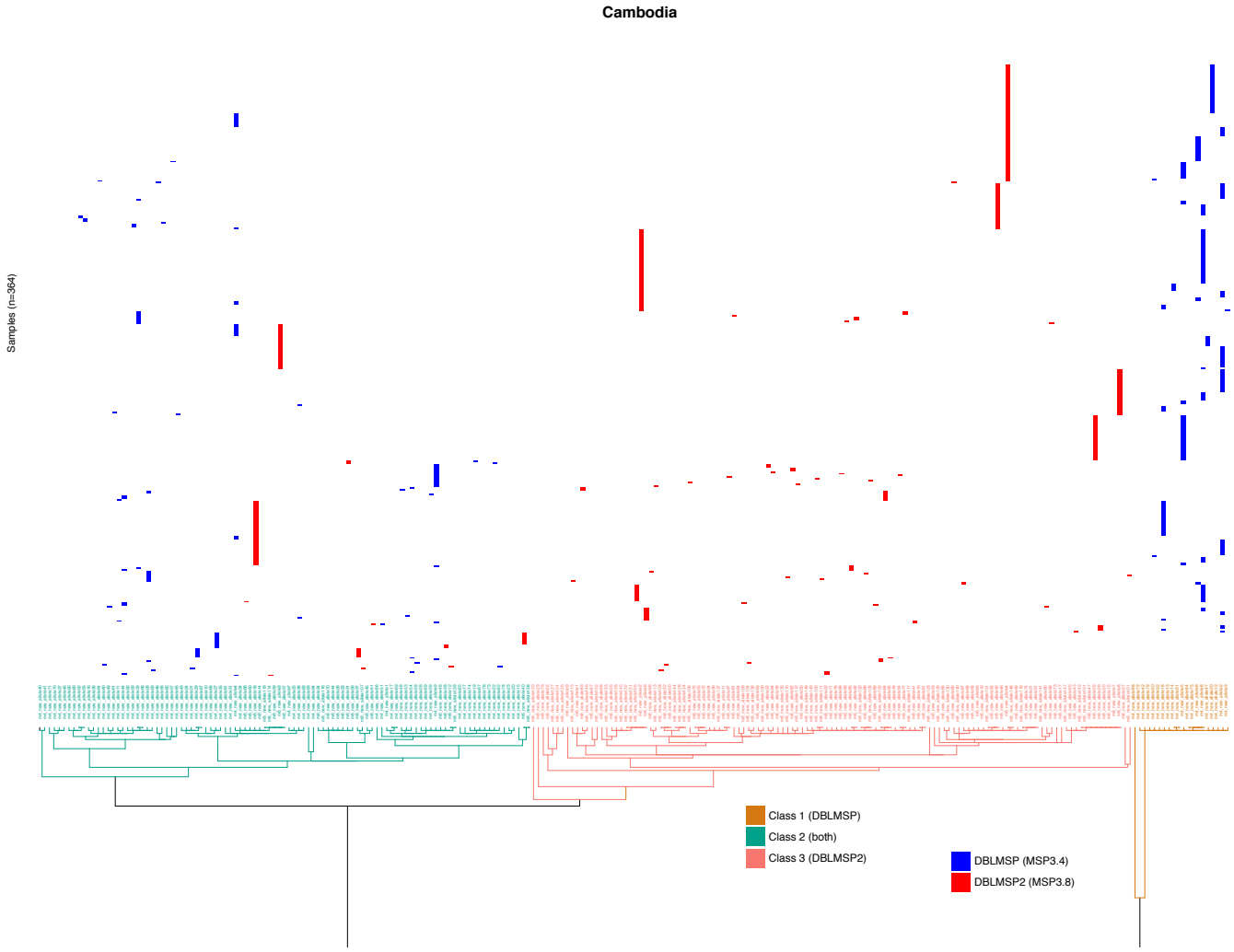


Figure 5.25: LD patterns in Cambodia. Each row represents a sample, with a red dot in the column corresponding to its *DBLMSP* allele and a blue dot in the column corresponding to its *DBLMSP2* allele. Samples are sorted according to genotype similarity, whereas alleles are sorted based on their distance in the phylogenetic tree.

5.8 Summary of results

We have introduced a method to build a population reference graph resource from publicly available whole-genome sequencing data generated by the Pf3k project. This resource represents known variants discovered in two hyper-polymorphic genes, *DBLMSP* and *DBLMSP2*, encoding surface antigens in the blood stage version of the malaria parasite *Plasmodium falciparum*. We have shown that using this reference graph as prior information, together with `gramtools`, improves the sequence inference of these two genes compared to standard methods relying on the 3D7 reference genome. The *DBLMSP* sequence was fully resolved in 1233 samples, increasing from 820 samples with 3D7. The *DBLMSP2* sequence was fully resolved in 1187 samples, increasing from 209 samples with 3D7. This analysis has given insight into variation previously inaccessible and could be extended to other genes encoding merozoite surface proteins to enable the creation of detailed catalogues of antigenic variation. Such catalogues would open up the possibility of higher resolution population genetic studies in order to understand how these genes evolve under immune pressure and identify prospective vaccine candidates.

We performed a preliminary population genetic analysis for *DBLMSP* and *DBLMSP2* based on the `gramtools` results. We have built a global diversity map for the DBL domains of the two genes, which revealed their allelic dimorphism is preserved at common frequencies within 14 countries around the world, confirming they are under balancing selection. Interestingly, we found extensive similarity between one of the dimorphic forms in each gene (91% on average) and *DBLMSP* alleles of this form are more similar to *DBLMSP2* alleles of both forms than to the rest of *DBLMSP* alleles. This suggests that *DBLMSP* has partially evolved as a paralog of *DBLMSP2*, with gene conversion between the two genes. However, it does not give any clues on the origin of the other *DBLMSP* form.

We have detected linkage disequilibrium between the dimorphic status of the two

genes, which is, to our knowledge, the first significant association between two dimorphic loci [Pearce et al., 2004]. Normally, this could indicate selection on some fitness advantage given by the interaction between the two genes, but it is not clear how non-allelic gene conversion affects the process. Gene conversion is also a confounding factor for mutation rate calculations [Nielsen et al., 2003] and selection tests [Casola and Hahn, 2009] that should be taken into account in future analyses.

Finally, there is a fraction of samples (172/1316) for which `gramtools` cannot infer the full *DBLMSP*/*DBLMSP2* sequence. This indicates that the genetic variation in this subset of samples is substantially different from the rest and cannot be discovered from short-read sequencing data with available methods. Hence, these samples should be sequenced with long-read technologies like PacBio in order to complete the catalogue of variants for these genes. More generally, `gramtools` can be used to cheaply detect samples with complex novel variants and prioritise them for long-read sequencing.

5.9 Discussion

The results presented here provide a detailed characterisation of the genetic variation exhibited by the two *DBLMSP* genes and outline a roadmap for studying other antigenic loci in *Plasmodium falciparum*. This opens up the possibility of creating a comprehensive map of antigenic diversity, which would enable the detection of selection signatures and new loci mediating interactions with the host immune system and virulence. However, some loci will be more difficult than others, as there is substantial variability in the complexity of sequence variation encountered.

At the low end of the spectrum, there is the EBA-175 gene, whose DBL domain comprises a simple dimorphism in the form of a ∼400 bp indel [Ware et al., 1993]. All studied versions of the gene contain either one or the other dymorphic

types, with negligible within-type diversity. The two main alleles appear to be derived from a single EBA-175 version carrying both lineages before the *Plasmodium reichenowi*/*Plasmodium falciparum* split [Yasukochi et al., 2017]. The function of the EBA-175 molecule is to bind the glycophorin A protein, found on the membrane of human red blood cells, initiating the merozoite invasion [Orlandi et al., 1992]. The role of the dimorphism in this interaction is unclear, but is presumed to be related to a dimorphic feature of the red blood cells receptors.

At the opposite end of the spectrum, there is the *var* gene family, which consists of 60 hypervariable genes, distributed across the 14 chromosomes, mostly in the subtelomeres. These genes undergo frequent non-allelic homologous recombination despite their low levels of sequence identity, generating extreme diversity in a structured manner, that preserves domain architecture. It has been argued that most of these recombination events occur during the asexual stage of the parasites, allowing continuous generation of new mosaic *var* gene sequences over the course of an infection [Claessens et al., 2014]. Such antigenic variation is a mechanism for immune evasion, since the parasite switches expression between the *var* genes – only one gene is expressed at a single time, the rest being transcriptionally silent. The resulting variant molecule is displayed on the surface of an infected red blood cell, which decreases immune recognition and clearance. This enables the parasite to hide and proliferate inside the cell, leading to severe disease [Scherf et al., 2008].

The *DBLMSP* loci lie somewhere in between the two extremes. Despite being highly polymorphic, their diversity is structured in only two clear allelic classes, as opposed to the nearly limitless multimorphism of the *var* genes. On the other hand, some similar features can be observed, such as the occurrence of sequence exchange between distinct members of the gene family and the presence of a DBL domain in both *var* [Kraemer and Smith, 2006] and *DBLMSP* genes [Singh et al., 2009]. Moreover, there is some evidence that *DBLMSP2* is controlled by a similar epigenetic

mechanism to that of the *var*s. Variant expression has been reported for *DBLMSP2* among different time points or parasite clones, a characteristic of subtelomeric genes [Amambua-Ngwa et al., 2012]. *DBLMSP2* is also one of the very few internal chromosome loci that is enriched for a histone mark of heterochromatin (H3K9me3) that is associated with differentially transcribed subtelomeric gene families, with silent members presenting high levels of H3K9me3 [Lopez-Rubio et al., 2009]. These facts point to the possibility of some functional redundancy among the diversity of *DBLMSP2* alleles, but it remains unclear how the dimorphic structure arose and why it is maintained in multiple populations. Our method enables access to genetic variants present in antigenic loci at a higher resolution than ever before and is a first step into shedding light on the selective pressures shaping their patterns of polymorphism.

Chapter 6

Final discussion

6.1 Contributions

Extending the traditional reference genome to a variation-aware reference structure has recently been proposed in order to overcome weaknesses of both read mapping and *de novo* assembly methods in characterising genetic variants in high-diversity regions. In this thesis, we developed a novel, low-memory data structure that integrates prior variants into the reference sequence and associated algorithms to perform genome inference, all implemented in a software called `gramtools` . This focus on evaluation at the level of the inferred genome stands in contrast to most other graph genome methods that focus purely on read mapping.

Our `gramtools` implementation is readily usable for megabase-sized genomes, being able to process a typical *Plasmodium falciparum* sample against a whole-genome PRG in under 6 hours single-threaded, while requiring a minimal amount of memory. We applied it to a large scale *Plasmodium falciparum* data set in order to study two unusually diverse, antigenic genes in detail. `gramtools` successfully inferred their sequence in over 1000 samples, because it was able to genotype alleles that were highly diverged from the reference sequence and therefore missed by standard meth-

ods. Moreover, it led to the discovery of novel alleles caused by a structural event that was previously unknown, despite not being designed to handle structural variants. Overall, we showed that `gramtools` can lead to substantially increased variant accessibility in hyper-diverse genomic regions compared to usual methods relying on a single reference genome. This pilot study sets up a framework for analysing genetic variants from other antigenic loci and could be a first step towards the construction of detailed catalogues of antigenic sequence in *Plasmodium falciparum*, similar to the human HLA databases. Such extensive documentation of genetic variability could be the basis of an "antigen-typing" system, analogous to HLA-typing, that classifies alleles based on the similarity of functional properties they induce.

While initially work on variation-aware references focused on how to best represent diversity or how to align reads across alternative sequences, more recent articles have shown the practical value of such data structures in the final goal of variant calling [Dilthey et al., 2015], [Valenzuela et al., 2015], [Novak et al., 2017b], [Eggertsson et al., 2017]. Our work further confirms that using extended references leads to improved variant calling. One key distinction is that our findings are in the context of pathogen genomics, whereas previous work has mainly been carried out in the context of human genomics. Our method cannot be applied to whole-genome human data yet, so a direct comparison could not be performed, due to limitations in computational performance, specifically execution time. Additional work is required to implement the speed optimisations suggested at the end of Chapter 4, but initial measurements look promising for improving runtimes considerably, as we will discuss in the next section. We believe this avenue is worth exploring due to the appeal of the low memory required by `gramtools` to encode a large number of human genomes[1] – only 27 GB to build a population reference graph index of the 1000 Genomes Project variant set.

[1]See Section 4.3.1.2

6.2 Limitations

While being able to provide additional insights into the extent of diversity in highly polymorphic genomic regions compared to traditional methods, the work presented in this thesis has 4 main limitations: computational speed of read quasimapping, lack of support for quasimapping with mismatches, lack of representation of structural variants and lack of multiploid inference.

As we showed in Chapter 4, some computational burden can be shifted from read quasimapping to k-mer precalculation. However, this can result in the k-mer indexing step prior to quasimapping being too slow in the case of large PRGs with many k-mers such as the ones representing whole-genome human variation. One of the optimisations discussed at the end of Chapter 4, which involves reusing index information from substrings shared between k-mers , has already been implemented by Robyn Ffrancon and has provided a speed-up of k-mer indexing by a factor of 50,000 for the whole-genome *Plasmodium falciparum* PRG. Although it has not been applied to the human PRG yet, we remain optimistic that this optimisation represents an important step in making `gramtools` scalable to large genomes.

Currently `gramtools` performs inference only based on reads that match the PRG perfectly. This implies that some coverage is lost due to sequencing errors, places a burden on the PRG to represent a large extent of the existing variation and confers little robustness to potential errors in the PRG construction. These issues did not affect considerably our study in Chapter 5 due to 2 factors: most samples analysed were Illumina sequenced at high depth of coverage and low error rate, and the PRG was constructed from a comprehensive variant panel obtained through 3 independent pipelines from a large set of samples belonging to different populations.

The way `gramtools` is set up allows extension to quasimap reads with mismatches and algorithmic ideas for doing this were outlined in Chapter 4. Implementing these ideas was not our main focus initially due to the fact that allowing mismatches in

the quasimapping step comes at the cost of some computational speed arising from the increased search space. However, from our latest measurements (see previous paragraph), the quasimapping speed bottleneck seems to be removed, so `gramtools` improvement efforts should be directed towards supporting quasimapping tolerant of mismatches. Once that goal is achieved and the loss of coverage is minimised, `gramtools` would have increased power to infer the genomes of minor strains in mixed infections, after adding the following functionality. The `--infer` step should be modified to support the inference of the most likely multiploid path in the PRG, such that each strain in a mixed sample has its own personalised reference. This could be done via an HMM based on coverages at junctions in the PRG, similar to the one described in [Dilthey et al., 2015].

The lack of support for structural variants was again somewhat motivated by the initial low quasimapping speed. Structural events tend to affect large genomic regions that usually contain other variants, so representing them is likely to increase the complexity of the PRG and hence, the quasimapping time. As discussed at the end of Chapter 3, it is straightforward to extend `gramtools` to represent nested alleles in the PRG at the cost of some computation time. This would be a solution for large deletions and insertions, however, the acyclic structure of the PRG impedes the natural representation of more complex events like inversions and translocations.

6.3 Future applications for `gramtools`

As argued in the introduction of this thesis, methods based on generalised reference genomes are most valuable in regions of high sequence diversity. However, the technical features of `gramtools` make it more suitable to particular species with certain genomic features, such as genome size, ploidy and frequent types of diversity.

As it stands, `gramtools` can be used to analyse whole-genome variation from

Plasmodium falciparum in unmixed samples and we expect it to improve the characterisation of indels, which are overly abundant in this species. It is also likely to perform well in diverse regions of the core genome that share characteristics with the *DBLMSP* genes, such as other genes of interest that express antigens on the surface of the merozoite: *MSP1*, *MSP3*, *MSP6* and the *SURFIN* family. It is less suitable to hyper-variable gene families located in the subtelomeres, such as *var*, *rifin* and *stevor*, expressing antigens on the infected red blood cells, due to accelerated non-allelic homologous recombination.

After the addition of multiploid inference, `gramtools` analyses of mixed infections in *Plasmodium falciparum* could give insights into genetic variants that promote parasite survival and proliferation in the host bloodstream. Diploid inference is also essential for analysing human genome variation, with *MHC* being the first point of interest. `gramtools` could perform well in this region once it incorporates nested alleles to accommodate for large deletions that are characteristic of this region.

It is tempting to apply `gramtools` to bacterial species. Their genomes are 1-2 orders of magnitude smaller than eukariotes, so computational performance is not an issue. Furthermore, for many species, complete assemblies are publicly available, which could readily be used as input for a PRG. However, different challenges arise due to transposable elements and horizontal transfer, which cause the gene order to be shuffled between individuals. It is difficult to encode such information in our linearised PRG structure, though one could think of building small PRGs at the gene level and then having a genome-level network that represents the different gene orderings. *Mycobacterium tuberculosis* is considered an atypical bacterial species that is unusually clonal, resistant to horizontal gene transfer and harbouring limited transpositions. It contains a large, hyper-variable gene family, *PE/PPE*, that is thought to be a source of antigenic variation and immune evasion, so this species could be a starting application target for the present version of `gramtools` in bacteria.

Finally, it is worth thinking about using `gramtools` for inferring viral genomes in the future, but it is essential that quasimapping reads with mismatches is supported first. The main challenge with viruses is the high mutation rate, that is up to 5-6 orders of magnitude higher than in bacteria and eukaryotes. As a consequence, they undergo rapid evolution within the host, such that any given infection contains a cloud of competing mutant strains, called a *quasispecies*. As there can be variability in drug response and virulence among strains, the need to construct the genomes of single strains arises. This is a distinct setting from the *Plasmodium falciparum* mixed infections, where different strains originate from different infections, because component genomes are much more closely related. However, it is more affordable to produce deep sequencing data on viruses due to their small, kilobase-sized genomes, which implies coverage on a larger number of strains. To handle this scenario, `gramtools` would need to have a built in system of choosing the optimal ploidy for the personalised reference, such that the main clades of the quasispecies phylogenetic tree are captured.

In the long term, as third generation sequencing technologies improve, complete genome inference is expected to be less of a challenge – long reads should overcome the limitations imposed by repetitive sequences and chromosomal rearrangements. However, we think the technical challenge of reconstructing phased sequences in samples containing multiple genomes will remain. This is of relevance in pathogen genomics, where samples often contain multiple strains, coming either from distinct infections, or from very fast evolution within the host (as exemplified earlier in malaria parasites and viruses). Characterising the relatedness between these strains and their frequencies can be informative for understanding within-host competition, drug response and immune escape. Sequence graphs are the most natural representation of the relationships between genomes and we believe the work presented here is well set up for further research in this direction.

6.4 Challenges and future directions for reference genome graphs

One major issue that remains in the area of reference graphs is that the initial step of constructing the data structure is not fully automatable. The extent of sequence variation captured in the reference graph is crucial to the accuracy of genome inference, but compromises must be made in the interest of computational performance. Currently, an automatic framework for deciding how much variation and in what form is optimal to include in the reference graph does not exist. A fully general graph collapses any shared nucleotide between any subset of samples from the reference panel and contains a path representing any sequence combination of known variants. In theory, such graph would maximise the power to infer any genome that is related to the genomes in the reference samples. But the structural complexity of such graph makes it unlikely to be computationally efficient in encoding whole-genomes from large reference cohorts and can hinder the inference procedure due to a large number of possible candidate alignments. An obvious way to limit the graph search space is to restrict to paths that are consistent with biological knowledge. For instance, one could incorporate LD information and eliminate paths that are not consistent with previously observed haplotypes. Haplotype frequency information can also assist in ranking alternative read mappings with similar alignment scores. Some work based on the Burrows-Wheeler Transform has already been done in this direction [Novak et al., 2017a] and we think it is worth exploring further.

Another uncertainty around the construction of reference graphs is that the same multi-nucleotide sequence deviation from a reference can be represented in multiple equivalent forms, as one or multiple variants (refer back to Fig. 2.1 and Fig. 4.6 for example), contingent on what scale is chosen to define variants. This ambiguity is reflected in the VCF file format itself and essentially stems from the lack of an

atomic unit of measure for the difference between two sequences. Despite implying the same sequence, different representations of complex variants result in different graph structures, with different numbers of nodes and edges, which impacts performance. But the choice of variant representation should balance computational efficiency with interpretability. Ideally, the variants called are in a format that is suggestive of the biological mechanism from which they arose, but that is only possible if they are initially represented in the graph in a way that reflects this mechanism. Questions regarding the definition of variant sites and alleles in a sequence graph are just starting to be explored. Paten et al. have proposed algorithms for decomposing graphs into motifs in the form of subgraphs with certain path structures, that are created by particular types of complex variation, such as nested variants and adjacent variants [Paten et al., 2017b], [Rosen et al., 2017]. They argue that these motifs could provide a natural way of defining graph sites for this type of variants, but their benefit for performance and inference remains to be evaluated.

From a more practical standpoint, the poor understanding of the trade-offs regarding variant representations in the reference graph result in available tools making different choices when building the data structure, ending up with distinct graph topologies that are difficult to compare. This can be overcome by comparing at the final level of genome inference, but it is still essential to know how the graph structure affects the generalisability of the results to different applications. There is a need for evaluations that directly assess the impact of input parameters controlling what is encoded in the graph on performance and inference. A user needs to understand what they are representing in the graph and how that influences the results without being required to grasp the technical details of the data structure. Only with publicly available resources enabling the user to gain such knowledge, one can hope to make reference graph-based pipelines widely adopted by the community.

Appendix A

Supplementary figures

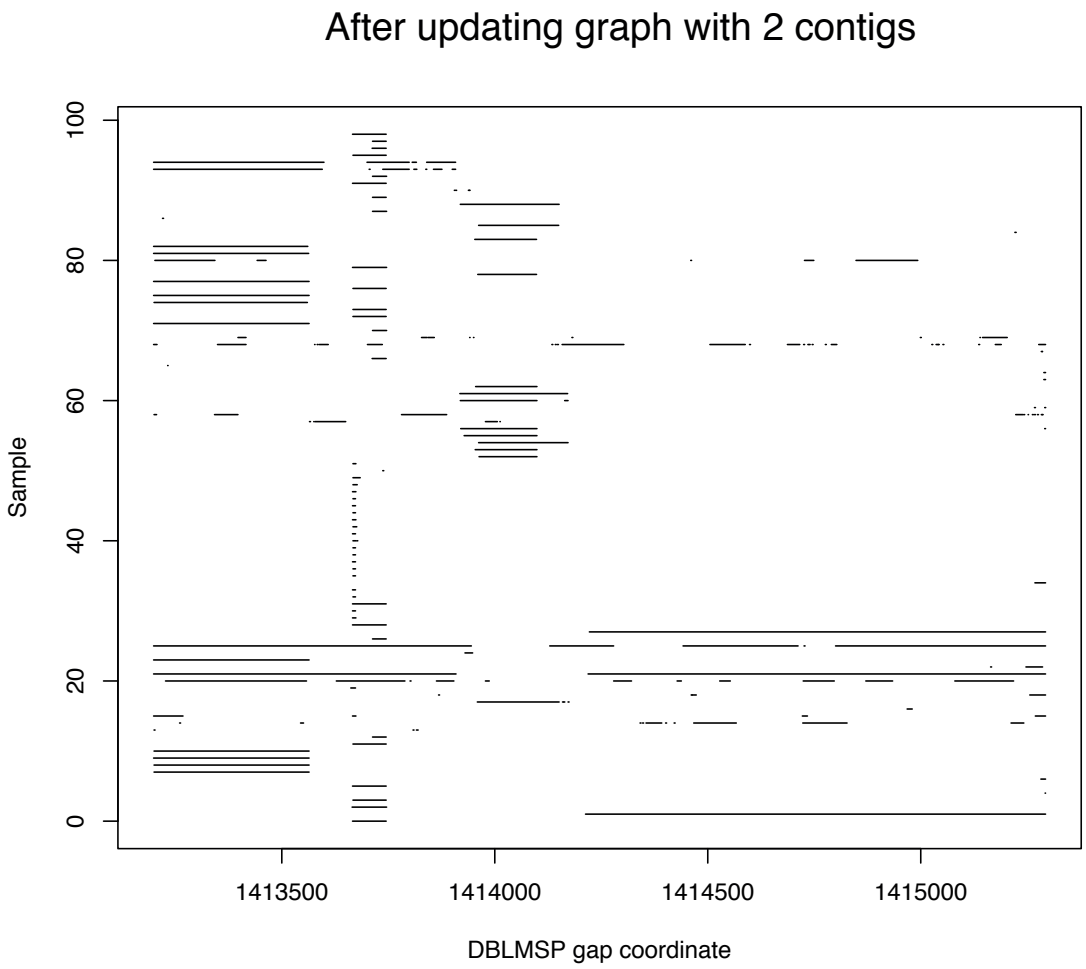


Figure A.1: Genomic coordinates of *DBLMSP* regions without read support when mapping to the personalised references in samples where the gene's sequence cannot be fully resolved. The inference was performed on the reference graph updated with 2 novel DBL domain haplotypes discovered from `Cortex` contigs, described in Sections 5.4.2.2 and B.4. The consistent gap in Fig. 5.10, produced by inference from the reference graph without these 2 contigs, is eliminated.

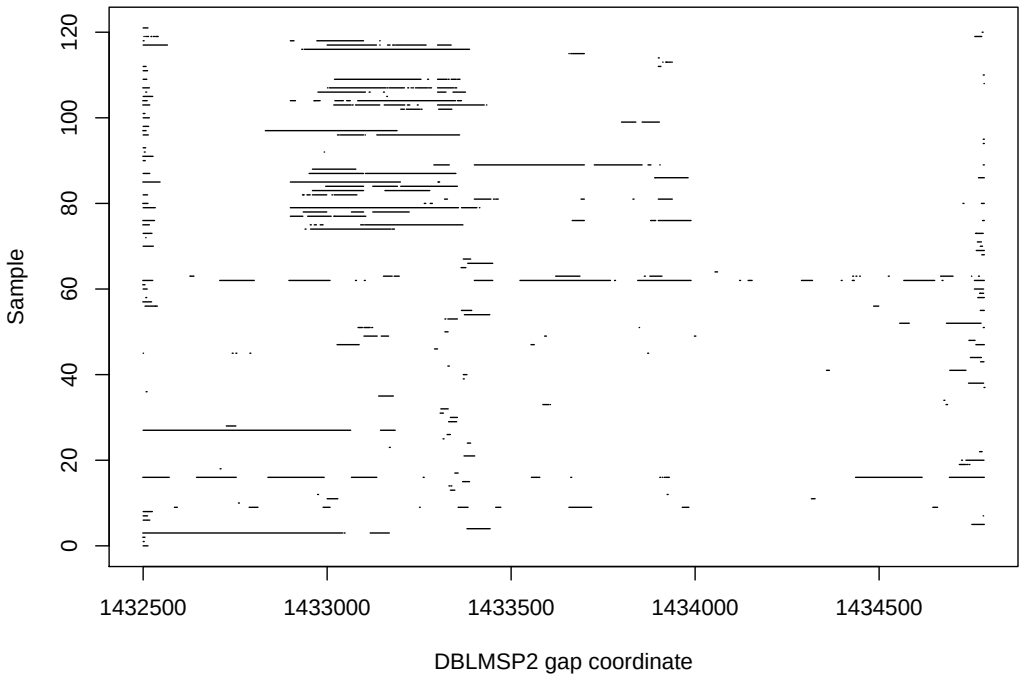


Figure A.2: Genomic coordinates of *DBLMSP2* regions without read support when mapping to the personalised references in samples where the gene's sequence cannot be fully resolved. The inference was performed on the joint *DBLMSP* and *DBLMSP2* reference graph.

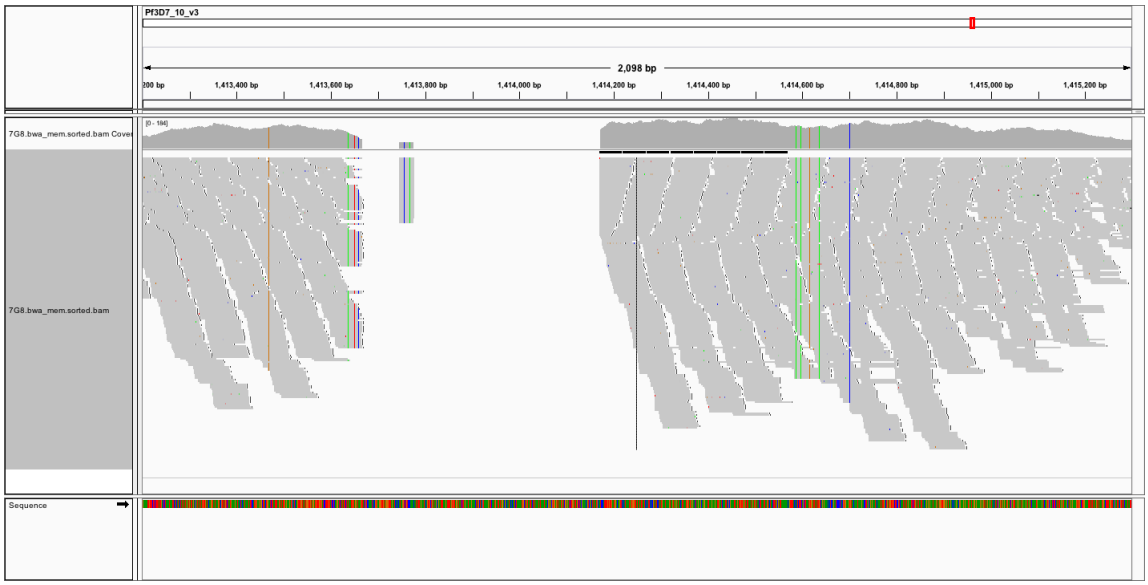


Figure A.3: IGV screenshot of read pileup in the *DBLMSP* region, with respect to the 3D7 reference for validation sample 7G8. The pileup has a gap of 476 nucleotides in the DBL domain and additional clustered SNPs.

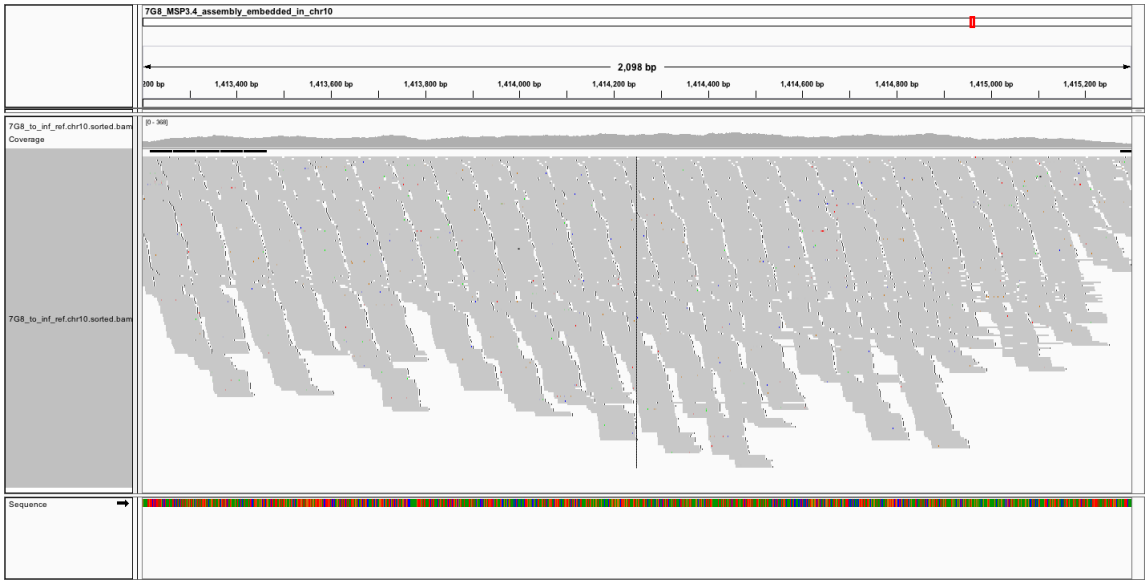


Figure A.4: IGV screenshot of read pileup in the *DBLMSP* region, with respect to the personalised reference for validation sample 7G8. The pileup shows even coverage and 9/9 SNPs removed compared to the figure above.

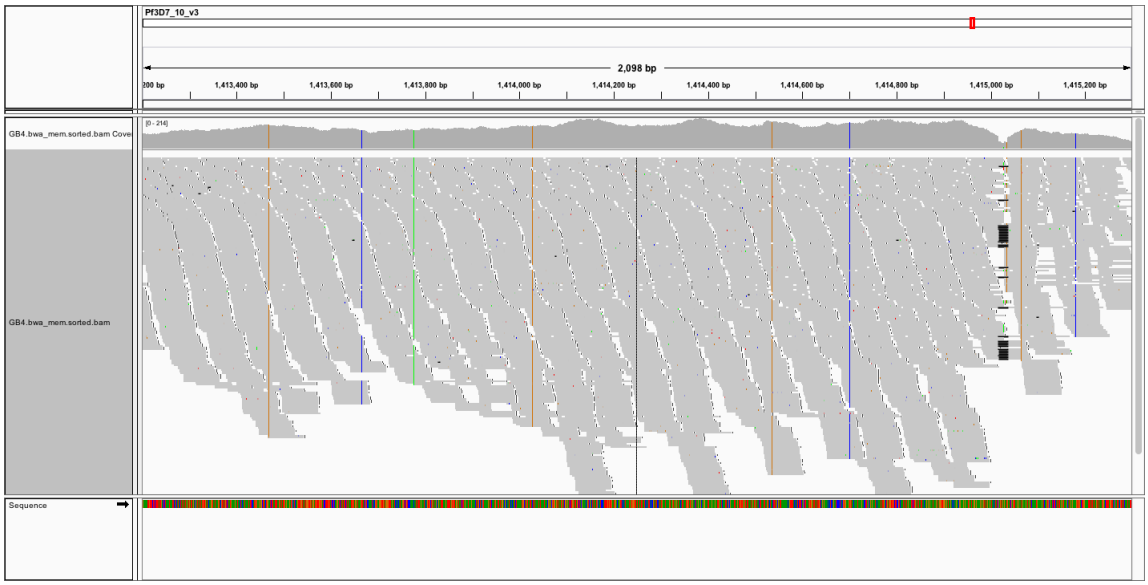


Figure A.5: IGV screenshot of read pileup in the *DBLMSP* region, with respect to the 3D7 reference for validation sample GB4. The pileup presents a dip in coverage between coordinates 1415015-1415030.

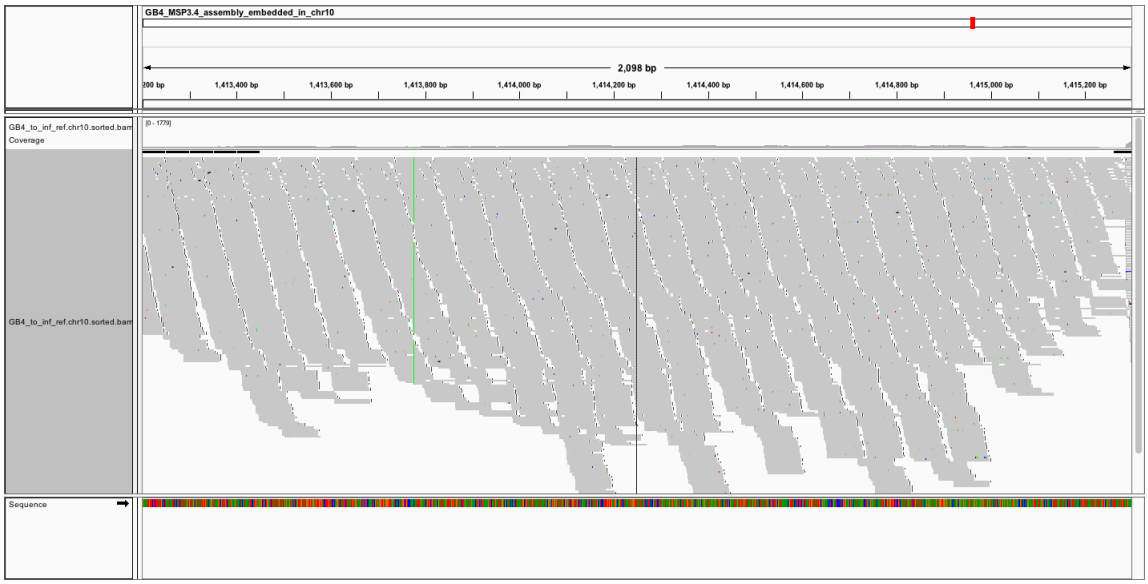


Figure A.6: IGV screenshot of read pileup in the *DBLMSP* region, with respect to the personalised reference for validation sample GB4. The dip in coverage in the figure above gets smoothed out and 7/8 SNPs are removed.

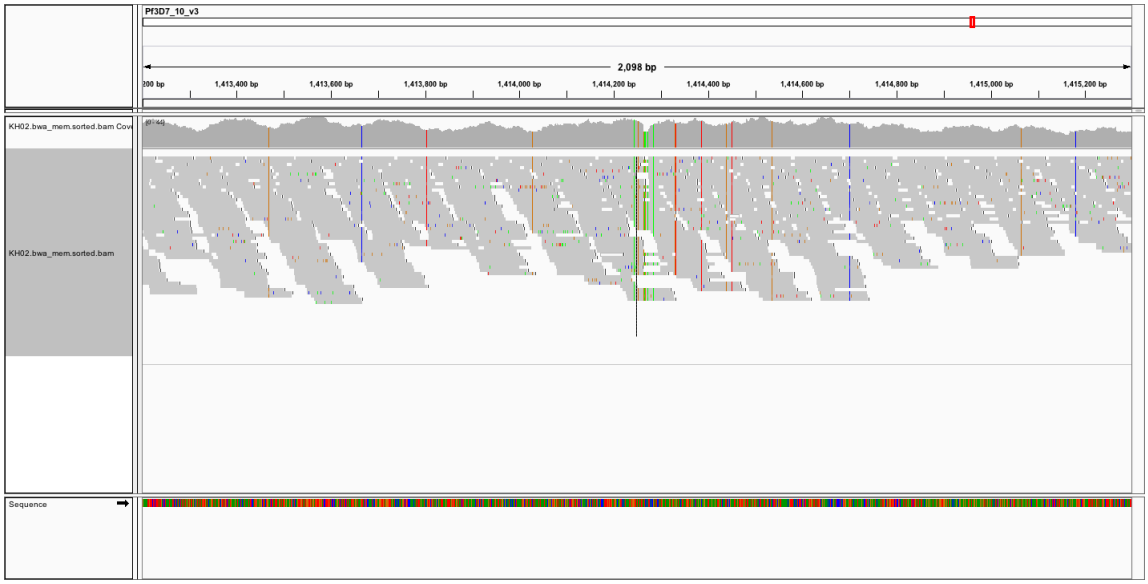


Figure A.7: IGV screenshot of read pileup in the *DBLMSP* region, with respect to the 3D7 reference for validation sample KH02. The pileup does not show any gaps, but shows clustered SNPs which could produce alignment problems.

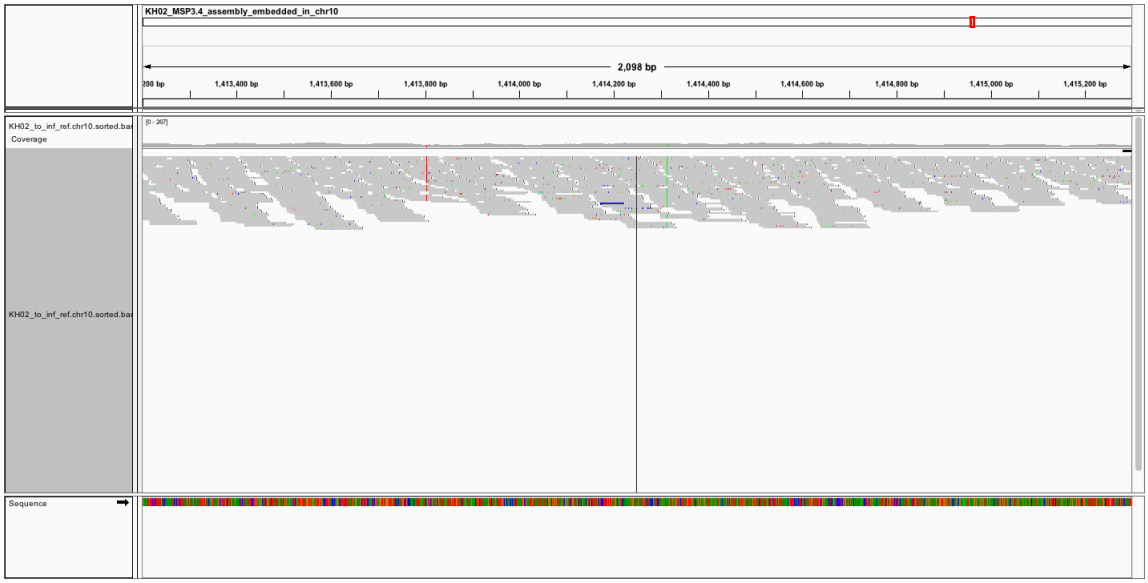


Figure A.8: IGV screenshot of read pileup in the *DBLMSP* region, with respect to the personalised reference for validation sample KH02. The pileup shows 18/20 SNPs removed from the figure above.

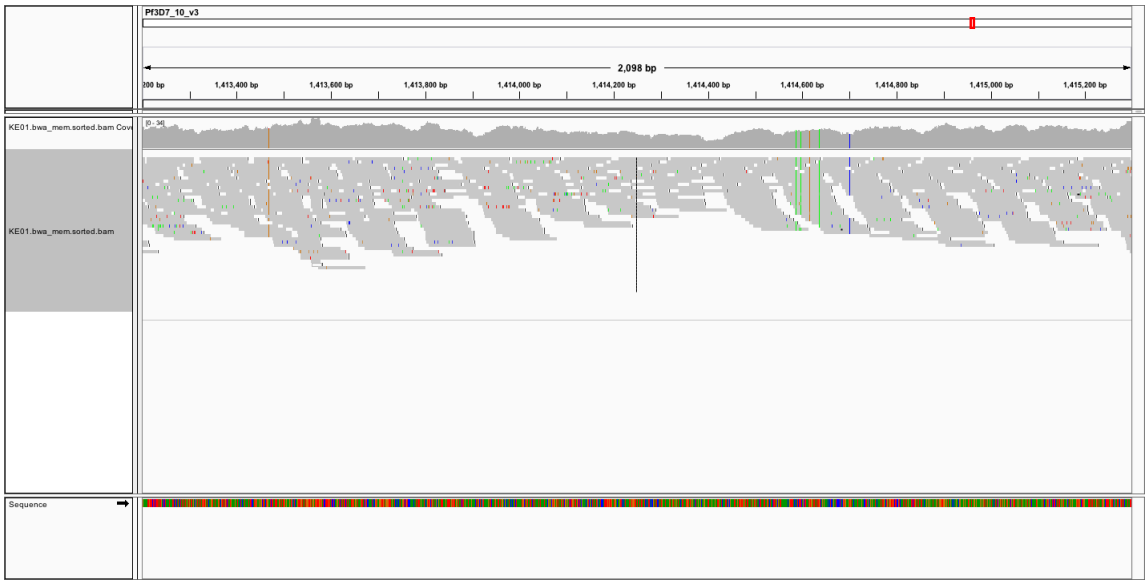


Figure A.9: IGV screenshot of read pileup in the *DBLMSP* region, with respect to the 3D7 reference for validation sample KE01.

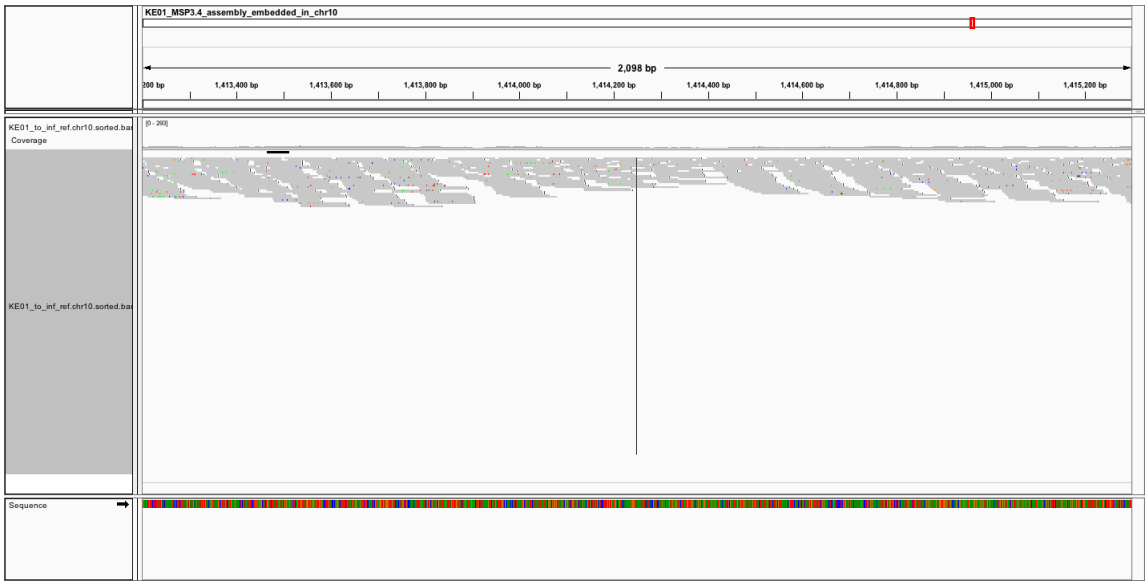


Figure A.10: IGV screenshot of read pileup in the *DBLMSP* region, with respect to the personalised reference for validation sample KE01. The pileup shows 6/6 SNPs removed from the figure above.

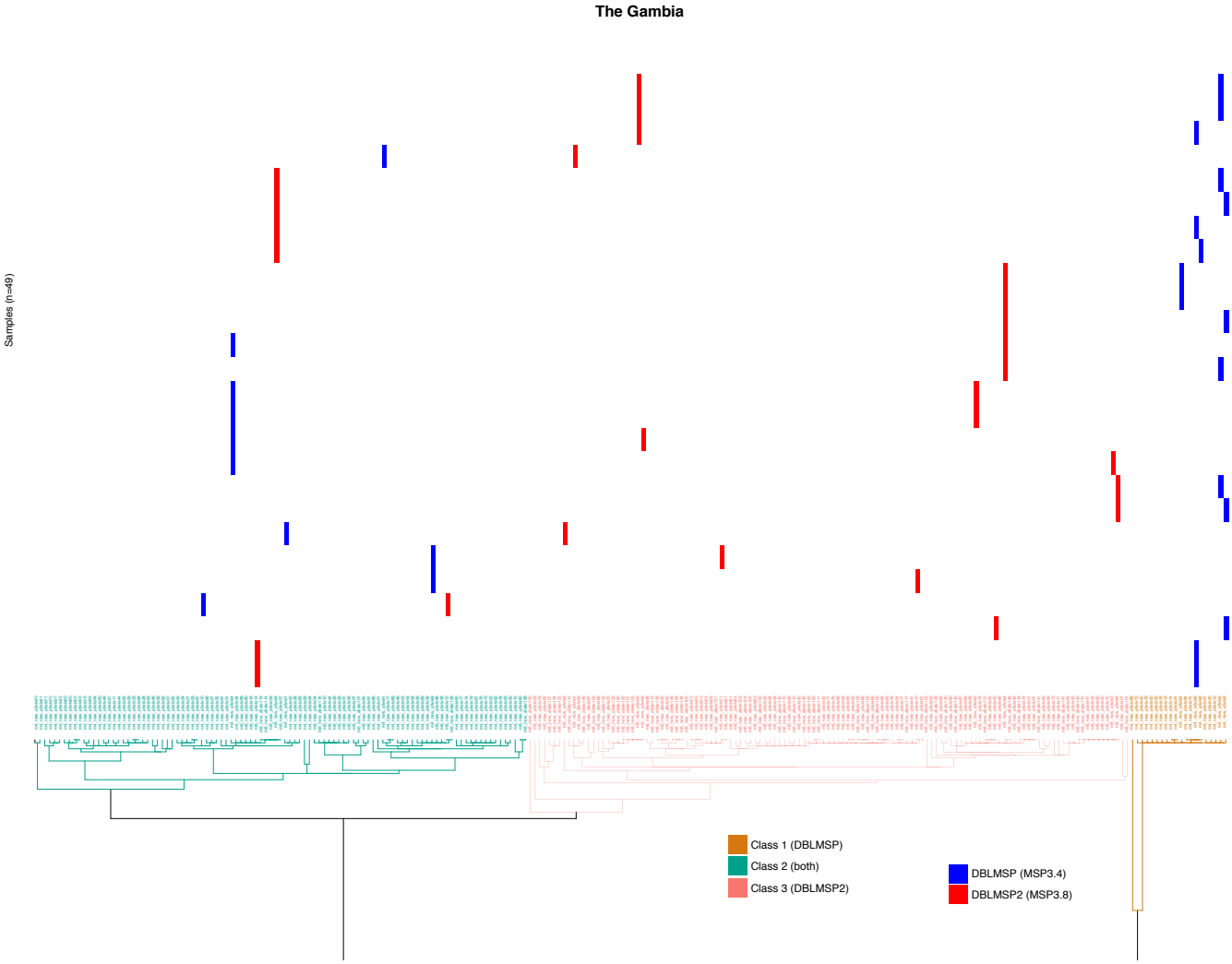


Figure A.11: LD patterns in Gambia. Each row represents a sample, with a red dot in the column corresponding to its DBLMSP allele and a blue dot in the column corresponding to its DBLMSP2 allele. Samples are sorted according to genotype similarity, whereas alleles are sorted based on their distance in the phylogenetic tree.

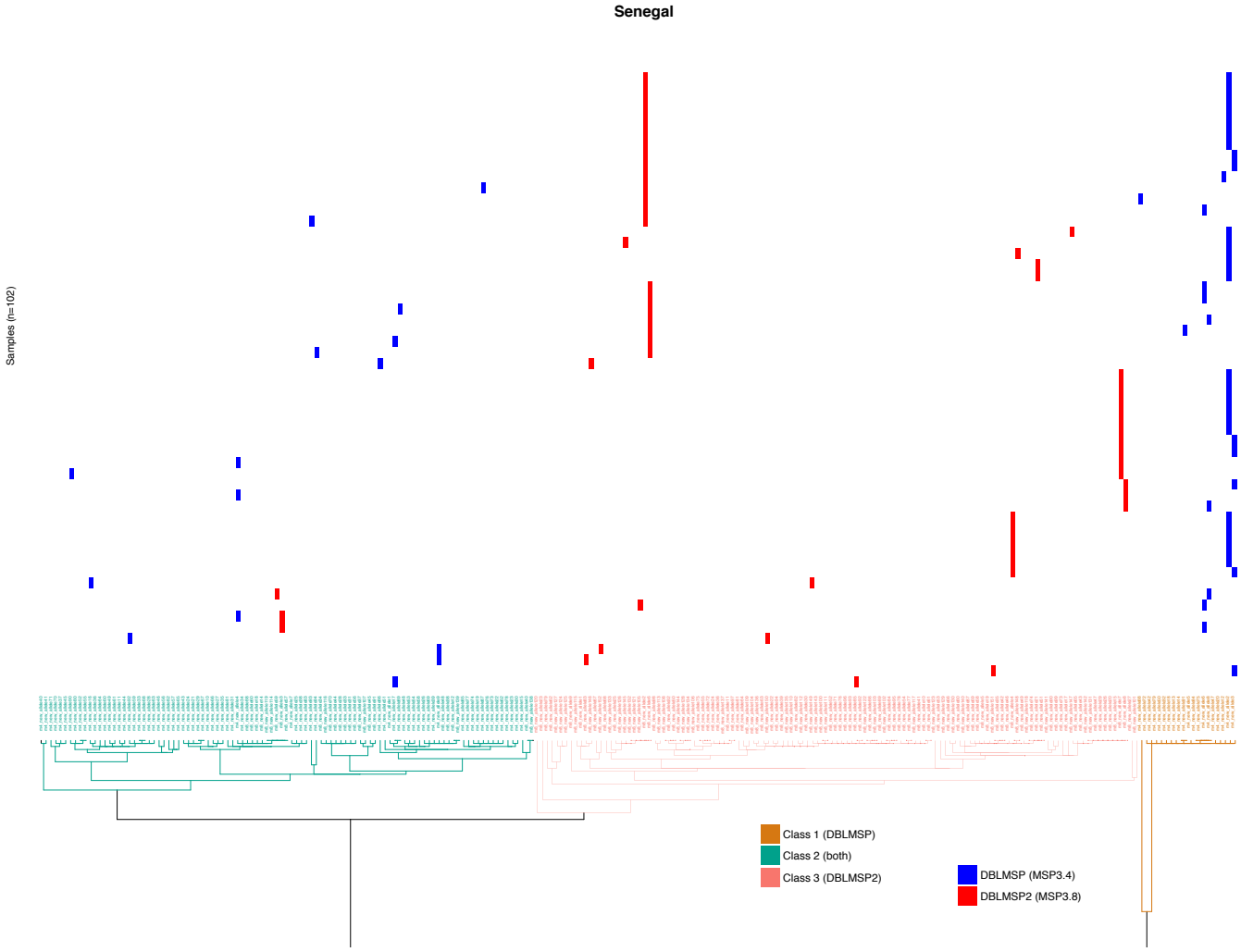


Figure A.12: LD patterns in Senegal. Each row represents a sample, with a red dot in the column corresponding to its DBLMSP allele and a blue dot in the column corresponding to its DBLMSP2 allele. Samples are sorted according to genotype similarity, whereas alleles are sorted based on their distance in the phylogenetic tree.

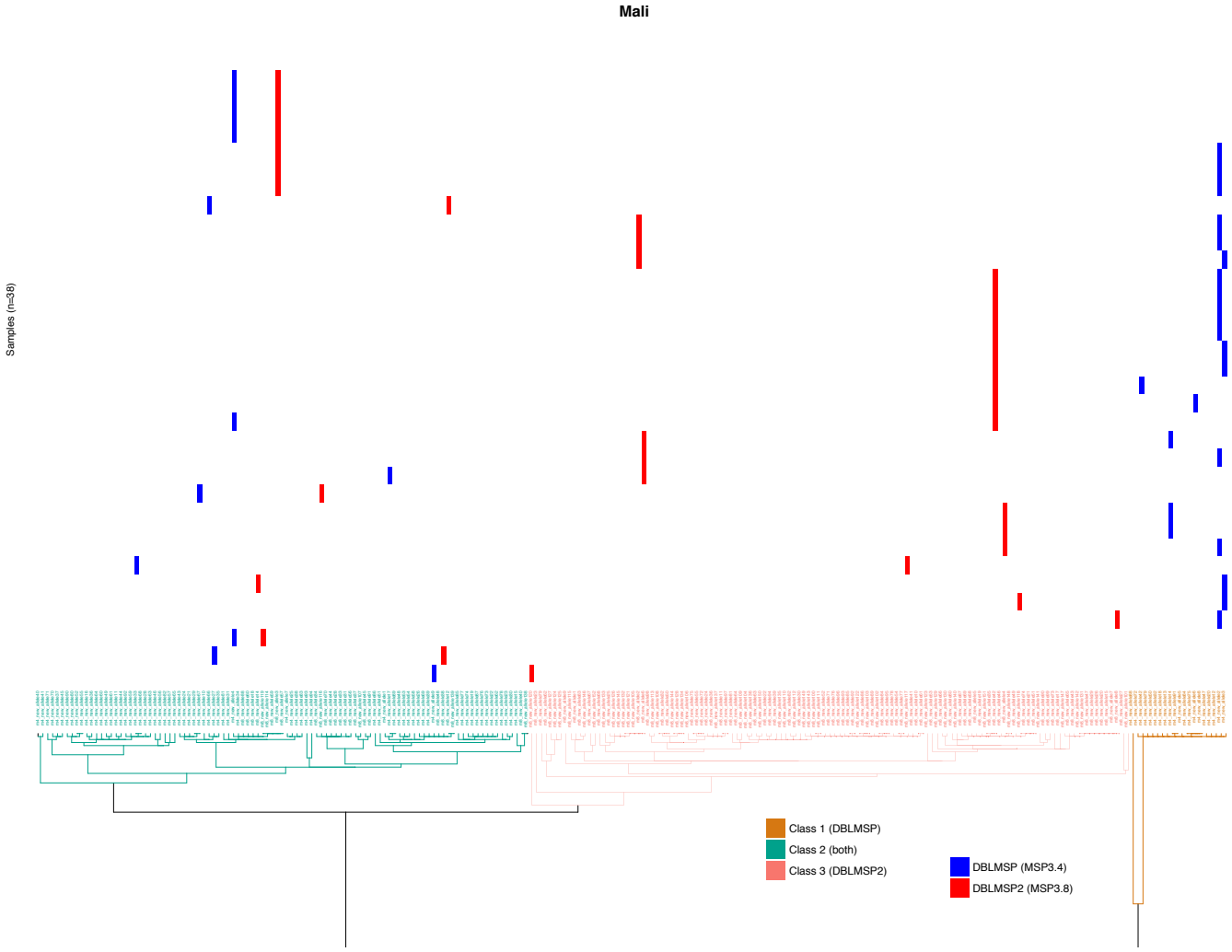


Figure A.13: LD patterns in Mali. Each row represents a sample, with a red dot in the column corresponding to its DBLMSP allele and a blue dot in the column corresponding to its DBLMSP2 allele. Samples are sorted according to genotype similarity, whereas alleles are sorted based on their distance in the phylogenetic tree.

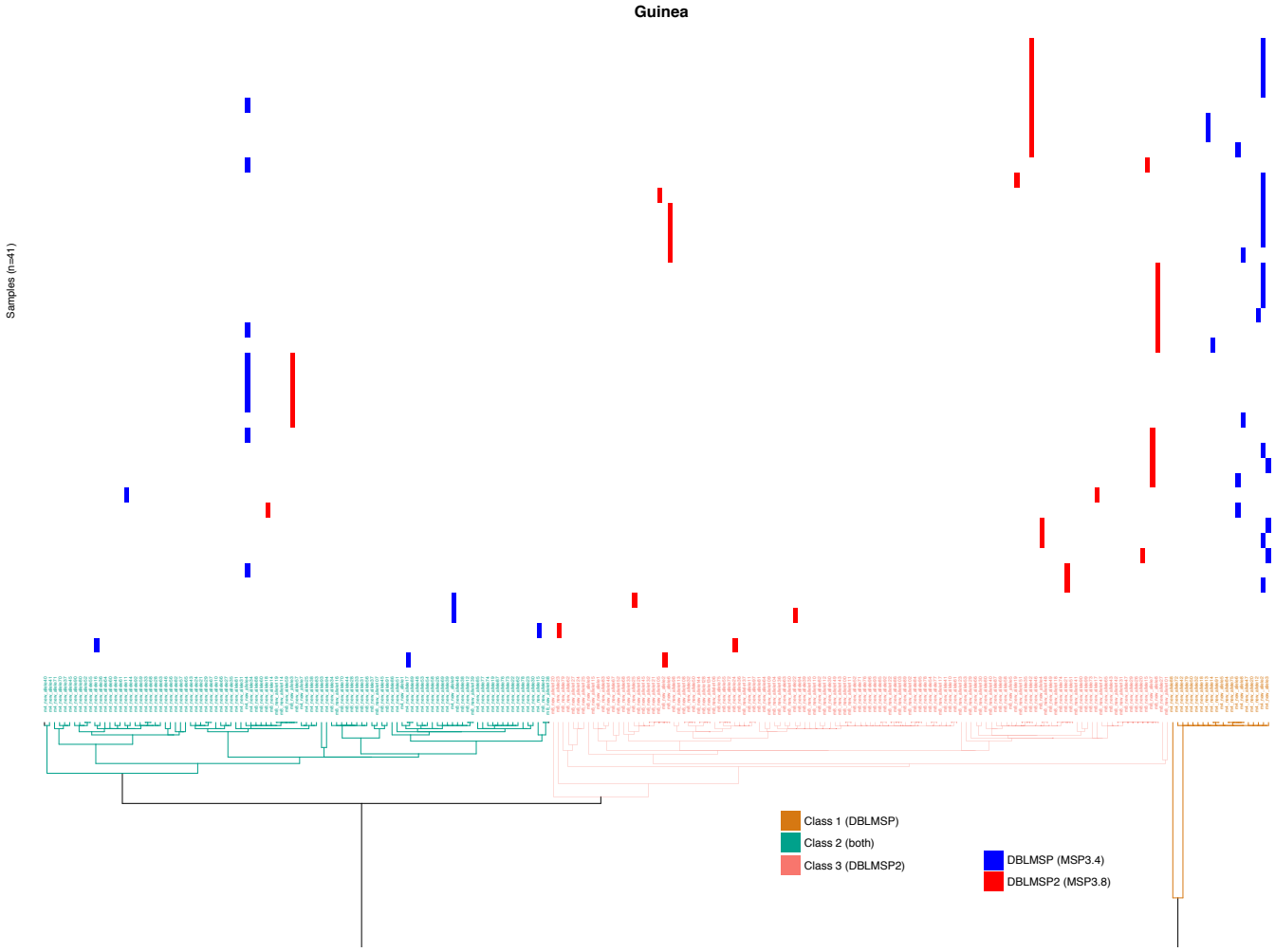


Figure A.14: LD patterns in Guinea. Each row represents a sample, with a red dot in the column corresponding to its DBLMSP allele and a blue dot in the column corresponding to its DBLMSP2 allele. Samples are sorted according to genotype similarity, whereas alleles are sorted based on their distance in the phylogenetic tree.

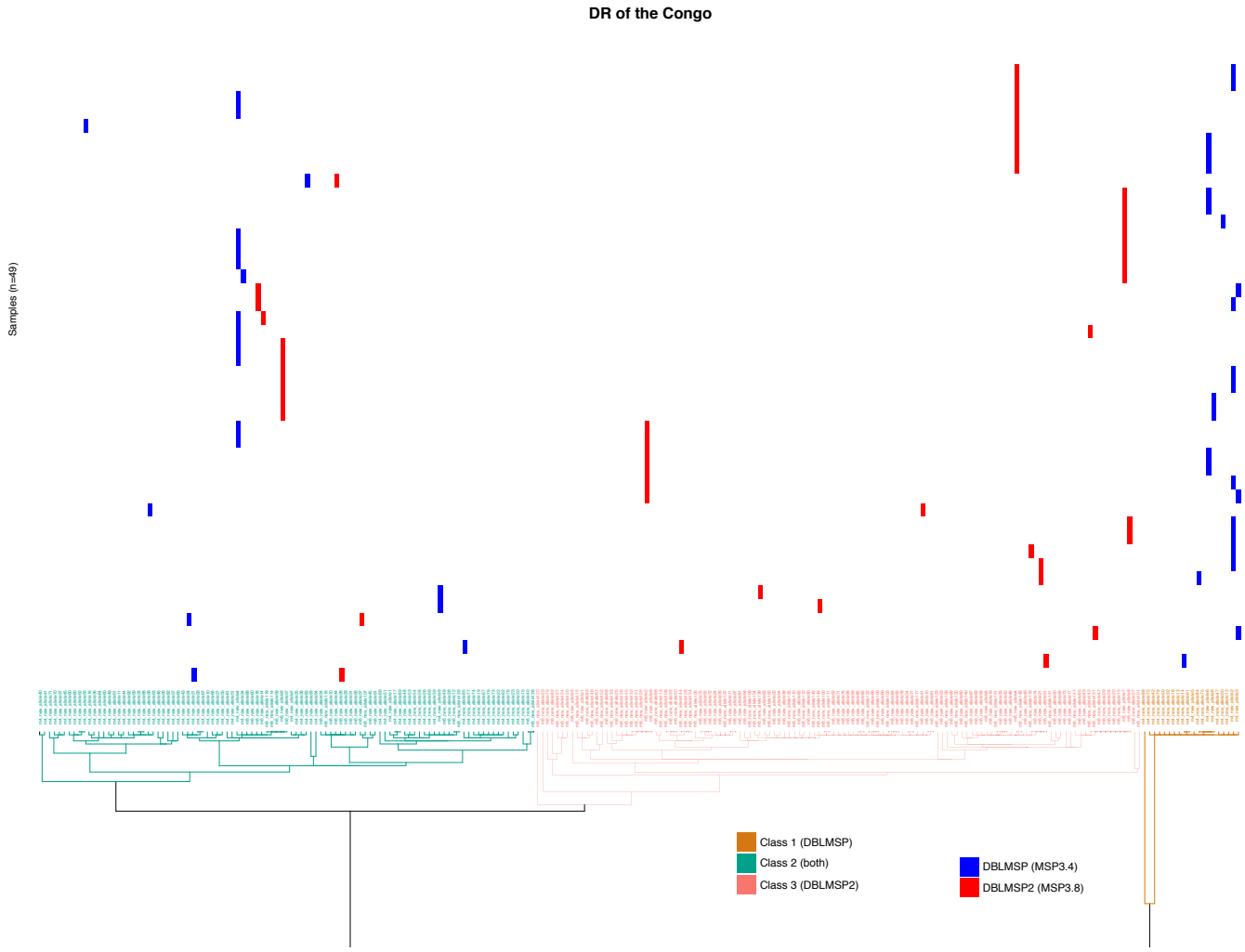


Figure A.15: LD patterns in Congo. Each row represents a sample, with a red dot in the column corresponding to its DBLMSP allele and a blue dot in the column corresponding to its DBLMSP2 allele. Samples are sorted according to genotype similarity, whereas alleles are sorted based on their distance in the phylogenetic tree.

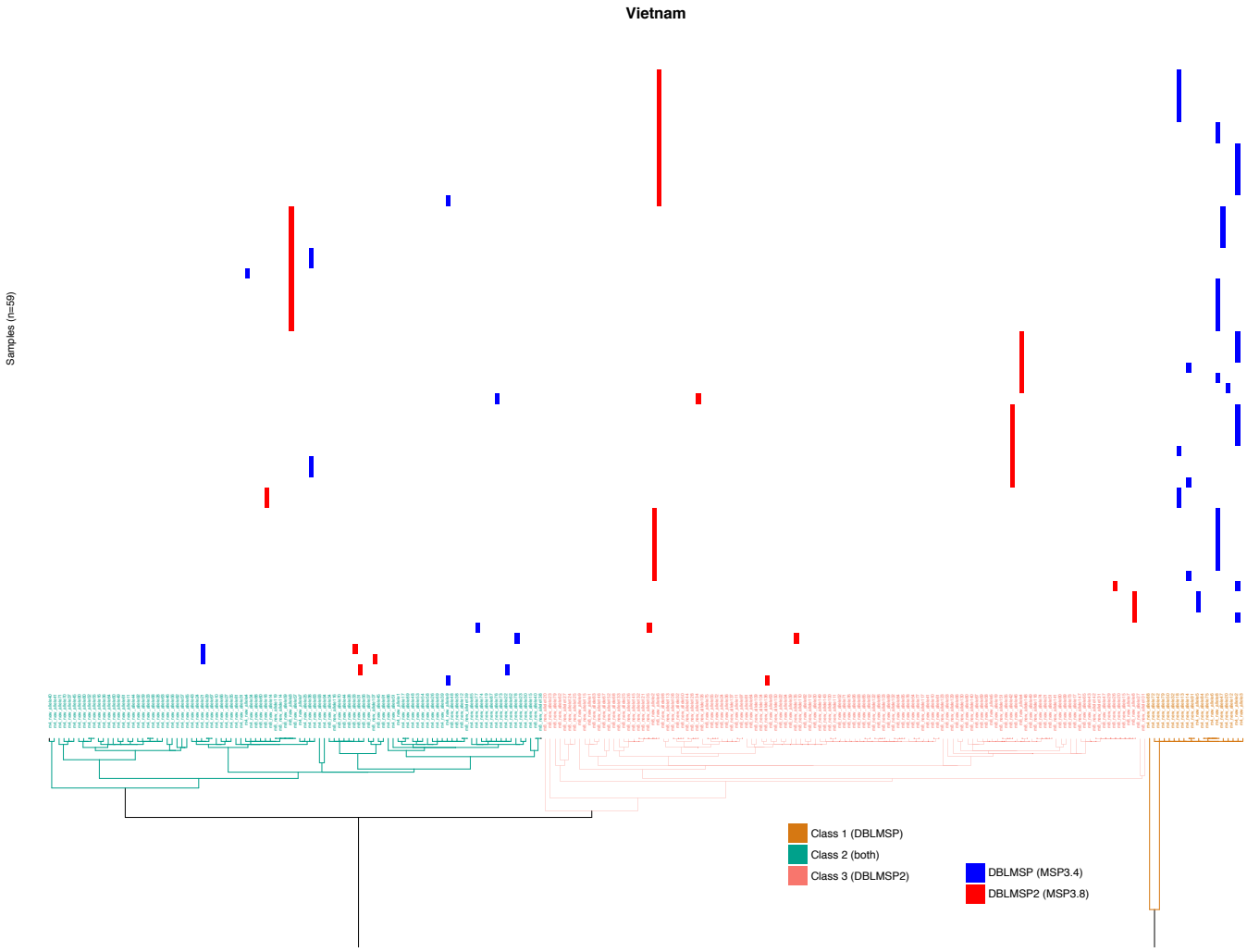


Figure A.16: LD patterns in Vietnam. Each row represents a sample, with a red dot in the column corresponding to its DBLMSP allele and a blue dot in the column corresponding to its DBLMSP2 allele. Samples are sorted according to genotype similarity, whereas alleles are sorted based on their distance in the phylogenetic tree.

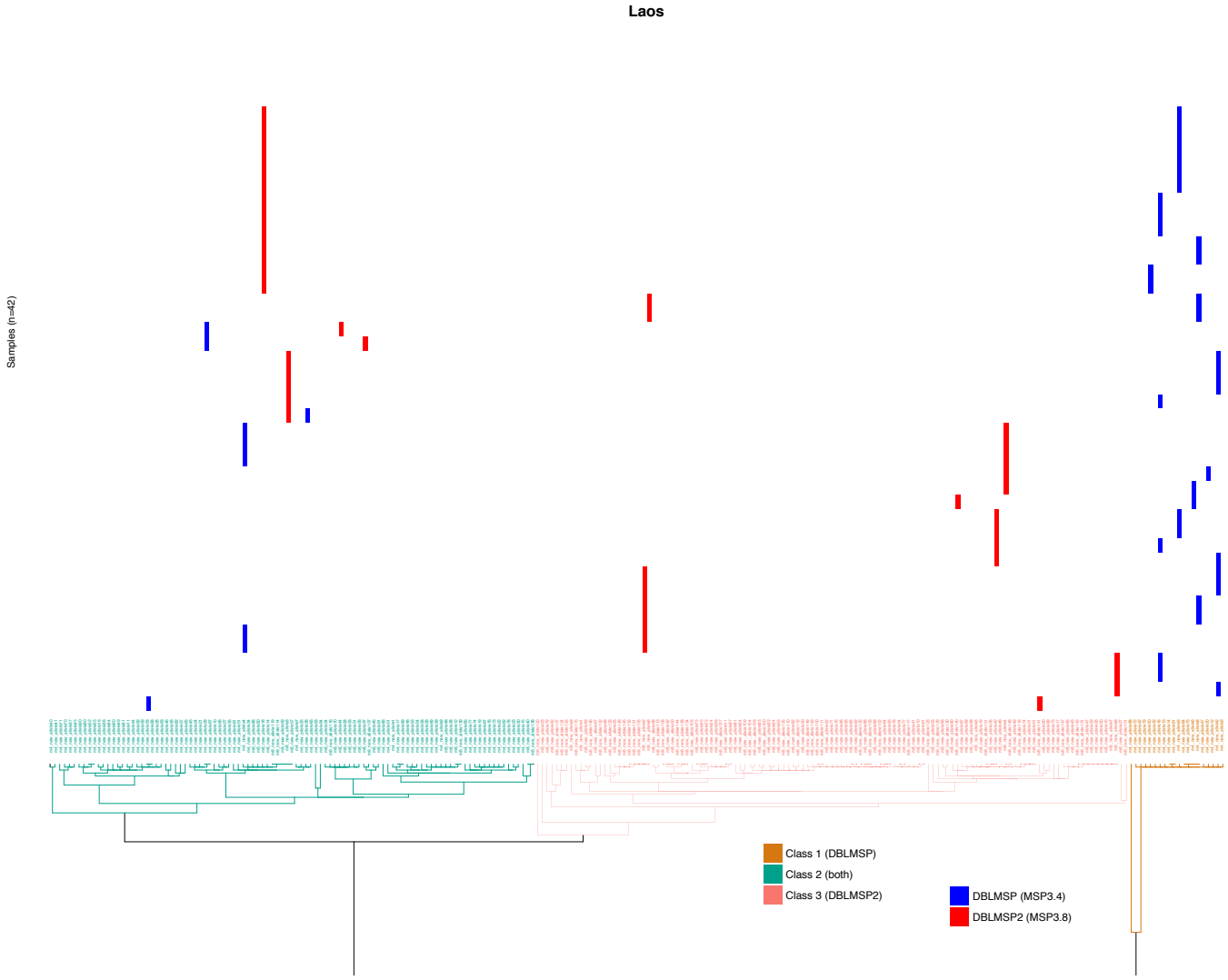


Figure A.17: LD patterns in Laos. Each row represents a sample, with a red dot in the column corresponding to its DBLMSP allele and a blue dot in the column corresponding to its DBLMSP2 allele. Samples are sorted according to genotype similarity, whereas alleles are sorted based on their distance in the phylogenetic tree.

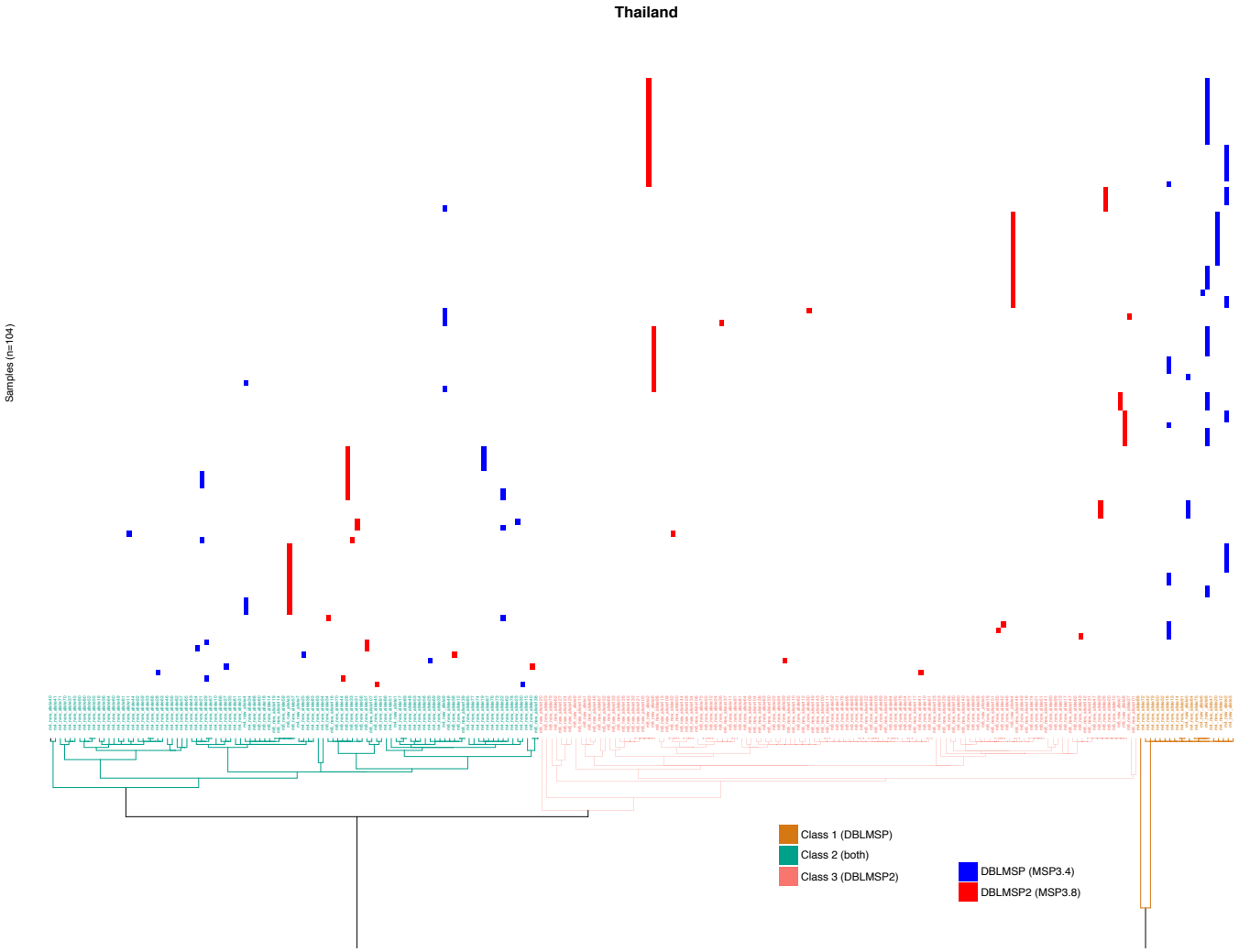


Figure A.18: LD patterns in Thailand. Each row represents a sample, with a red dot in the column corresponding to its DBLMSP allele and a blue dot in the column corresponding to its DBLMSP2 allele. Samples are sorted according to genotype similarity, whereas alleles are sorted based on their distance in the phylogenetic tree.

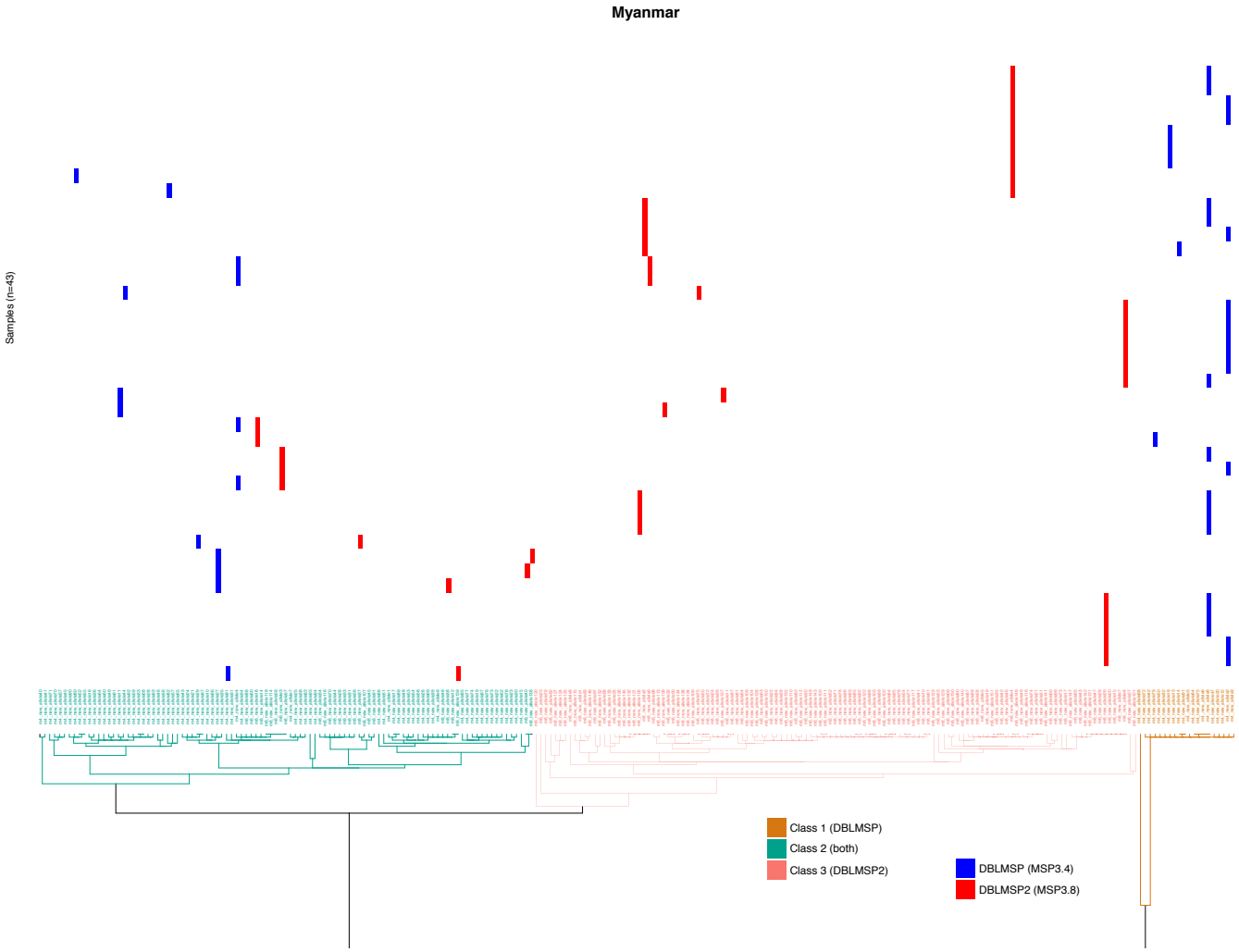


Figure A.19: LD patterns in Myanmar. Each row represents a sample, with a red dot in the column corresponding to its DBLMSP allele and a blue dot in the column corresponding to its DBLMSP2 allele. Samples are sorted according to genotype similarity, whereas alleles are sorted based on their distance in the phylogenetic tree.

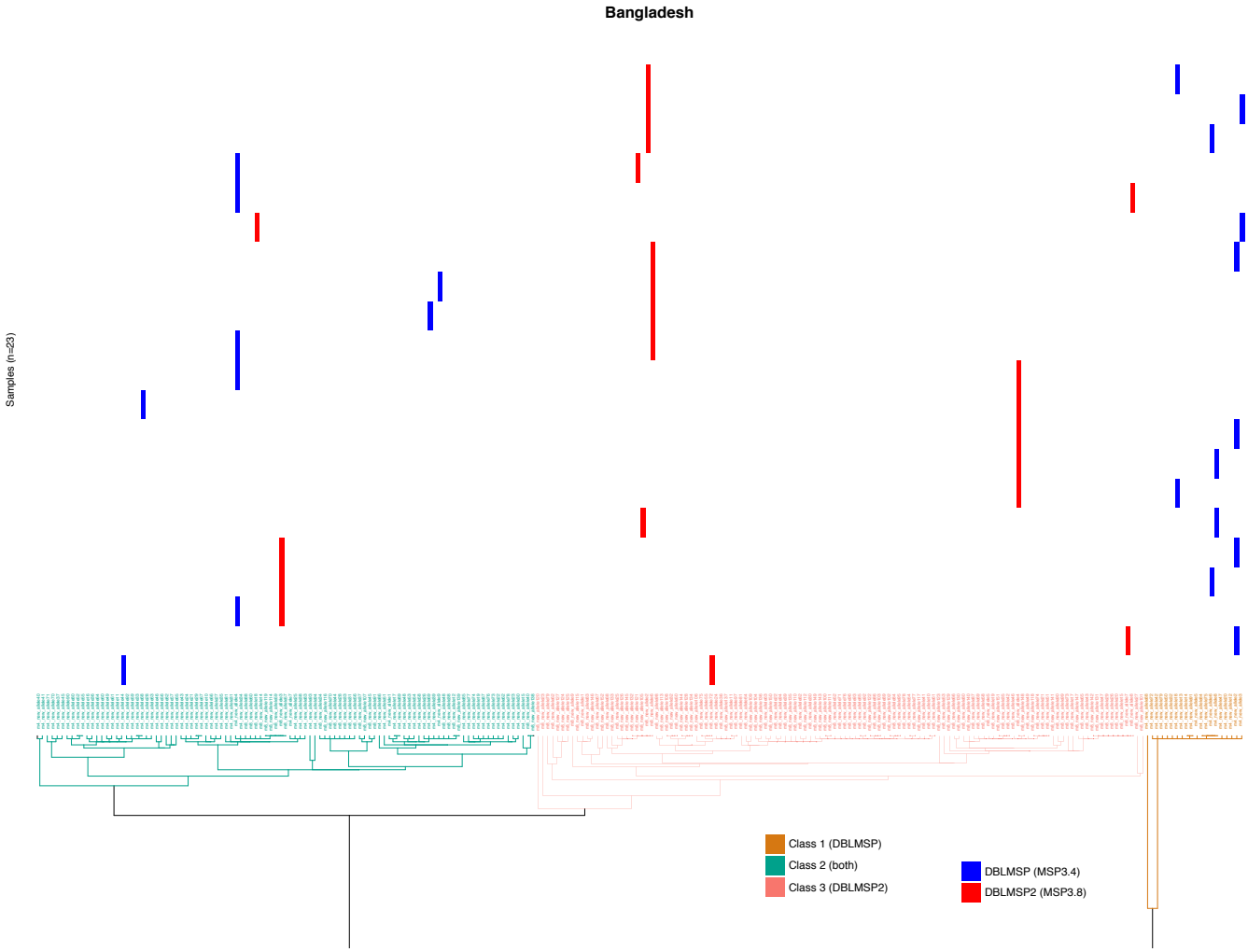


Figure A.20: LD patterns in Bangladesh. Each row represents a sample, with a red dot in the column corresponding to its DBLMSP allele and a blue dot in the column corresponding to its DBLMSP2 allele. Samples are sorted according to genotype similarity, whereas alleles are sorted based on their distance in the phylogenetic tree.

Appendix B

Methods

B.1 Graph input

B.1.1 Initial *DBLMSP*-only graph

The graph input integrates information obtained through 3 independent methods: de-novo assemblies from PacBio data, variants called by the `GATK` [DePristo et al., 2011] best practices pipeline and variants called by the `Cortex` [Iqbal et al., 2012] independent workflow from Illumina data (schematic in Fig. 5.5 (a)). The `GATK` genotyper is based on read alignment to the 3D7 reference genome, whereas `Cortex` is based on comparing an assembly graph of the sample with that of the reference genome, and determining variant coordinates by mapping flanking regions to the 3D7 reference genome. Due to its partial de-novo assembly, `Cortex` has been shown to be able to access some of the complex variation in divergent regions of the genome, as long as the flanking sequences are sufficiently homologous to 3D7 to be aligned.

The `GATK` calls were generated by collaborators from the Pf3k project and are available on the project website, but we used `GQT` [Layer, 2016] to select variants segregating only in the field samples, thus excluding samples from genetic crosses and laboratory samples that would be used for validation. We generated the `Cortex` calls

using the wrapper script run_calls.pl (within the software package) with a k-mer of size 31 and the independent workflow.

The PacBio assemblies were generated by collaborators from the Pf3k project, but we converted them to variant calls using the wrapper script run_calls.pl from `Cortex` with the –gt_assemblies option. The variant calls obtained through these 3 methods were combined into a single VCF file with the script combine_vcfs.pl that is part of the `Cortex` package.

The final file fed into the graph construction script is a wormtable [Kelleher et al., 2013] index of a VCF file containing 413 variants present in the *DBLMSP* gene. Note, however, that some of them are several hundred base pairs long, since they are derived from contigs assembled by `Cortex`. The following section describes how these variants and the haplotypes arising from their allelic combinations are represented in the graph.

B.1.2 Joint *DBLMSP* and *DBLMSP2* graph

The reference panel and methods used to call variants and integrate them into a single VCF file were the same as in the previous section. We added the 2 novel *DBLMSP* contigs specified in Section B.4 to the final combined VCF using a custom *python* script.

In addition to the single gene graph, we also included the novel segment of *DBLMSP* sequence from the validation strain GN01, i.e. the 400 bp sequence that `gramtools` could not recover in Section 5.4.1 (Fig. 5.7). We added this sequence (extracted from the PacBio assembly) directly to its graph locus after constructing the graph (detailed in the next section) and confirming that the sequence is novel indeed.

B.2 Construction of reference graph

A reference graph is constructed by iterating through the list of variants and building a catalogue of alternative alleles or haplotypes at each variable location (Fig. 5.5 (b)). We sketch the steps required in the subsequent paragraphs. We give pseudocode for the construction of the final joint graph of *DBLMSP* and *DBLMSP2* is given in Algorithm 4. The initial graph of *DBLMSP* only was constructed through a slight variant of this algorithm. We will highlight in text where and what the differences are.

In the case of overlapping variants, the variable locus is defined by the smallest left-coordinate and the largest right-coordinate from the set of variants that overlap. Then the set of alternative sequences covering the locus is obtained by padding the variants with the reference sequence up to the left-most and right-most coordinates (Algorithm 4, line 14). The result of this step is the set of observed sequences at each variable locus in the graph.

Algorithm 3 Cluster overlapping variants

Input: parsed VCF *variants* in coordinate order
Output: list *clusters* containing lists of overlapping variants or single variants

```
1:  i ← 0
2:  clusters = ∅
3:  for all v ∈ variants do
4:      if v.start ≤ clusters[i].end then
5:          clusters[i].append(v)
6:          if clusters[i].end < v.end then
7:              clusters[i].end = v.end
8:      else
9:          i ← i + 1
10:         clusters[i] = ∅
11:         clusters[i].append(v)
12:         clusters[i].end = v.end
```

In order to have the graph support the detection of novel recombinants, new paths corresponding to combinations of observed sequences needed to be created, for reads

to map to. This can be done in two ways, as follows. The first option was chosen to build the *DBLMSP*-only graph, whereas the second option was chosed to build the final joint graph.

The simplest way to support recombinants is by adding their sequences to the set of alternatives from a variable locus. Whenever all sequences from a variable locus share an aligned k-mer in the MSA, we treated that k-mer as a recombination breakpoint and constructed new sequences representing recombinants between observed sequences at that respective breakpoint. However, the size of the k-mer needs to be chosen carefully; a small k-mer can produce a combinatorial explosion in the number of sequences added. We chose $k = 21$, but this posed an issue in the DBL domain, where the two dimorphic types are so diverged, that no *21*-mer shared by all sequences can be found[1]. However, there is extended sharing between sequences belonging to the same dimorphic type. Moreover, previous studies reported frequent recombination within dimorphic type, but limited recombination between types. Therefore, in order to allow new recombinants within type, but not between types, we identified shared *21*-mers within a type and built new haplotypes that would arise from a recombination at that spot. These were added to the set of observed sequences at the graph locus representing the DBL domain, rendering a total of 3,383 alternative sequences. This method was used in the *DBLMSP*-only graph construction and would replace the **collapse** function in Algorithm 4, line 18. The rest of that algorithm still applies to this graph.

Although the recombinants representation we just described is simple and convenient, it leads to graphs with large sites containing many long alleles. Such sites can create difficulties in inference, as it emerged in Section 5.7, where a cross-dimorphic class recombinant could not be detected, or in Section 5.6, where additional unobserved alleles led to poorer inference. In order to overcome these issues, we chose

[1]maximum sharing is a *12*-mer; however, if we had chosen $k = 12$, the DBL locus in the graph would have contained tens of thousands of sequences

a more natural way to support recombinants when we constructed the final, joint gene graph. If all possible sequences from a variable locus share an aligned k-mer in the MSA, there is the option of collapsing it onto a single graph branch, leaving two graph sites containing the sequences adjacent to the k-mer on each side (Fig. B.1, Algorithm 5). This would allow reads to be mapped along all the pair combinations of alleles at the two sites. Normally the size of the collapsed k-mer would need to be chosen considering the performance trade offs discussed in Chapter 4, Section 4.6. However, our current graph represents only a small genomic region, so computation time is not an issue. Hence, we chose a small $k = 6$, in order to allow many recombination paths. We performed a multiple sequence alignment of the catalogue of sequences from each variable locus and split the locus into two whenever a shared *6*-mer was found. This step determines the final list of variable loci in the graph and their coordinates.

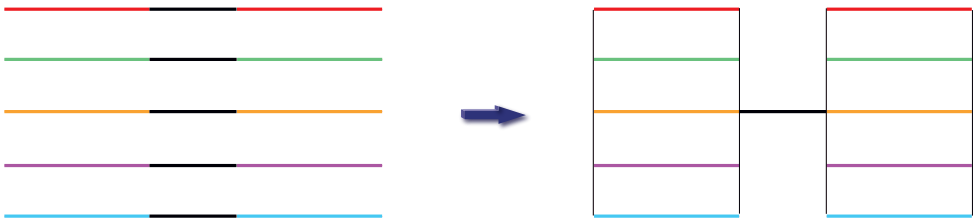


Figure B.1: Toy example of shared k-mer collapsing. On the left, a multiple sequence alignment of 5 sequences sharing a k-mer (black line). After collapsing, there are 2 MSAs are left, bridged by the k-mer . The pseudocode for this step is described in Algorithm 5. Note that some details have been omitted from the algorithm for clarity.

At this point, the list of variable loci in the graph, and the catalogue of haplotypes they contain, is complete and can be converted to a linearised graph for vBWT indexing. Note that the vBWT encoding does not allow distinct variation markers to be adjacent, so adjacent variants (where the start of variant 2 is right next to the end of variant 1) are considered 'overlapping' and collated within the same locus in

Algorithm 4 Graph construction

Input: reference genome ref, parsed VCF $variants$, collapse k-mer size k
Output: linear PRG, masks over sites and alleles

$prg = ""$
$mask_sites = \emptyset$
$mask_alleles = \emptyset$
$site \leftarrow 5$
$prev_cl_end \leftarrow 0$
$clusters =$ **cluster_overlap_var**$(variants)$
for all $c \in clusters$ **do**
    $prg = prg + ref[prev_cl_end : c.start]$
    $mask_sites = mask_sites + [0] * (c.start - prev_cl_end)$
    $mask_alleles = mask_alleles + [0] * (c.start - prev_cl_end)$
    $prev_cl_end = c.end$
    $seqs = \{ref[c.start : c.end]\}$
    for all $v \in c$ **do**
        $seqs.append(ref[c.start : v.start] + v.ALT + ref[v.end : c.end])$
    $collapsed_haps = \emptyset$
    if $len(seqs) > 2$ **then**
        $msa =$ **clustalw**$(seqs)$
        $collapsed_haps =$ **collapse**(msa, k)
    if $len(seq) \leq 2$ **or** $collapsed_haps = \emptyset$ **then**
        $collapsed_haps.append(seqs)$
    for all $haps \in collapsed_haps$ **do**
        $no_alts = len(haps)$
        if $no_alts > 1$ **then**
            $prg = prg + str(site)$
            $mask_sites = mask_sites + [0]$
            $mask_alleles = mask_alleles + [0]$
            for all $i \in 1..no_alts$ **do**
                $prg = prg + haps[i-1] + str(site + i\%no_alts)$
                $mask_sites = mask_sites + [str(site)] * len(haps[i-1]) + [0]$
                $mask_alleles = mask_alleles + [str(i+1)] * len(haps[i-1]) + [0]$
            $site = site + 2$
        else
            $prg = prg + haps[0]$
            $mask_sites = mask_sites + [0] * haps[0]$
            $mask_alleles = mask_alleles + [0] * haps[0]$
$prg = prg + ref[prev_cl_end :]$
$mask_sites = mask_sites + [0] * len(ref[prev_cl_end :])$
$mask_alleles = mask_alleles + [0] * len(ref[prev_cl_end :])$

Algorithm 5 Collapse aligned shared k-mer in MSA

Input: multiple sequence alignment in clustal format msa, k-mer size k

Output: list new_haps containing: list of haplotypes for the locus before the collapsed k-mer , collapsed k-mer , list of haplotypes for the locus after the collapsed k-mer

$i \leftarrow 0$
$prev_i = 1$
$l = \text{length}(msa)$
$first_seq = msa[1]$
$left_locus = \emptyset$
$new_haps = []$
while $i + k \leq l$ **do**
    $k2 \leftarrow 0$
    while $i + k + k2 \leq l$ **do**
        $share \leftarrow 0$
        for all $seq \in msa$ **do**
            if $seq[i : (i + k + k2)] = first_seq[i : (i + k + k2)]$ **then**
                $share \leftarrow share + 1$
        if $share = no_seq_msa$ **then**
            $k2 \leftarrow k2 + 1$
    if $k2 = 0$ **then**
        $i \leftarrow i + 1$
    else
        for all $seq \in msa$ **do**
            $left_locus = left_locus \cup \{seq[(prev_i - 1) : i]\}$
        $new_haps = new_haps.\text{append}(left_locus)$
        if $i + k + k2 \leq l$ **then**
            $new_haps = new_haps.\text{append}(\{first_seq[i : (i + k + k2 - 1)]\})$
        else
            $new_haps = new_haps.\text{append}(\{first_seq[i : l]\})$

the graph.

The compressed suffix array index for the *DBLMSP*-only reference graph can be stored in a file of size 9 MB. The additional mask data structures over the variable sites and the alternative alleles required by `gramtools` are constructed at the same time as the linear PRG, as it can be seen from Algorithm 4. The reference graph contains 150,144 unique 9-mers whose suffix array intervals were precalculated by `gramtools` and stored in a file of size 8.3 MB.

The compressed suffix array index for the joint gene reference graph can be stored in a file of size 5 MB. The graph contains 150,550 unique 9-mers whose suffix array intervals were precalculated by `gramtools` and stored in a file of size 8.3 MB.

B.3 Defining the edit distance between a read pileup and a reference sequence

Chromosome 10 was divided into 100 bp windows after filtering out the subtelomeric regions. Then the windows were split into categories of 5% GC content and the median coverage in each category was calculated. A gap was defined as a base position whose read coverage is smaller than 1/10 of the median coverage across all windows with a similar GC content. This measure was chosen in order to account for common coverage fluctuations along regions of the *P. falciparum* genome with varying levels of local GC content [Sepúlveda et al., 2013].

At a base position that is not classified as gap, a mismatch was defined if the number of reads that disagree with the reference base is higher than the number of reads that agree.

B.4 Discovery of novel contigs

One initial explanation for the consistent gaps in Fig. 5.10 was that these are true deletions. To test this, we cut out the sequence with no coverage from the personalised references and remapped the reads, to see whether there would be reads mapping across the deleted junction. No reads supported the deletion, all were clipped around the deleted junction.

We then hypothesised that the cause of these gaps could be that the variation present in this subset of samples is not contained in the reference graph. We first looked in the raw variant calls for these samples to see whether there were any variants missed by the graph construction procedure. None of the samples had anything called in the regions covering the gaps, so we suspected they might have novel complex variants that had not been discovered by `Cortex`. We investigated this further by mapping the raw contigs assembled by `Cortex` to the personalised references inferred from the reference graph with `BWA-MEM`. We found 2 samples with contigs anchored in *DBLMSP* which are spanning the entire gap (Fig. B.2 (b), Fig. B.3 (b)).

The variants contained in these 2 contigs were not called by `Cortex` because mapping them to 3D7 produces a split alignment where the ends of the contigs covering the *DBLMSP* gap map to the DBL domain in *DBLMSP2*(Fig. B.2 (a), Fig. B.3 (a)). On the other hand, the personalised references contain the variants surrounding the gap region and are therefore, closer to the contigs, determining them to map confidently in DBLMSP. However, one of the contigs (Fig. B.2 (a)) contains a 278 bp sequence repeated in the DBL domain of DBLMSP2, which is also present in the PacBio assembly of one lab sample that we used for validation, GN01 (Fig. B.4). This suggests the occurence of non-allelic gene conversion events between these 2 paralogous genes as an additional source of variation.

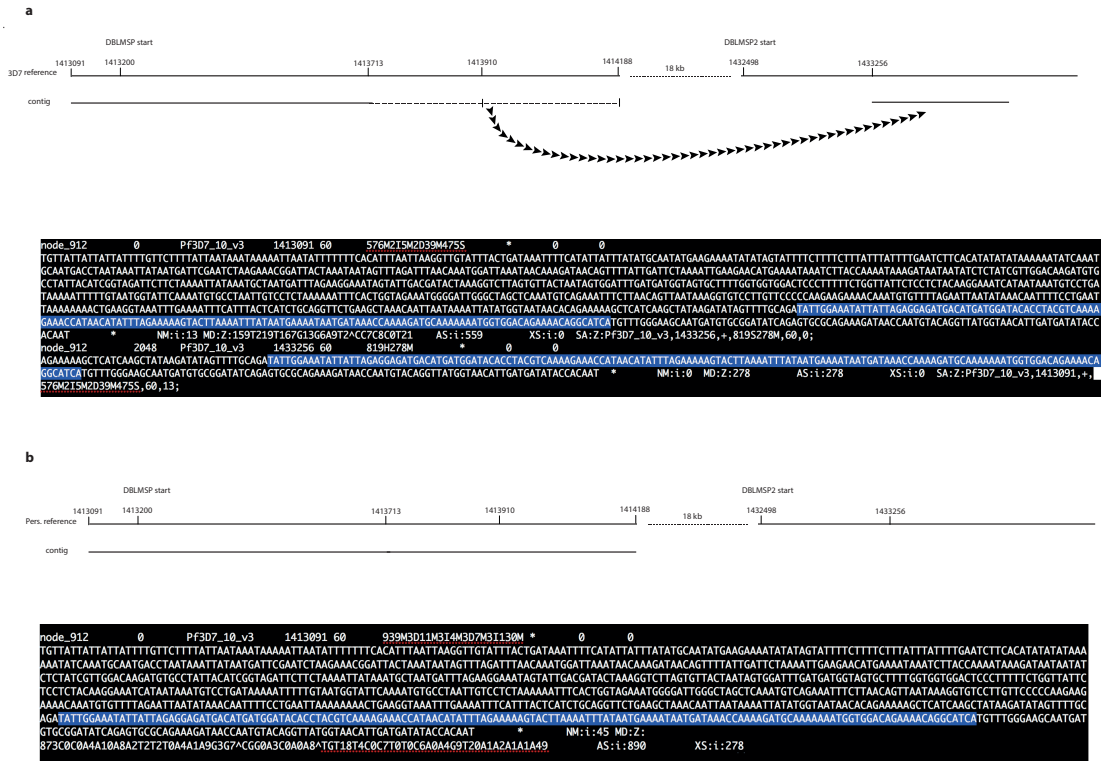


Figure B.2: Mapping of first novel contig to 3D7 **(a)**, and to the graph-inferred reference **(b)**. The contig section covering the pileup gaps is highlighted in blue. Top diagram and extract from SAM file show how most of the contig maps in DBLMSP, but its tail (including blue section) has a better match in the DBL domain of DBLMSP2. Bottom diagram and SAM extract show the contig maps entirely in *DBLMSP* when the graph-inferred reference is used.

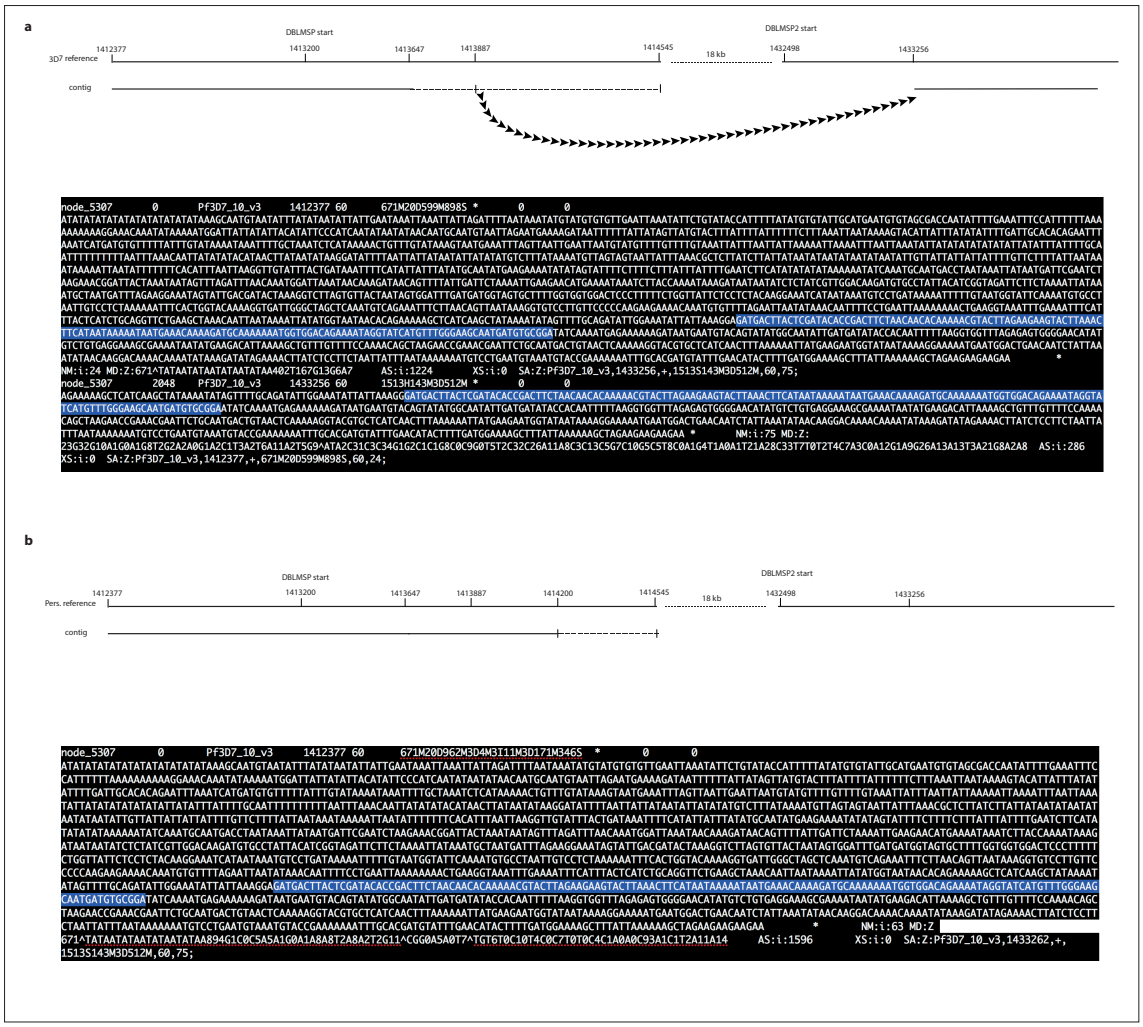


Figure B.3: Mapping of second novel contig to 3D7 **(a)**, and to the graph-inferred reference **(b)**. The contig section covering the pileup gaps is highlighted in blue. Top diagram and extract from SAM file show how most of the contig maps in DBLMSP, but its tail (including blue section) has a better match in the DBL domain of DBLMSP2. Bottom diagram and SAM extract show the contig maps entirely in *DBLMSP* when the graph-inferred reference is used.

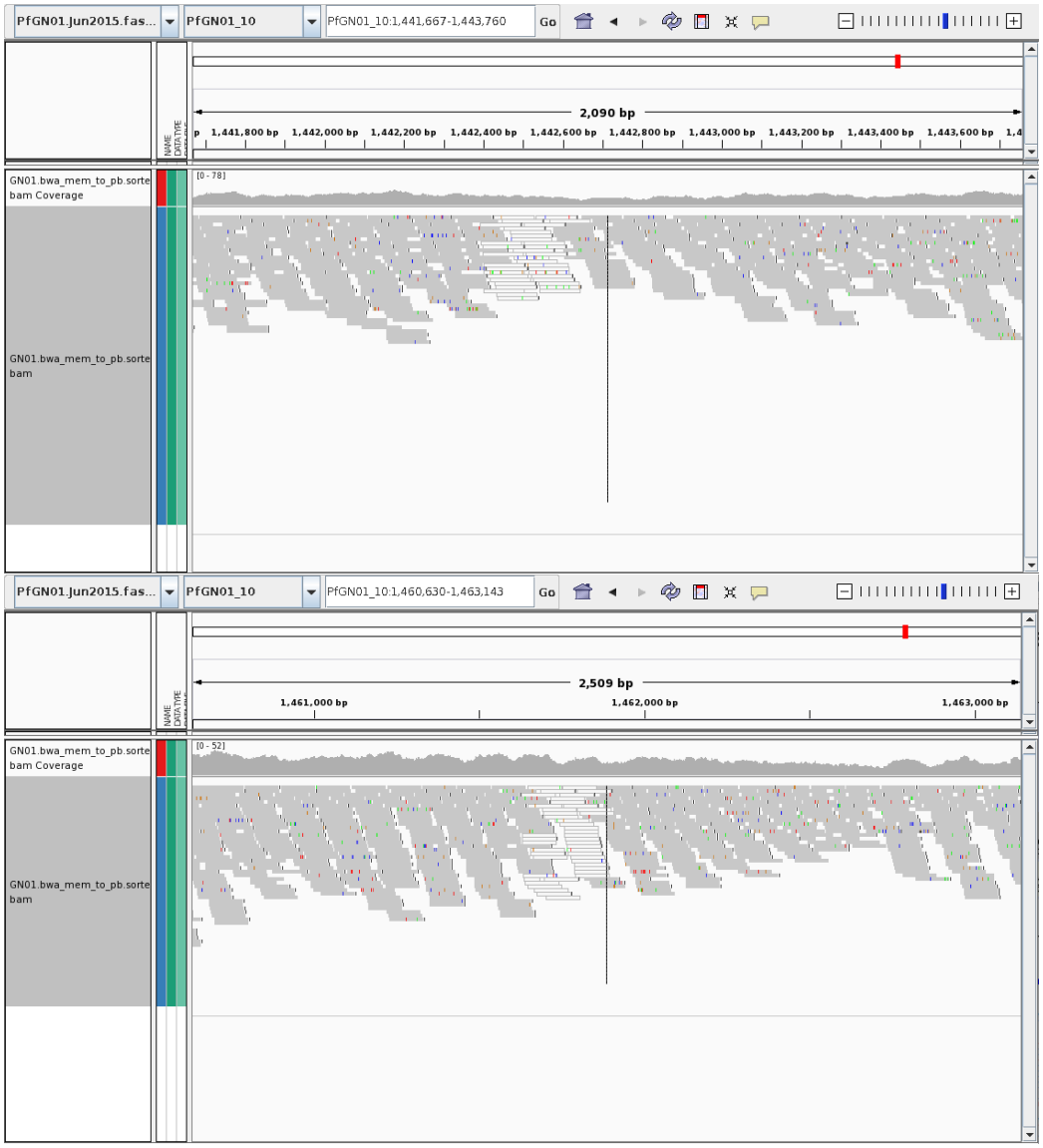


Figure B.4: IGV screenshot of pileups of Illumina reads mapped to the PacBio assembly for sample GN01, in the *DBLMSP* region (top) and the *DBLMSP2*region (bottom). White reads map equally well in both genes because of a 278 bp sequence shared by their DBL domains.

Bibliography

[Aflitos et al., 2014] Aflitos, S., Schijlen, E., de Jong, H., de Ridder, D., Smit, S., Finkers, R., Wang, J., Zhang, G., Li, N., Mao, L., Bakker, F., Dirks, R., Breit, T., Gravendeel, B., Huits, H., Struss, D., Swanson-Wagner, R., van Leeuwen, H., van Ham, R. C., Fito, L., Guignier, L., Sevilla, M., Ellul, P., Ganko, E., Kapur, A., Reclus, E., de Geus, B., van de Geest, H., te Lintel Hekkert, B., van Haarst, J., Smits, L., Koops, A., Sanchez-Perez, G., van Heusden, A. W., Visser, R., Quan, Z., Min, J., Liao, L., Wang, X., Wang, G., Yue, Z., Yang, X., Xu, N., Schranz, E., Smets, E., Vos, R., Rauwerda, J., Ursem, R., Schuit, C., Kerns, M., van den Berg, J., Vriezen, W., Janssen, A., Datema, E., Jahrman, T., Moquet, F., Bonnet, J., Peters, S., and Peters, S. (2014). Exploring genetic variation in the tomato (*Solanum* section *Lycopersicon*) clade by whole-genome sequencing. *The Plant Journal*, 80(1):136–148.

[Amambua-Ngwa et al., 2012] Amambua-Ngwa, A., Tetteh, K. K. A., Manske, M., Gomez-Escobar, N., Stewart, L. B., Deerhake, M. E., Cheeseman, I. H., Newbold, C. I., Holder, A. A., Knuepfer, E., Janha, O., Jallow, M., Campino, S., MacInnis, B., Kwiatkowski, D. P., and Conway, D. J. (2012). Population Genomic Scan for Candidate Signatures of Balancing Selection to Guide Antigen Characterization in Malaria Parasites. *PLoS Genetics*, 8(11).

[Barry and Arnott, 2014] Barry, A. E. and Arnott, A. (2014). Strategies for designing and monitoring malaria vaccines targeting diverse antigens.

[Barry et al., 2009] Barry, A. E., Schultz, L., Buckee, C. O., and Reeder, J. C. (2009). Contrasting population structures of the genes encoding ten leading vaccine-candidate antigens of the human malaria parasite, Plasmodium falciparum. *PLoS ONE*, 4(12).

[Barry et al., 2011] Barry, A. E., Trieu, A., Fowkes, F. J. I., Pablo, J., Kalantari-Dehaghi, M., Jasinskas, A., Tan, X., Kayala, M. A., Tavul, L., Siba, P. M., Day, K. P., Baldi, P., Felgner, P. L., and Doolan, D. L. (2011). The Stability and Complexity of Antibody Responses to the Major Surface Antigen of Plasmodium falciparum Are Associated with Age in a Malaria Endemic Area. *Molecular & Cellular Proteomics*, 10(11):M111.008326–M111.008326.

[Bennett, 2009] Bennett, P. M. (2009). Plasmid encoded antibiotic resistance: acquisition and transfer of antibiotic resistance genes in bacteria. *British Journal of Pharmacology*, 153(S1):S347–S357.

[Bhatt et al., 2015] Bhatt, S., Weiss, D. J., Cameron, E., Bisanzio, D., Mappin, B., Dalrymple, U., Battle, K., Moyes, C. L., Henry, A., Eckhoff, P. A., Wenger, E. A., Briët, O., Penny, M. A., Smith, T. A., Bennett, A., Yukich, J., Eisele, T. P., Griffin, J. T., Fergus, C. A., Lynch, M., Lindgren, F., Cohen, J. M., Murray, C. L. J., Smith, D. L., Hay, S. I., Cibulskis, R. E., and Gething, P. W. (2015). The effect of malaria control on Plasmodium falciparum in Africa between 2000 and 2015. *Nature*, 526(7572):207–211.

[Binks and Conway, 1999] Binks, R. H. and Conway, D. J. (1999). The major allelic dimorphisms in four Plasmodium falciparum merozoite proteins are not associated with alternative pathways of erythrocyte invasion.

[Blumenfeld and Huang, 1995] Blumenfeld, O. O. and Huang, C.-H. (1995). Molecular genetics of the glycophorin gene family, the antigens for MNSs blood groups:

Multiple gene rearrangements and modulation of splice site usage result in extensive diversification. *Human Mutation*, 6(3):199–209.

[Bowe et al., 2012] Bowe, A., Onodera, T., Sadakane, K., and Shibuya, T. (2012). Succinct de Bruijn Graphs. *Algorithms in Bioinformatics. WABI 2012. Lecture Notes in Computer Science*, 7534.

[Bradley et al., 2015] Bradley, P., Gordon, N. C., Walker, T. M., Dunn, L., Heys, S., Huang, B., Earle, S., Pankhurst, L. J., Anson, L., de Cesare, M., Piazza, P., Votintseva, A. A., Golubchik, T., Wilson, D. J., Wyllie, D. H., Diel, R., Niemann, S., Feuerriegel, S., Kohl, T. A., Ismail, N., Omar, S. V., Smith, E. G., Buck, D., McVean, G., Walker, A. S., Peto, T. E. A., Crook, D. W., and Iqbal, Z. (2015). Rapid antibiotic-resistance predictions from genome sequence data for Staphylococcus aureus and Mycobacterium tuberculosis. *Nature Communications*, 6:10063.

[Bray et al., 2016] Bray, N. L., Pimentel, H., Melsted, P., and Pachter, L. (2016). Near-optimal probabilistic RNA-seq quantification. *Nature Biotechnology*, 34(5):525–527.

[Burrows and Wheeler, 1994] Burrows, M. and Wheeler, D. (1994). A block-sorting lossless data compression algorithm. *Algorithm, Data Compression*, (124):18.

[Casillas and Barbadilla, 2017] Casillas, S. and Barbadilla, A. (2017). Molecular population genetics.

[Casola and Hahn, 2009] Casola, C. and Hahn, M. W. (2009). Gene conversion among paralogs results in moderate false detection of positive selection using likelihood methods. *Journal of Molecular Evolution*, 68(6):679–687.

[Chaisson and Pevzner, 2008] Chaisson, M. J. and Pevzner, P. A. (2008). Short read fragment assembly of bacterial genomes. *Genome research*, 18(2):324–30.

[Chikhi et al., 2014] Chikhi, R., Limasset, A., Jackman, S., Simpson, J., and Medvedev, P. (2014). On the representation of de Bruijn graphs. *ArXiv e-prints*.

[Chikhi and Rizk, 2012] Chikhi, R. and Rizk, G. (2012). Space-efficient and exact de Bruijn graph representation based on a bloom filter. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 7534 LNBI, pages 236–248. Springer, Berlin, Heidelberg.

[Chikhi and Rizk, 2013] Chikhi, R. and Rizk, G. (2013). Space-efficient and exact de Bruijn graph representation based on a Bloom filter. *Algorithms for Molecular Biology*, 8(1):22.

[Chiu et al., 2015] Chiu, C. Y. H., Hodder, A. N., Lin, C. S., Hill, D. L., Li Wai Suen, C. S. N., Schofield, L., Siba, P. M., Mueller, I., Cowman, A. F., and Hansen, D. S. (2015). Antibodies to the Plasmodium falciparum Proteins MSPDBL1 and MSPDBL2 Opsonize Merozoites, Inhibit Parasite Growth, and Predict Protection from Clinical Malaria. *Journal of Infectious Diseases*, 212(3):406–415.

[Church et al., 2011] Church, D. M., Schneider, V. A., Graves, T., Auger, K., Cunningham, F., Bouk, N., Chen, H. C., Agarwala, R., McLaren, W. M., Ritchie, G. R. S., Albracht, D., Kremitzki, M., Rock, S., Kotkiewicz, H., Kremitzki, C., Wollam, A., Trani, L., Fulton, L., Fulton, R., Matthews, L., Whitehead, S., Chow, W., Torrance, J., Dunn, M., Harden, G., Threadgold, G., Wood, J., Collins, J., Heath, P., Griffiths, G., Pelan, S., Grafham, D., Eichler, E. E., Weinstock, G., Mardis, E. R., Wilson, R. K., Howe, K., Flicek, P., and Hubbard, T. (2011). Modernizing reference genome assemblies. *PLoS Biology*, 9(7):e1001091.

[Church et al., 2015] Church, D. M., Schneider, V. A., Steinberg, K., Schatz, M. C., Quinlan, A. R., Chin, C.-S., Kitts, P. A., Aken, B., Marth, G. T., Hoffman, M. M.,

Herrero, J., Mendoza, M. L., Durbin, R., and Flicek, P. (2015). Extending reference assembly models. *Genome Biology*, 16(1):13.

[Claessens et al., 2014] Claessens, A., Hamilton, W. L., Kekre, M., Otto, T. D., Faizullabhoy, A., Rayner, J. C., and Kwiatkowski, D. (2014). Generation of Antigenic Diversity in Plasmodium falciparum by Structured Rearrangement of Var Genes During Mitosis. *PLoS Genetics*, 10(12):e1004812.

[Collins, 2003] Collins, F. S. (2003). The Human Genome Project: Lessons from Large-Scale Biology. *Science*, 300(5617):286–290.

[Coluzzi, 2002] Coluzzi, M. (2002). A Polytene Chromosome Analysis of the Anopheles gambiae Species Complex. *Science*, 298(5597):1415–1418.

[Conway and Bromage, 2011] Conway, T. C. and Bromage, A. J. (2011). Succinct data structures for assembling large genomes. *Bioinformatics*, 27(4):479–486.

[Cowman et al., 2012] Cowman, A. F., Berry, D., and Baum, J. (2012). The cellular and molecular basis for malaria parasite invasion of the human red blood cell. *Journal of Cell Biology*, 198(6):961–971.

[Cox, 2010] Cox, F. E. (2010). History of the discovery of the malaria parasites and their vectors. *Parasites and Vectors*, 3(1):5.

[Crosnier et al., 2016] Crosnier, C., Iqbal, Z., Knuepfer, E., Maciuca, S., Perrin, A. J., Kamuyu, G., Goulding, D., Bustamante, L. Y., Miles, A., Moore, S. C., Dougan, G., Holder, A. A., Kwiatkowski, D. P., Rayner, J. C., Pleass, R. J., and Wright, G. J. (2016). Binding of plasmodium falciparum merozoite surface proteins DBLMSP and DBLMSP2 to human immunoglobulin M Is conserved among broadly diverged sequence variants. *Journal of Biological Chemistry*, 291(27):14285–14299.

[Daher et al., 2010] Daher, L. J., Demanga, C. G., Prieur, E., Pérignon, J. L., Bouharoun-Tayoun, H., and Druilhe, P. (2010). Toward the rational design of a malaria vaccine construct using the MSP3 family as an example: Contribution of immunogenicity studies in models. *Infection and Immunity*, 78(1):477–485.

[DePristo et al., 2011] DePristo, M. a., Banks, E., Poplin, R., Garimella, K. V., Maguire, J. R., Hartl, C., Philippakis, A. a., del Angel, G., Rivas, M. a., Hanna, M., McKenna, A., Fennell, T. J., Kernytsky, A. M., Sivachenko, A. Y., Cibulskis, K., Gabriel, S. B., Altshuler, D., and Daly, M. J. (2011). A framework for variation discovery and genotyping using next-generation DNA sequencing data. *Nature genetics*, 43(5):491–498.

[Didelot et al., 2012] Didelot, X., Bowden, R., Wilson, D. J., Peto, T. E. A., and Crook, D. W. (2012). Transforming clinical microbiology with bacterial genome sequencing. *Nature Reviews Genetics*, 13(9):601–612.

[Dilthey et al., 2015] Dilthey, A., Cox, C., Iqbal, Z., Nelson, M. R., and McVean, G. (2015). Improved genome inference in the MHC using a population reference graph. *Nature Genetics*, 47(6):682–688.

[Doolan et al., 2009] Doolan, D. L., Dobaño, C., and Baird, J. K. (2009). Acquired immunity to Malaria.

[Durbin et al., 2010] Durbin, R. M., Altshuler, D. L., Durbin, R. M., Abecasis, G. R., Bentley, D. R., Chakravarti, A., Clark, A. G., Collins, F. S., De La Vega, F. M., Donnelly, P., Egholm, M., Flicek, P., Gabriel, S. B., Gibbs, R. A., Knoppers, B. M., Lander, E. S., Lehrach, H., Mardis, E. R., McVean, G. A., Nickerson, D. A., Peltonen, L., Schafer, A. J., Sherry, S. T., Wang, J., Wilson, R. K., Gibbs, R. A., Deiros, D., Metzker, M., Muzny, D., Reid, J., Wheeler, D., Wang, J., Li, J., Jian, M., Li, G., Li, R., Liang, H., Tian, G., Wang, B., Wang, J., Wang, W., Yang,

H., Zhang, X., Zheng, H., Lander, E. S., Altshuler, D. L., Ambrogio, L., Bloom, T., Cibulskis, K., Fennell, T. J., Gabriel, S. B., Jaffe, D. B., Shefler, E., Sougnez, C. L., Bentley, D. R., Gormley, N., Humphray, S., Kingsbury, Z., Koko-Gonzales, P., Stone, J., McKernan, K. J., Costa, G. L., Ichikawa, J. K., Lee, C. C., Sudbrak, R., Lehrach, H., Borodina, T. A., Dahl, A., Davydov, A. N., Marquardt, P., Mertes, F., Nietfeld, W., Rosenstiel, P., Schreiber, S., Soldatov, A. V., Timmermann, B., Tolzmann, M., Egholm, M., Affourtit, J., Ashworth, D., Attiya, S., Bachorski, M., Buglione, E., Burke, A., Caprio, A., Celone, C., Clark, S., Conners, D., Desany, B., Gu, L., Guccione, L., Kao, K., Kebbel, A., Knowlton, J., Labrecque, M., McDade, L., Mealmaker, C., Minderman, M., Nawrocki, A., Niazi, F., Pareja, K., Ramenani, R., Riches, D., Song, W., Turcotte, C., Wang, S., Mardis, E. R., Wilson, R. K., Dooling, D., Fulton, L., Fulton, R., Weinstock, G., Durbin, R. M., Burton, J., Carter, D. M., Churcher, C., Coffey, A., Cox, A., Palotie, A., Quail, M., Skelly, T., Stalker, J., Swerdlow, H. P., Turner, D., De Witte, A., Giles, S., Gibbs, R. A., Wheeler, D., Bainbridge, M., Challis, D., Sabo, A., Yu, F., Yu, J., Wang, J., Fang, X., Guo, X., Li, R., Li, Y., Luo, R., Tai, S., Wu, H., Zheng, H., Zheng, X., Zhou, Y., Li, G., Wang, J., Yang, H., Marth, G. T., Garrison, E. P., Huang, W., Indap, A., Kural, D., Lee, W.-P., Fung Leong, W., Quinlan, A. R., Stewart, C., Stromberg, M. P., Ward, A. N., Wu, J., Lee, C., Mills, R. E., Shi, X., Daly, M. J., DePristo, M. A., Altshuler, D. L., Ball, A. D., Banks, E., Bloom, T., Browning, B. L., Cibulskis, K., Fennell, T. J., Garimella, K. V., Grossman, S. R., Handsaker, R. E., Hanna, M., Hartl, C., Jaffe, D. B., Kernytsky, A. M., Korn, J. M., Li, H., Maguire, J. R., McCarroll, S. A., McKenna, A., Nemesh, J. C., Philippakis, A. A., Poplin, R. E., Price, A., Rivas, M. A., Sabeti, P. C., Schaffner, S. F., Shefler, E., Shlyakhter, I. A., Cooper, D. N., Ball, E. V., Mort, M., Phillips, A. D., Stenson, P. D., Sebat, J., Makarov, V., Ye, K., Yoon, S. C., Bustamante, C. D., Clark, A. G., Boyko, A., Degenhardt, J., Gravel, S., Gutenkunst, R. N.,

Kaganovich, M., Keinan, A., Lacroute, P., Ma, X., Reynolds, A., Clarke, L., Flicek, P., Cunningham, F., Herrero, J., Keenen, S., Kulesha, E., Leinonen, R., McLaren, W. M., Radhakrishnan, R., Smith, R. E., Zalunin, V., Zheng-Bradley, X., Korbel, J. O., Stütz, A. M., Humphray, S., Bauer, M., Keira Cheetham, R., Cox, T., Eberle, M., James, T., Kahn, S., Murray, L., Chakravarti, A., Ye, K., De La Vega, F. M., Fu, Y., Hyland, F. C. L., Manning, J. M., McLaughlin, S. F., Peckham, H. E., Sakarya, O., Sun, Y. A., Tsung, E. F., Batzer, M. A., Konkel, M. K., Walker, J. A., Sudbrak, R., Albrecht, M. W., Amstislavskiy, V. S., Herwig, R., Parkhomchuk, D. V., Sherry, S. T., Agarwala, R., Khouri, H. M., Morgulis, A. O., Paschall, J. E., Phan, L. D., Rotmistrovsky, K. E., Sanders, R. D., Shumway, M. F., Xiao, C., McVean, G. A., Auton, A., Iqbal, Z., Lunter, G., Marchini, J. L., Moutsianas, L., Myers, S., Tumian, A., Desany, B., Knight, J., Winer, R., Craig, D. W., Beckstrom-Sternberg, S. M., Christoforides, A., Kurdoglu, A. A., Pearson, J. V., Sinari, S. A., Tembe, W. D., Haussler, D., Hinrichs, A. S., Katzman, S. J., Kern, A., Kuhn, R. M., Przeworski, M., Hernandez, R. D., Howie, B., Kelley, J. L., Cord Melton, S., Abecasis, G. R., Li, Y., Anderson, P., Blackwell, T., Chen, W., Cookson, W. O., Ding, J., Min Kang, H., Lathrop, M., Liang, L., Moffatt, M. F., Scheet, P., Sidore, C., Snyder, M., Zhan, X., Zöllner, S., Awadalla, P., Casals, F., Idaghdour, Y., Keebler, J., Stone, E. A., Zilversmit, M., Jorde, L., Xing, J., Eichler, E. E., Aksay, G., Alkan, C., Hajirasouliha, I., Hormozdiari, F., Kidd, J. M., Cenk Sahinalp, S., Sudmant, P. H., Mardis, E. R., Chen, K., Chinwalla, A., Ding, L., Koboldt, D. C., McLellan, M. D., Dooling, D., Weinstock, G., Wallis, J. W., Wendl, M. C., Zhang, Q., Durbin, R. M., Albers, C. A., Ayub, Q., Balasubramaniam, S., Barrett, J. C., Carter, D. M., Chen, Y., Conrad, D. F., Danecek, P., Dermitzakis, E. T., Hu, M., Huang, N., Hurles, M. E., Jin, H., Jostins, L., Keane, T. M., Quang Le, S., Lindsay, S., Long, Q., MacArthur, D. G., Montgomery, S. B., Parts, L., Stalker, J., Tyler-Smith, C., Walter, K., Zhang, Y., Gerstein, M. B., Snyder, M.,

Abyzov, A., Balasubramanian, S., Bjornson, R., Du, J., Grubert, F., Habegger, L., Haraksingh, R., Jee, J., Khurana, E., Lam, H. Y. K., Leng, J., Jasmine Mu, X., Urban, A. E., Zhang, Z., Li, Y., Luo, R., Marth, G. T., Garrison, E. P., Kural, D., Quinlan, A. R., Stewart, C., Stromberg, M. P., Ward, A. N., Wu, J., Lee, C., Mills, R. E., Shi, X., McCarroll, S. A., Banks, E., DePristo, M. A., Handsaker, R. E., Hartl, C., Korn, J. M., Li, H., Nemesh, J. C., Sebat, J., Makarov, V., Ye, K., Yoon, S. C., Degenhardt, J., Kaganovich, M., Clarke, L., Smith, R. E., Zheng-Bradley, X., Korbel, J. O., Humphray, S., Keira Cheetham, R., Eberle, M., Kahn, S., Murray, L., Ye, K., De La Vega, F. M., Fu, Y., Peckham, H. E., Sun, Y. A., Batzer, M. A., Konkel, M. K., Walker, J. A., Xiao, C., Iqbal, Z., Desany, B., Blackwell, T., Snyder, M., Xing, J., Eichler, E. E., Aksay, G., Alkan, C., Hajirasouliha, I., Hormozdiari, F., Kidd, J. M., Chen, K., Chinwalla, A., Ding, L., McLellan, M. D., Wallis, J. W., Hurles, M. E., Conrad, D. F., Walter, K., Zhang, Y., Gerstein, M. B., Snyder, M., Abyzov, A., Du, J., Grubert, F., Haraksingh, R., Jee, J., Khurana, E., Lam, H. Y. K., Leng, J., Jasmine Mu, X., Urban, A. E., Zhang, Z., Gibbs, R. A., Bainbridge, M., Challis, D., Coafra, C., Dinh, H., Kovar, C., Lee, S., Muzny, D., Nazareth, L., Reid, J., Sabo, A., Yu, F., Yu, J., Marth, G. T., Garrison, E. P., Indap, A., Fung Leong, W., Quinlan, A. R., Stewart, C., Ward, A. N., Wu, J., Cibulskis, K., Fennell, T. J., Gabriel, S. B., Garimella, K. V., Hartl, C., Shefler, E., Sougnez, C. L., Wilkinson, J., Clark, A. G., Gravel, S., Grubert, F., Clarke, L., Flicek, P., Smith, R. E., Zheng-Bradley, X., Sherry, S. T., Khouri, H. M., Paschall, J. E., Shumway, M. F., Xiao, C., McVean, G. A., Katzman, S. J., Abecasis, G. R., Blackwell, T., Mardis, E. R., Dooling, D., Fulton, L., Fulton, R., Koboldt, D. C., Durbin, R. M., Balasubramaniam, S., Coffey, A., Keane, T. M., MacArthur, D. G., Palotie, A., Scott, C., Stalker, J., Tyler-Smith, C., Gerstein, M. B., Balasubramanian, S., Chakravarti, A., Knoppers, B. M., Abecasis, G. R., Bustamante, C. D., Gharani, N., Gibbs, R. A., Jorde, L., Kaye, J. S., Kent, A.,

Li, T., McGuire, A. L., McVean, G. A., Ossorio, P. N., Rotimi, C. N., Su, Y., Toji, L. H., Tyler-Smith, C., Brooks, L. D., Felsenfeld, A. L., McEwen, J. E., Abdallah, A., Juenger, C. R., Clemm, N. C., Collins, F. S., Duncanson, A., Green, E. D., Guyer, M. S., Peterson, J. L., Schafer, A. J., Abecasis, G. R., Altshuler, D. L., Auton, A., Brooks, L. D., Durbin, R. M., Gibbs, R. A., Hurles, M. E., and McVean, G. A. (2010). A map of human genome variation from population-scale sequencing. *Nature*, 467(7319):1061–1073.

[Dutilh et al., 2014] Dutilh, B. E., Thompson, C. C., Vicente, A. C., Marin, M. A., Lee, C., Silva, G. G., Schmieder, R., Andrade, B. G., Chimetto, L., Cuevas, D., Garza, D. R., Okeke, I. N., Aboderin, A., Spangler, J., Ross, T., Dinsdale, E. A., Thompson, F. L., Harkins, T. T., and Edwards, R. A. (2014). Comparative genomics of 274 Vibrio cholerae genomes reveals mobile functions structuring three niche dimensions. *BMC Genomics*, 15(1):654.

[Eggertsson et al., 2017] Eggertsson, H. P., Jonsson, H., Kristmundsdottir, S., Hjartarson, E., Kehr, B., Masson, G., Zink, F., Hjorleifsson, K. E., Jonasdottir, A., Jonasdottir, A., Jonsdottir, I., Gudbjartsson, D. F., Melsted, P., Stefansson, K., and Halldorsson, B. V. (2017). Graphtyper enables population-scale genotyping using pangenome graphs. *Nature Genetics*, 49(11):1654–1660.

[Evans and Amyes, 2014] Evans, B. A. and Amyes, S. G. B. (2014). OXA β-lactamases. *Clinical microbiology reviews*, 27(2):241–63.

[Ferragina and Manzini, 2000] Ferragina, P. and Manzini, G. (2000). Opportunistic data structures with applications. In *Proceeding FOCS '00 Proceedings of the 41st Annual Symposium on Foundations of Computer ScienceProceeding FOCS '00 Proceedings of the 41st Annual Symposium on Foundations of Computer Science*, pages 390–398. IEEE Comput. Soc.

[Ferreira and Hartl, 2007] Ferreira, M. U. and Hartl, D. L. (2007). Plasmodium falciparum: Worldwide sequence diversity and evolution of the malaria vaccine candidate merozoite surface protein-2 (MSP-2). *Experimental Parasitology*, 115(1):32–40.

[Feuk et al., 2006] Feuk, L., Carson, A. R., and Scherer, S. W. (2006). Structural variation in the human genome. *Nature Reviews Genetics*, 7(2):85–97.

[Fonseca et al., 2012] Fonseca, N. A., Rung, J., Brazma, A., and Marioni, J. C. (2012). Tools for mapping high-throughput sequencing data.

[Fowkes et al., 2010] Fowkes, F. J. I., Richards, J. S., Simpson, J. A., and Beeson, J. G. (2010). The relationship between anti-merozoite antibodies and incidence of Plasmodium falciparum malaria: A systematic review and meta-analysis. *PLoS Medicine*, 7(1).

[Gardner et al., 2002] Gardner, M. J., Hall, N., Fung, E., White, O., Berriman, M., Hyman, R. W., Carlton, J. M., Pain, A., Nelson, K. E., Bowman, S., Paulsen, I. T., James, K., Eisen, J. A., Rutherford, K., Salzberg, S. L., Craig, A., Kyes, S., Chan, M.-S., Nene, V., Shallom, S. J., Suh, B., Peterson, J., Angiuoli, S., Pertea, M., Allen, J., Selengut, J., Haft, D., Mather, M. W., Vaidya, A. B., Martin, D. M. A., Fairlamb, A. H., Fraunholz, M. J., Roos, D. S., Ralph, S. A., McFadden, G. I., Cummings, L. M., Subramanian, G. M., Mungall, C., Venter, J. C., Carucci, D. J., Hoffman, S. L., Newbold, C., Davis, R. W., Fraser, C. M., and Barrell, B. (2002). Genome sequence of the human malaria parasite Plasmodium falciparum. *Nature*, 419(6906):498–511.

[Garrison and Marth, 2012] Garrison, E. and Marth, G. (2012). Haplotype-based variant detection from short-read sequencing. *ArXiv e-prints*.

[Genton et al., 2002] Genton, B., Betuela, I., Felger, I., AlYaman, F., Anders, R. F., Saul, A., Rare, L., Baisor, M., Lorry, K., Brown, G. V., Pye, D., Irving, D. O., Smith, T. A., Beck, H., and Alpers, M. P. (2002). A Recombinant BloodStage Malaria Vaccine Reduces Plasmodium falciparum Density and Exerts Selective Pressure on Parasite Populations in a Phase 12b Trial in Papua New Guinea. *The Journal of Infectious Diseases*, 185(6):820–827.

[Gnerre et al., 2011] Gnerre, S., MacCallum, I., Przybylski, D., Ribeiro, F. J., Burton, J. N., Walker, B. J., Sharpe, T., Hall, G., Shea, T. P., Sykes, S., Berlin, A. M., Aird, D., Costello, M., Daza, R., Williams, L., Nicol, R., Gnirke, A., Nusbaum, C., Lander, E. S., and Jaffe, D. B. (2011). High-quality draft assemblies of mammalian genomes from massively parallel sequence data. *Proceedings of the National Academy of Sciences*, 108(4):1513–1518.

[Gog et al., 2014] Gog, S., Beller, T., Moffat, A., and Petri, M. (2014). From Theory to Practice: Plug and Play with Succinct Data Structures. In *Proceedings of the 13th International Symposium on Experimental Algorithms - Volume 8504*, pages 326–337. Springer-Verlag New York, Inc.

[Gomes et al., 2016] Gomes, P. S., Bhardwaj, J., Rivera-Correa, J., Freire-De-Lima, C. G., and Morrot, A. (2016). Immune escape strategies of malaria parasites.

[Grossi et al., 2003] Grossi, R., Gupta, A., and Vitter, J. S. (2003). High-order entropy-compressed text indexes. *SODA '03 Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms*, 39(1):841–850.

[Guan and Sung, 2016] Guan, P. and Sung, W.-K. (2016). Structural variation detection using next-generation sequencing data. *Methods*, 102:36–49.

[Hartl, 2004] Hartl, D. L. (2004). The origin of malaria: mixed messages from genetic diversity. *Nature Reviews MicroBiology*, 2(1):15–22.

[Hartl et al., 2002] Hartl, D. L., Nielsen, K. M., Volkman, S. K., Wirth, D. F., Barry, A. E., Day, K. P., and Winzeler, E. A. (2002). The paradoxical population genetics of Plasmodium falciparum. *TRENDS in Parasitology*, 18(6).

[Hatem et al., 2013] Hatem, A., Bozda, D., Toland, A. E., and Çatalyürek, Ü. V. (2013). Benchmarking short sequence mapping tools. *BMC Bioinformatics*, 14(1):184.

[Higgins and Sharp, 1988] Higgins, D. G. and Sharp, P. M. (1988). CLUSTAL: a package for performing multiple sequence alignment on a microcomputer. *Gene*, 73(1):237–44.

[Hodder et al., 2012] Hodder, A. N., Czabotar, P. E., Uboldi, A. D., Clarke, O. B., Lin, C. S., Healer, J., Smith, B. J., and Cowman, A. F. (2012). Insights into duffy binding-like domains through the crystal structure and function of the merozoite surface protein MSPDBL2 from Plasmodium falciparum. *Journal of Biological Chemistry*, 287(39):32922–32939.

[Horton et al., 2008] Horton, R., Gibson, R., Coggill, P., Miretti, M., Allcock, R. J., Almeida, J., Forbes, S., Gilbert, J. G. R., Halls, K., Harrow, J. L., Hart, E., Howe, K., Jackson, D. K., Palmer, S., Roberts, A. N., Sims, S., Stewart, C. A., Traherne, J. A., Trevanion, S., Wilming, L., Rogers, J., de Jong, P. J., Elliott, J. F., Sawcer, S., Todd, J. A., Trowsdale, J., and Beck, S. (2008). Variation analysis and gene annotation of eight MHC haplotypes: The MHC Haplotype Project. *Immunogenetics*, 60(1):1–18.

[Huang et al., 2013] Huang, L., Popic, V., and Batzoglou, S. (2013). Short read alignment with populations of genomes. *Bioinformatics (Oxford, England)*, 29(13):i361–70.

[Hughes, 2002] Hughes, A. L. (2002). Natural selection and the diversification of vertebrate immune effectors. *Immunological Reviews*, 190(1):161–168.

[Iqbal et al., 2012] Iqbal, Z., Caccamo, M., Turner, I., Flicek, P., and McVean, G. (2012). De novo assembly and genotyping of variants using colored de Bruijn graphs. *Nature genetics*, 44(2):226–32.

[Jacoby and Munoz-Price, 2005] Jacoby, G. A. and Munoz-Price, L. S. (2005). The New β-Lactamases. *New England Journal of Medicine*, 352(4):380–391.

[Jiang et al., 2012] Jiang, W., Johnson, C., Jayaraman, J., Simecek, N., Noble, J., Moffatt, M. F., Cookson, W. O., Trowsdale, J., and Traherne, J. A. (2012). Copy number variation leads to considerable diversity for B but not A haplotypes of the human KIR genes encoding NK cell receptors. *Genome Research*, 22(10):1845–1854.

[Kelleher et al., 2013] Kelleher, J., Ness, R. W., and Halligan, D. L. (2013). Processing genome scale tabular data with wormtable. *BMC bioinformatics*, 14(1):356.

[Ko et al., 2011] Ko, W. Y., Kaercher, K. A., Giombini, E., Marcatili, P., Froment, A., Ibrahim, M., Lema, G., Nyambo, T. B., Omar, S. A., Wambebe, C., Ranciaro, A., Hirbo, J. B., and Tishkoff, S. A. (2011). Effects of natural selection and gene conversion on the evolution of human glycophorins coding for MNS blood polymorphisms in malaria-endemic African populations. *American Journal of Human Genetics*, 88(6):741–754.

[Kraemer and Smith, 2006] Kraemer, S. M. and Smith, J. D. (2006). A family affair: var genes, PfEMP1 binding, and malaria disease.

[Lander et al., 2001] Lander, E. S., Linton, L. M., Birren, B., Nusbaum, C., Zody, M. C., Baldwin, J., Devon, K., Dewar, K., Doyle, M., FitzHugh, W., Funke, R.,

Gage, D., Harris, K., Heaford, A., Howland, J., Kann, L., Lehoczky, J., LeVine, R., McEwan, P., McKernan, K., Meldrim, J., Mesirov, J. P., Miranda, C., Morris, W., Naylor, J., Raymond, C., Rosetti, M., Santos, R., Sheridan, A., Sougnez, C., Stange-Thomann, N., Stojanovic, N., Subramanian, A., Wyman, D., Rogers, J., Sulston, J., Ainscough, R., Beck, S., Bentley, D., Burton, J., Clee, C., Carter, N., Coulson, A., Deadman, R., Deloukas, P., Dunham, A., Dunham, I., Durbin, R., French, L., Grafham, D., Gregory, S., Hubbard, T., Humphray, S., Hunt, A., Jones, M., Lloyd, C., McMurray, A., Matthews, L., Mercer, S., Milne, S., Mullikin, J. C., Mungall, A., Plumb, R., Ross, M., Shownkeen, R., Sims, S., Waterston, R. H., Wilson, R. K., Hillier, L. W., McPherson, J. D., Marra, M. A., Mardis, E. R., Fulton, L. A., Chinwalla, A. T., Pepin, K. H., Gish, W. R., Chissoe, S. L., Wendl, M. C., Delehaunty, K. D., Miner, T. L., Delehaunty, A., Kramer, J. B., Cook, L. L., Fulton, R. S., Johnson, D. L., Minx, P. J., Clifton, S. W., Hawkins, T., Branscomb, E., Predki, P., Richardson, P., Wenning, S., Slezak, T., Doggett, N., Cheng, J.-F., Olsen, A., Lucas, S., Elkin, C., Uberbacher, E., Frazier, M., Gibbs, R. A., Muzny, D. M., Scherer, S. E., Bouck, J. B., Sodergren, E. J., Worley, K. C., Rives, C. M., Gorrell, J. H., Metzker, M. L., Naylor, S. L., Kucherlapati, R. S., Nelson, D. L., Weinstock, G. M., Sakaki, Y., Fujiyama, A., Hattori, M., Yada, T., Toyoda, A., Itoh, T., Kawagoe, C., Watanabe, H., Totoki, Y., Taylor, T., Weissenbach, J., Heilig, R., Saurin, W., Artiguenave, F., Brottier, P., Bruls, T., Pelletier, E., Robert, C., Wincker, P., Rosenthal, A., Platzer, M., Nyakatura, G., Taudien, S., Rump, A., Smith, D. R., Doucette-Stamm, L., Rubenfield, M., Weinstock, K., Lee, H. M., Dubois, J., Yang, H., Yu, J., Wang, J., Huang, G., Gu, J., Hood, L., Rowen, L., Madan, A., Qin, S., Davis, R. W., Federspiel, N. A., Abola, A. P., Proctor, M. J., Roe, B. A., Chen, F., Pan, H., Ramser, J., Lehrach, H., Reinhardt, R., McCombie, W. R., de la Bastide, M., Dedhia, N., Blöcker, H., Hornischer, K., Nordsiek, G., Agarwala, R., Aravind, L., Bailey, J. A., Bateman, A., Batzoglou, S.,

Birney, E., Bork, P., Brown, D. G., Burge, C. B., Cerutti, L., Chen, H.-C., Church, D., Clamp, M., Copley, R. R., Doerks, T., Eddy, S. R., Eichler, E. E., Furey, T. S., Galagan, J., Gilbert, J. G. R., Harmon, C., Hayashizaki, Y., Haussler, D., Hermjakob, H., Hokamp, K., Jang, W., Johnson, L. S., Jones, T. A., Kasif, S., Kaspryzk, A., Kennedy, S., Kent, W. J., Kitts, P., Koonin, E. V., Korf, I., Kulp, D., Lancet, D., Lowe, T. M., McLysaght, A., Mikkelsen, T., Moran, J. V., Mulder, N., Pollara, V. J., Ponting, C. P., Schuler, G., Schultz, J., Slater, G., Smit, A. F. A., Stupka, E., Szustakowki, J., Thierry-Mieg, D., Thierry-Mieg, J., Wagner, L., Wallis, J., Wheeler, R., Williams, A., Wolf, Y. I., Wolfe, K. H., Yang, S.-P., Yeh, R.-F., Collins, F., Guyer, M. S., Peterson, J., Felsenfeld, A., Wetterstrand, K. A., Myers, R. M., Schmutz, J., Dickson, M., Grimwood, J., Cox, D. R., Olson, M. V., Kaul, R., Raymond, C., Shimizu, N., Kawasaki, K., Minoshima, S., Evans, G. A., Athanasiou, M., Schultz, R., Patrinos, A., Morgan, M. J., de Jong, P., Catanese, J. J., Osoegawa, K., Shizuya, H., Choi, S., Chen, Y. J., Szustakowki, J., and International Human Genome Sequencing Consortium (2001). Initial sequencing and analysis of the human genome. *Nature*, 409(6822):860–921.

[Langmead and Salzberg, 2012] Langmead, B. and Salzberg, S. L. (2012). Fast gapped-read alignment with Bowtie 2. *Nature methods*, 9(4):357–9.

[Langmead et al., 2009] Langmead, B., Trapnell, C., Pop, M., and Salzberg, S. L. (2009). Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome biology*, 10(3):R25.

[Lapierre and Gogarten, 2009] Lapierre, P. and Gogarten, J. P. (2009). Estimating the size of the bacterial pan-genome.

[Lechler and Warrens, 2000] Lechler, R. and Warrens, A. N. (2000). *HLA in health and disease*. Academic Press.

[Lee et al., 2002] Lee, C., Grasso, C., and Sharlow, M. F. (2002). Multiple sequence alignment using partial order graphs. *Bioinformatics*, 18(3):452–464.

[Leffler et al., 2017] Leffler, E. M., Band, G., Busby, G. B. J., Kivinen, K., Le, Q. S., Clarke, G. M., Bojang, K. A., Conway, D. J., Jallow, M., Sisay-Joof, F., Bougouma, E. C., Mangano, V. D., Modiano, D., Sirima, S. B., Achidi, E., Apinjoh, T. O., Marsh, K., Ndila, C. M., Peshu, N., Williams, T. N., Drakeley, C., Manjurano, A., Reyburn, H., Riley, E., Kachala, D., Molyneux, M., Nyirongo, V., Taylor, T., Thornton, N., Tilley, L., Grimsley, S., Drury, E., Stalker, J., Cornelius, V., Hubbart, C., Jeffreys, A. E., Rowlands, K., Rockett, K. A., Spencer, C. C. A., Kwiatkowski, D. P., and Malaria Genomic Epidemiology Network, M. G. E. (2017). Resistance to malaria through structural variation of red blood cell invasion receptors. *Science (New York, N.Y.)*, 356(6343):eaam6393.

[Leffler et al., 2013] Leffler, E. M., Gao, Z., Pfeifer, S., Segurel, L., Auton, A., Venn, O., Bowden, R., Bontrop, R., Wall, J. D., Sella, G., Donnelly, P., McVean, G., and Przeworski, M. (2013). Multiple Instances of Ancient Balancing Selection Shared Between Humans and Chimpanzees. *Science*, 339(6127):1578–1582.

[Li, 2011] Li, H. (2011). A statistical framework for SNP calling, mutation discovery, association mapping and population genetical parameter estimation from sequencing data. *Bioinformatics*, 27(21):2987–2993.

[Li, 2012] Li, H. (2012). Exploring single-sample SNP and INDEL calling with whole-genome de novo assembly. *Bioinformatics (Oxford, England)*, 28(14):1838–44.

[Li, 2013] Li, H. (2013). Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM. *ArXiv e-prints*.

[Li and Durbin, 2009] Li, H. and Durbin, R. (2009). Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics (Oxford, England)*, 25(14):1754–60.

[Li and Durbin, 2010] Li, H. and Durbin, R. (2010). Fast and accurate long-read alignment with Burrows-Wheeler transform. *Bioinformatics (Oxford, England)*, 26(5):589–95.

[Li et al., 2009] Li, H., Handsaker, B., Wysoker, A., Fennell, T., Ruan, J., Homer, N., Marth, G., Abecasis, G., and Durbin, R. (2009). The Sequence Alignment/Map format and SAMtools. *Bioinformatics*, 25(16):2078–2079.

[Li and Homer, 2010] Li, H. and Homer, N. (2010). A survey of sequence alignment algorithms for next-generation sequencing. *Briefings in bioinformatics*, 11(5):473–83.

[Li et al., 2008] Li, H., Ruan, J., and Durbin, R. (2008). Mapping short DNA sequencing reads and calling variants using mapping quality scores. *Genome research*, 18(11):1851–8.

[Li et al., 2010] Li, R., Zhu, H., Ruan, J., Qian, W., Fang, X., Shi, Z., Li, Y., Li, S., Shan, G., Kristiansen, K., Li, S., Yang, H., Wang, J., and Wang, J. (2010). De novo assembly of human genomes with massively parallel short read sequencing. *Genome research*, 20(2):265–72.

[Limasset et al., 2017] Limasset, A., Rizk, G., Chikhi, R., and Peterlongo, P. (2017). Fast and scalable minimal perfect hashing for massive key sets. *ArXiv e-prints*.

[Liu et al., 2012] Liu, P., Carvalho, C. M. B., Hastings, P. J., and Lupski, J. R. (2012). Mechanisms for recurrent and complex human genomic rearrangements. *Current opinion in genetics & development*, 22(3):211–20.

[Liu et al., 2014] Liu, Y., Popp, B., and Schmidt, B. (2014). CUSHAW3: Sensitive and accurate base-space and color-space short-read alignment with hybrid seeding. *PLoS ONE*, 9(1):e86869.

[Lopez-Rubio et al., 2009] Lopez-Rubio, J. J., Mancio-Silva, L., and Scherf, A. (2009). Genome-wide Analysis of Heterochromatin Associates Clonally Variant Gene Regulation with Perinuclear Repressive Centers in Malaria Parasites. *Cell Host and Microbe*, 5(2):179–190.

[Lupski and Stankiewicz, 2005] Lupski, J. R. and Stankiewicz, P. (2005). Genomic Disorders: Molecular Mechanisms for Rearrangements and Conveyed Phenotypes. *PLoS Genetics*, 1(6):e49.

[Maciuca et al., 2016] Maciuca, S., Elias, C. D. O., McVean, G., and Iqbal, Z. (2016). A natural encoding of genetic variation in a burrows-wheeler transform to enable mapping and genome inference. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 9838, pages 222–233. Springer, Cham.

[Maier et al., 2003] Maier, A. G., Duraisingh, M. T., Reeder, J. C., Patel, S. S., Kazura, J. W., Zimmerman, P. A., and Cowman, A. F. (2003). Plasmodium falciparum erythrocyte invasion through glycophorin C and selection for Gerbich negativity in human populations. *Nature medicine*, 9(1):87–92.

[Mäkinen et al., 2015] Mäkinen, V., Belazzougui, D., Cunial, F., and Tomescu, A. I. (2015). *Genome-scale algorithm design: Biological sequence analysis in the era of high-throughput sequencing.*

[McColl and Anders, 1997] McColl, D. J. and Anders, R. F. (1997). Conservation of structural motifs and antigenic diversity in the Plasmodium falciparum merozoite surface protein-3 (MSP-3). *Molecular and Biochemical Parasitology*, 90(1):21–31.

[Miles et al., 2016] Miles, A., Iqbal, Z., Vauterin, P., Pearson, R., Campino, S., Theron, M., Gould, K., Mead, D., Drury, E., O'Brien, J., Rubio, V. R., Macinnis, B., Mwangi, J., Samarakoon, U., Ranford-Cartwright, L., Ferdig, M., Hayton, K., Su, X. Z., Wellems, T., Rayner, J., McVean, G., and Kwiatkowski, D. (2016). Indels, structural variation, and recombination drive genomic diversity in Plasmodium falciparum. *Genome Research*, 26(9):1288–1299.

[Miller et al., 1993] Miller, L. H., Roberts, T., Shahabuddin, M., and McCutchan, T. F. (1993). Analysis of sequence diversity in the Plasmodium falciparum merozoite surface protein-1 (MSP-1). *Molecular and biochemical parasitology*, 59(1):1–14.

[Myers, 2005] Myers, E. W. (2005). The fragment assembly string graph. *Bioinformatics*, 21(SUPPL. 2):ii79–ii85.

[Nguyen et al., 2014] Nguyen, N., Hickey, G., Raney, B. J., Armstrong, J., Clawson, H., Zweig, A., Karolchik, D., Kent, W. J., Haussler, D., and Paten, B. (2014). Comparative assembly hubs: Web-accessible browsers for comparative genomics. *Bioinformatics*, 30(23):3293–3301.

[Nielsen et al., 2003] Nielsen, K. M., Kasper, J., Choi, M., Bedford, T., Kristiansen, K., Wirth, D. F., Volkman, S. K., Lozovsky, E. R., and Hartl, D. L. (2003). Gene conversion as a source of nucleotide diversity in Plasmodium falciparum. *Molecular Biology and Evolution*, 20(5):726–734.

[Nielsen et al., 2011] Nielsen, R., Paul, J. S., Albrechtsen, A., and Song, Y. S. (2011). Genotype and SNP calling from next-generation sequencing data. *Nature Reviews Genetics*, 12(6):443–451.

[Novak et al., 2017a] Novak, A. M., Garrison, E., and Paten, B. (2017a). A graph extension of the positional Burrows-Wheeler transform and its applications. *Algorithms for molecular biology : AMB*, 12:18.

[Novak et al., 2017b] Novak, A. M., Hickey, G., Garrison, E., Blum, S., Connelly, A., Dilthey, A., Eizenga, J., Elmohamed, M. A. S., Guthrie, S., Kahles, A., Keenan, S., Kelleher, J., Kural, D., Li, H., Lin, M. F., Miga, K., Ouyang, N., Rakocevic, G., Smuga-Otto, M., Zaranek, A. W., Durbin, R., McVean, G., Haussler, D., and Paten, B. (2017b). Genome Graphs. *bioRxiv*.

[Novak et al., 2015] Novak, A. M., Rosen, Y., Haussler, D., and Paten, B. (2015). Canonical, stable, general mapping using context schemes. *Bioinformatics*, 31(22):3569–3576.

[Ochola et al., 2010] Ochola, L. I., Tetteh, K. K. A., Stewart, L. B., Riitho, V., Marsh, K., and Conway, D. J. (2010). Allele frequency-based and polymorphism-versus-divergence indices of balancing selection in a new filtered set of polymorphic genes in Plasmodium falciparum. *Molecular biology and evolution*, 27(10):2344–51.

[Orlandi et al., 1992] Orlandi, P. A., Klotz, F. W., and Haynes, J. D. (1992). A malaria invasion receptor, the 175-kilodalton erythrocyte binding antigen of Plasmodium falciparum recognizes the terminal Neu5Ac(alpha 2-3)Gal- sequences of glycophorin A. *The Journal of Cell Biology*, 116(4).

[Osier et al., 2014] Osier, F. H., Mackinnon, M. J., Crosnier, C., Fegan, G., Kamuyu, G., Wanaguru, M., Ogada, E., McDade, B., Rayner, J. C., Wright, G. J., and Marsh, K. (2014). New antigens for a multicomponent blood-stage malaria vaccine. *Science Translational Medicine*, 6(247):247ra102–247ra102.

[Ouattara et al., 2015] Ouattara, A., Barry, A. E., Dutta, S., Remarque, E. J., Beeson, J. G., and Plowe, C. V. (2015). Designing malaria vaccines to circumvent antigen variability. *Vaccine*, 33(52):7506–7512.

[Ouattara et al., 2013] Ouattara, A., Takala-Harrison, S., Thera, M. A., Coulibaly, D., Niangaly, A., Saye, R., Tolo, Y., Dutta, S., Gray Heppner, D., Soisson, L., Diggs, C. L., Vekemans, J., Cohen, J., Blackwelder, W. C., Dube, T., Laurens, M. B., Doumbo, O. K., and Plowe, C. V. (2013). Molecular basis of allele-specific efficacy of a blood-stage malaria vaccine: Vaccine development implications. *Journal of Infectious Diseases*, 207(3):511–519.

[Paradis et al., 2004] Paradis, E., Claude, J., and Strimmer, K. (2004). APE: Analyses of phylogenetics and evolution in R language. *Bioinformatics*, 20(2):289–290.

[Paten et al., 2011] Paten, B., Diekhans, M., Earl, D., John, J. S., Ma, J., Suh, B., and Haussler, D. (2011). Cactus Graphs for Genome Comparisons. *Journal of Computational Biology*, 18(3):469–481.

[Paten et al., 2017a] Paten, B., Novak, A. M., Eizenga, J. M., and Garrison, E. (2017a). Genome graphs and the evolution of genome inference.

[Paten et al., 2017b] Paten, B., Novak, A. M., Garrison, E., and Hickey, G. (2017b). Superbubbles, ultrabubbles and cacti. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 10229 LNCS, pages 173–189. Cold Spring Harbor Laboratory.

[Patro et al., 2015] Patro, R., Duggal, G., Love, M. I., Irizarry, R. A., and Kingsford, C. (2015). Salmon provides accurate, fast, and bias-aware transcript expression estimates using dual-phase inference. *http://www.biorxiv.org/content/early/2015/06/27/021592*, page 021592.

[Patro et al., 2017] Patro, R., Duggal, G., Love, M. I., Irizarry, R. A., and Kingsford, C. (2017). Salmon provides fast and bias-aware quantification of transcript expression. *Nature Methods*, 14(4):417–419.

[Patro et al., 2014] Patro, R., Mount, S. M., and Kingsford, C. (2014). Sailfish enables alignment-free isoform quantification from RNA-seq reads using lightweight algorithms. *Nature Biotechnology*, 32(5):462–464.

[Pearce et al., 2004] Pearce, J. A., Triglia, T., Hodder, A. N., Jackson, D. C., Cowman, A. F., and Anders, R. F. (2004). Plasmodium falciparum Merozoite Surface Protein 6 Is A Dimorphic Antigen. *Infection and Immunity*, 72(4):2321–2328.

[Pell et al., 2012] Pell, J., Hintze, A., Canino-Koning, R., Howe, A., Tiedje, J. M., and Brown, C. T. (2012). Scaling metagenome sequence assembly with probabilistic de Bruijn graphs. *Proceedings of the National Academy of Sciences of the United States of America*, 109(33):13272–7.

[Pertea et al., 2016] Pertea, M., Kim, D., Pertea, G. M., Leek, J. T., and Salzberg, S. L. (2016). Transcript-level expression analysis of RNA-seq experiments with HISAT, StringTie and Ballgown. *Nature Protocols*, 11(9):1650–1667.

[Pevzner et al., 2001] Pevzner, P. A., Tang, H., and Waterman, M. S. (2001). An Eulerian path approach to DNA fragment assembly. *Proceedings of the National Academy of Sciences of the United States of America*, 98(17):9748–53.

[Pickrell et al., 2009] Pickrell, J. K., Coop, G., Novembre, J., Kudaravalli, S., Li, J. Z., Absher, D., Srinivasan, B. S., Barsh, G. S., Myers, R. M., Feldman, M. W., and Pritchard, J. K. (2009). Signals of recent positive selection in a worldwide sample of human populations. *Genome research*, 19(5):826–37.

[Poirel et al., 2010] Poirel, L., Naas, T., and Nordmann, P. (2010). Diversity, epidemiology, and genetics of class D β-lactamases.

[Raman et al., 2002] Raman, R., Raman, V., and Rao, S. S. (2002). Succinct indexable dictionaries with applications to encoding k-ary trees and multisets. pages 233–242.

[Reinert et al., 2015] Reinert, K., Langmead, B., Weese, D., and Evers, D. J. (2015). Alignment of Next-Generation Sequencing Reads. *Annual Review of Genomics and Human Genetics*, 16(1):133–151.

[Reuter et al., 2015] Reuter, J. A., Spacek, D. V., and Snyder, M. P. (2015). High-Throughput Sequencing Technologies.

[Rice et al., 2014] Rice, B. L., Acosta, M. M., Pacheco, M. A., Carlton, J. M., Barnwell, J. W., and Escalante, A. A. (2014). The origin and diversification of the merozoite surface protein 3 (msp3) multi-gene family in Plasmodium vivax and related parasites. *Molecular Phylogenetics and Evolution*, 78(1):172–184.

[Rimmer et al., 2014] Rimmer, A., Phan, H., Mathieson, I., Iqbal, Z., Twigg, S. R. F., Wilkie, A. O. M., McVean, G., Lunter, G., and Lunter, G. (2014). Integrating mapping-, assembly- and haplotype-based approaches for calling variants in clinical sequencing applications. *Nature Genetics*, 46(8):912–918.

[Rosen et al., 2017] Rosen, Y., Eizenga, J., and Paten, B. (2017). Describing the local structure of sequence graphs. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 10252 LNCS, pages 24–46. Cold Spring Harbor Laboratory.

[Roy, 2015] Roy, S. W. (2015). The Plasmodium gaboni genome illuminates allelic dimorphism of immunologically important surface antigens in P. falciparum. *Infection, Genetics and Evolution*, 36:441–449.

[Roy et al., 2008] Roy, S. W., Ferreira, M. U., and Hartl, D. L. (2008). Evolution of allelic dimorphism in malarial surface antigens. *Heredity*, 100(2):103–110.

[RTS, 2015] RTS, S. C. T. P. (2015). Efficacy and safety of RTS,S/AS01 malaria vaccine with or without a booster dose in infants and children in Africa: final results of a phase 3, individually randomised, controlled trial. *The Lancet*, 386(9988):31–45.

[Scherf et al., 2008] Scherf, A., Lopez-Rubio, J. J., and Riviere, L. (2008). Antigenic Variation in Plasmodium falciparum. *Annu. Rev. Microbiol*, 62:445–70.

[Schnattinger et al., 2012] Schnattinger, T., Ohlebusch, E., and Gog, S. (2012). Bidirectional search in a string with wavelet trees and bidirectional matching statistics. *Information and Computation*, 213:13–22.

[Schneeberger et al., 2009] Schneeberger, K., Hagmann, J., Ossowski, S., Warthmann, N., Gesing, S., Kohlbacher, O., and Weigel, D. (2009). Simultaneous alignment of short reads against multiple genomes. *Genome Biology*, 10(9):R98.

[Schwartz et al., 2012] Schwartz, L., Brown, G. V., Genton, B., and Moorthy, V. S. (2012). A review of malaria vaccine clinical projects based on the WHO rainbow table. *Malaria Journal*, 11(1):11.

[Sepúlveda et al., 2013] Sepúlveda, N., Campino, S. G., Assefa, S. A., Sutherland, C. J., Pain, A., and Clark, T. G. (2013). A Poisson hierarchical modelling approach to detecting copy number variation in sequence coverage data. *BMC Genomics*, 14.

[Sharakhov et al., 2006] Sharakhov, I. V., White, B. J., Sharakhova, M. V., Kayondo, J., Lobo, N. F., Santolamazza, F., Della Torre, A., Simard, F., Collins, F. H., and Besansky, N. J. (2006). Breakpoint structure reveals the unique origin of an interspecific chromosomal inversion (2La) in the Anopheles gambiae complex. *Proceedings of the National Academy of Sciences of the United States of America*, 103(16):6258–6262.

[Sievers et al., 2011] Sievers, F., Wilm, A., Dineen, D., Gibson, T. J., Karplus, K., Li, W., Lopez, R., McWilliam, H., Remmert, M., Söding, J., Thompson, J. D., and Higgins, D. G. (2011). Fast, scalable generation of high-quality protein multiple sequence alignments using Clustal Omega. *Molecular systems biology*, 7(1):539.

[Simpson and Durbin, 2010] Simpson, J. T. and Durbin, R. (2010). Efficient construction of an assembly string graph using the FM-index. *Bioinformatics (Oxford, England)*, 26(12):i367–73.

[Simpson and Durbin, 2012] Simpson, J. T. and Durbin, R. (2012). Efficient de novo assembly of large genomes using compressed data structures. *Genome research*, 22(3):549–56.

[Simpson et al., 2009] Simpson, J. T., Wong, K., Jackman, S. D., Schein, J. E., Jones, S. J. M., and Birol, I. (2009). ABySS: a parallel assembler for short read sequence data. *Genome research*, 19(6):1117–23.

[Singh et al., 2009] Singh, S., Soe, S., Weisman, S., Barnwell, J. W., Pérignon, J. L., and Druilhe, P. (2009). A conserved multi-gene family induces cross-reactive antibodies effective in defense against Plasmodium falciparum. *PloS one*, 4(4):e5410.

[Sirén, 2017] Sirén, J. (2017). Indexing Variation Graphs. In *2017 Proceedings of the Ninteenth Workshop on Algorithm Engineering and Experiments (ALENEX)*, pages 13–27, Philadelphia, PA. Society for Industrial and Applied Mathematics.

[Sirén et al., 2014] Sirén, J., Välimäki, N., and Mäkinen, V. (2014). Indexing graphs for path queries with applications in genome research. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 11(2):375–388.

[Smith and Waterman, 1981] Smith, T. F. and Waterman, M. S. (1981). Identification of common molecular subsequences. *Journal of Molecular Biology*, 147(1):195–197.

[Soulama et al., 2011] Soulama, I., Bigoga, J. D., Ndiaye, M., Bougouma, E. C., Quagraine, J., Casimiro, P. N., Stedman, T. T., and Sirima, S. B. (2011). Genetic diversity of polymorphic vaccine candidate antigens (apical membrane antigen-1, merozoite surface protein-3, and erythrocyte binding antigen-175) in Plasmodium falciparum isolates from Western and Central Africa. *American Journal of Tropical Medicine and Hygiene*, 84(2):276–284.

[Srivastava et al., 2016] Srivastava, A., Sarkar, H., Gupta, N., and Patro, R. (2016). RapMap: a rapid, sensitive and accurate tool for mapping RNA-seq reads to transcriptomes. *Bioinformatics*, 32(12):i192–i200.

[Stanisic et al., 2013] Stanisic, D. I., Barry, A. E., and Good, M. F. (2013). Escaping the immune system: How the malaria parasite makes vaccine development a challenge.

[Stefansson et al., 2005] Stefansson, H., Helgason, A., Thorleifsson, G., Steinthorsdottir, V., Masson, G., Barnard, J., Baker, A., Jonasdottir, A., Ingason, A., Gudnadottir, V. G., Desnica, N., Hicks, A., Gylfason, A., Gudbjartsson, D. F., Jonsdottir, G. M., Sainz, J., Agnarsson, K., Birgisdottir, B., Ghosh, S., Olafsdottir, A., Cazier, J.-B., Kristjansson, K., Frigge, M. L., Thorgeirsson, T. E., Gulcher, J. R., Kong, A., and Stefansson, K. (2005). A common inversion under selection in Europeans. *Nature Genetics*, 37(2):129–137.

[Sudmant et al., 2015] Sudmant, P. H., Rausch, T., Gardner, E. J., Handsaker, R. E., Abyzov, A., Huddleston, J., Zhang, Y., Ye, K., Jun, G., Hsi-Yang Fritz, M., Konkel, M. K., Malhotra, A., Stütz, A. M., Shi, X., Paolo Casale, F., Chen, J., Hormozdiari, F., Dayama, G., Chen, K., Malig, M., Chaisson, M. J. P., Walter, K., Meiers, S., Kashin, S., Garrison, E., Auton, A., Lam, H. Y. K., Jasmine Mu, X., Alkan, C., Antaki, D., Bae, T., Cerveira, E., Chines, P., Chong, Z., Clarke, L., Dal, E., Ding, L., Emery, S., Fan, X., Gujral, M., Kahveci, F., Kidd, J. M., Kong, Y., Lameijer, E.-

W., McCarthy, S., Flicek, P., Gibbs, R. A., Marth, G., Mason, C. E., Menelaou, A., Muzny, D. M., Nelson, B. J., Noor, A., Parrish, N. F., Pendleton, M., Quitadamo, A., Raeder, B., Schadt, E. E., Romanovitch, M., Schlattl, A., Sebra, R., Shabalin, A. A., Untergasser, A., Walker, J. A., Wang, M., Yu, F., Zhang, C., Zhang, J., Zheng-Bradley, X., Zhou, W., Zichner, T., Sebat, J., Batzer, M. A., McCarroll, S. A., Mills, R. E., Gerstein, M. B., Bashir, A., Stegle, O., Devine, S. E., Lee, C., Eichler, E. E., Korbel, J. O., and Korbel, J. O. (2015). An integrated map of structural variation in 2,504 human genomes. *Nature*, 526(7571):75–81.

[Sundararaman et al., 2016] Sundararaman, S. A., Plenderleith, L. J., Liu, W., Loy, D. E., Learn, G. H., Li, Y., Shaw, K. S., Ayouba, A., Peeters, M., Speede, S., Shaw, G. M., Bushman, F. D., Brisson, D., Rayner, J. C., Sharp, P. M., and Hahn, B. H. (2016). Genomes of cryptic chimpanzee Plasmodium species reveal key evolutionary events leading to human malaria. *Nature Communications*, 7:11078.

[Takala et al., 2009] Takala, S. L., Coulibaly, D., Thera, M. A., Batchelor, A. H., Cummings, M. P., Escalante, A. A., Ouattara, A., Traore, K., Niangaly, A., Djimde, A. A., Doumbo, O. K., and Plowe, C. V. (2009). Extreme Polymorphism in a Vaccine Antigen and Risk of Clinical Malaria: Implications for Vaccine Development. *Science Translational Medicine*, 1(2):2ra5–2ra5.

[Takala and Plowe, 2009] Takala, S. L. and Plowe, C. V. (2009). Genetic diversity and malaria vaccine design, testing and efficacy: Preventing and overcoming 'vaccine resistant malaria'.

[Tan et al., 2010] Tan, J. C., Tan, A., Checkley, L., Honsa, C. M., and Ferdig, M. T. (2010). Variable numbers of tandem repeats in Plasmodium falciparum genes. *Journal of molecular evolution*, 71(4):268–78.

[Tetteh et al., 2013] Tetteh, K. K. A., Osier, F. H. A., Salanti, A., Kamuyu, G., Drought, L., Failly, M., Martin, C., Marsh, K., and Conway, D. J. (2013). Analysis of antibodies to newly described plasmodium falciparum merozoite antigens supports MSPDBL2 as a predicted target of naturally acquired immunity. *Infection and Immunity*, 81(10):3835–3842.

[Tetteh et al., 2009] Tetteh, K. K. A., Stewart, L. B., Ochola, L. I., Amambua-Ngwa, A., Thomas, A. W., Marsh, K., Weedall, G. D., and Conway, D. J. (2009). Prospective identification of malaria parasite genes under balancing selection. *PloS one*, 4(5):e5568.

[Tettelin et al., 2005] Tettelin, H., Masignani, V., Cieslewicz, M. J., Donati, C., Medini, D., Ward, N. L., Angiuoli, S. V., Crabtree, J., Jones, A. L., Durkin, A. S., DeBoy, R. T., Davidsen, T. M., Mora, M., Scarselli, M., Margarit y Ros, I., Peterson, J. D., Hauser, C. R., Sundaram, J. P., Nelson, W. C., Madupu, R., Brinkac, L. M., Dodson, R. J., Rosovitz, M. J., Sullivan, S. A., Daugherty, S. C., Haft, D. H., Selengut, J., Gwinn, M. L., Zhou, L., Zafar, N., Khouri, H., Radune, D., Dimitrov, G., Watkins, K., O'Connor, K. J. B., Smith, S., Utterback, T. R., White, O., Rubens, C. E., Grandi, G., Madoff, L. C., Kasper, D. L., Telford, J. L., Wessels, M. R., Rappuoli, R., and Fraser, C. M. (2005). Genome analysis of multiple pathogenic isolates of Streptococcus agalactiae: Implications for the microbial "pan-genome". *Proceedings of the National Academy of Sciences*, 102(39):13950–13955.

[Tettelin et al., 2008] Tettelin, H., Riley, D., Cattuto, C., and Medini, D. (2008). Comparative genomics: the bacterial pan-genome. *Current opinion in microbiology*, 11(5):472–7.

[The 1000 Genomes Project Consortium, 2015] The 1000 Genomes Project Consortium (2015). A global reference for human genetic variation. *Nature*, 526(7571):68–74.

[The Computational Pan-Genomics Consortium, 2016] The Computational Pan-Genomics Consortium (2016). Computational pan-genomics: status, promises and challenges. *Briefings in Bioinformatics*, page bbw089.

[Thera et al., 2011] Thera, M. A., Doumbo, O. K., Coulibaly, D., Laurens, M. B., Ouattara, A., Kone, A. K., Guindo, A. B., Traore, K., Traore, I., Kouriba, B., Diallo, D. A., Diarra, I., Daou, M., Dolo, A., Tolo, Y., Sissoko, M. S., Niangaly, A., Sissoko, M., Takala-Harrison, S., Lyke, K. E., Wu, Y., Blackwelder, W. C., Godeaux, O., Vekemans, J., Dubois, M.-C., Ballou, W. R., Cohen, J., Thompson, D., Dube, T., Soisson, L., Diggs, C. L., House, B., Lanar, D. E., Dutta, S., Heppner, D. G., and Plowe, C. V. (2011). A Field Trial to Assess a Blood-Stage Malaria Vaccine. *New England Journal of Medicine*, 365(11):1004–1013.

[Thera and Plowe, 2012] Thera, M. A. and Plowe, C. V. (2012). Vaccines for Malaria: How Close Are We? *Annual Review of Medicine*, 63(1):345–357.

[Trask et al., 1998] Trask, B. J., Massa, H., Brand-Arpon, V., Chan, K., Friedman, C., Nguyen, O. T., Eichler, E., van den Engh, G., Rouquier, S., Shizuya, H., and Giorgi, D. (1998). Large multi-chromosomal duplications encompass many members of the olfactory receptor gene family in the human genome. *Human molecular genetics*, 7(13):2007–2020.

[Valenzuela et al., 2015] Valenzuela, D., Välimäki, N., Pitkänen, E., and Mäkinen, V. (2015). On enhancing variation detection through pan-genome indexing. *bioRxiv*, page 021444.

[Ware et al., 1993] Ware, L. A., Kain, K. C., Lee Sim, B., Haynes, J., Baird, J., and Lanar, D. E. (1993). Two alleles of the 175-kilodalton Plasmodium falciparum erythrocyte binding antigen. *Molecular and Biochemical Parasitology*, 60(1):105–109.

[Weigel and Mott, 2009] Weigel, D. and Mott, R. (2009). The 1001 Genomes Project for Arabidopsis thaliana. *Genome Biology*, 10(5):107.

[WHO, 2016] WHO (2016). *World Malaria Report*.

[Wickramarachchi et al., 2009] Wickramarachchi, T., Cabrera, A. L., Sinha, D., Dhawan, S., Chandran, T., Devi, Y. S., Kono, M., Spielmann, T., Gilberger, T. W., Chauhan, V. S., and Mohmmed, A. (2009). A novel Plasmodium falciparum erythrocyte binding protein associated with the merozoite surface, PfDBLMSP. *International Journal for Parasitology*, 39(7):763–773.

[Yasukochi et al., 2017] Yasukochi, Y., Naka, I., Patarapotikul, J., Hananantachai, H., and Ohashi, J. (2017). Evolution of Fseg/Cseg dimorphism in region III of the Plasmodium falciparum eba-175 gene. *Infection, Genetics and Evolution*, 49:251–255.

[Zerbino and Birney, 2008] Zerbino, D. R. and Birney, E. (2008). Velvet: algorithms for de novo short read assembly using de Bruijn graphs. *Genome research*, 18(5):821–9.

[Zhu et al., 2017] Zhu, S. J., Almagro-Garcia, J., and McVean, G. (2017). Deconvoluting multiple infections in Plasmodium falciparum from high throughput sequencing data. *bioRxiv*.

[Zilversmit et al., 2010] Zilversmit, M. M., Volkman, S. K., DePristo, M. A., Wirth, D. F., Awadalla, P., and Hartl, D. L. (2010). Low-complexity regions in Plasmodium falciparum: missing links in the evolution of an extreme genome. *Molecular biology and evolution*, 27(9):2198–209.

[Zook et al., 2014] Zook, J. M., Chapman, B., Wang, J., Mittelman, D., Hofmann, O., Hide, W., and Salit, M. (2014). Integrating human sequence data sets provides

a resource of benchmark SNP and indel genotype calls. *Nature Biotechnology*, 32(3):246–251.