UNIVERSITY OF CALIFORNIA, SAN DIEGO

Algorithms and Methods for Characterizing Genetic Variability in Humans

A dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Computer Science

by

Christine Lo

Committee in charge:

Professor Vineet Bafna, Chair
Professor Ronald Graham
Professor Pavel Pevzner
Professor Bing Ren
Professor Kun Zhang

2014

The dissertation of Christine Lo is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

Chair

University of California, San Diego

2014

DEDICATION

To my parents.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

This research would not have been possible without the support of many people. First and foremost, I'd like to thank my advisor, Vineet Bafna, for his guidance and patience throughout my graduate studies. Vineet has given me the freedom to pursue my research interests while always being available to provide advice and encouragement. I'd also like to thank Pavel Pevzner, for his insightful comments and feedback. I'd also like to thank the rest of my committee members- Kun Zhang, Ron Graham, and Bing Ren- for their advice and time.

I have been blessed to work with some of the best and brightest in the field, and like to thank all my collaborators, co-authors, current and former lab mates, in particular, Shay, Christina, Sangwoo, Ali, Vikas, Bjarni, Nitin, Natalie, Jocelyne, Anand, Roy, Doruk, Sunghee, and Seong.

Last but definitely not least, I'd like to thank my family and friends for exploring the world with me, for laughter, for support, and for love.

Chapter 2, in part, is a reprint of the material as it appears in BMC Bioinformatics 2011. Lo C, Bashir A, Bansal V, and Bafna V. "Strobe sequence design for haplotype assembly." The dissertation author was the primary investigator and author of this paper.

Chapter 3, in part, is a reprint of the material as it appears in Genome Biology 2013. Lo, Christine; Liu, Rui; Lee, Jehyuk; Robasky, Kimberly; Byrne, Susan; Lucchesi, Carolina; Aach, John; Church, George; Bafna, Vineet; Zhang, Kun."On the design of clone-based haplotyping." The dissertation author was the primary investigator and author of this paper.

Chapter 4, in full, is a reprint of the material as it appears in BMC Bioinformatics 2013. Lo, Christine; Kim, Sangwoo, Zakov, Shay; Bafna, Vineet. "Evaluating genome architecture of a complex region via generalized bipartite matching." The dissertation author was the primary investigator and author of this paper.

Chapter 5, in part, is currently being prepared for submission for publication of the material. Lo, Christine; Bafna, Vineet. The dissertation author was the primary investigator and author of this paper.

VITA

2009	Bachelor of Science in Electrical Engineering and Computer Sciences, University of California, Berkeley
2012	Master of Science in Computer Science, University of California, San Diego
2014	Doctor of Philosophy in Computer Science, University of California, San Diego

PUBLICATIONS

“Barcode-based identification in KIR region of human genome”. C. Lo, S. Zakov, S. Kim, and V. Bafna. In preparation, 2014.

“SeeSite: Characterizing Relationships between Splice Junctions and Splicing Enhancers”. C. Lo, B. Kakaradov, D. Lokshtanov, and C. Boucher. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 2014.

“Evaluating genome architecture of a complex region via generalized bipartite matching”. C. Lo, S. Kim, S. Zakov, and V. Bafna. BMC Bioinformatics, 2013.

“On the design of clone-based haplotyping”. C. Lo, R. Liu, J. Lee, K. Robasky, S. Byrne, C. Lucchesi, J. Aach, G. Church, V. Bafna, and K. Zhang. Genome Bioloy, 2013.

“Outlier detection for DNA fragment assembly”. C. Boucher, C. Lo, and D. Lokshtanov. arXiv Pre-print ArXiv, 2011.

“Global DNA hypomethylation coupled to repressive chromatin domain formation and gene silencing in breast cancer”. G. C. Hon, R. D. Hawkins, O. L. Caballero, C. Lo, R. Lister, M. Pelizzola, A. Valsesia, Z. Ye, S. Kuan, L. E. Edsall, A. A. Camargo, B. J. Stevenson, J. R. Ecker, V. Bafna, R. L. Strausberg, A. J. Simpson, and B. Ren. Genome Research, 2011.

“Strobe sequence design for haplotype assembly”. C. Lo, A. Bashir, V. Bansal, and V. Bafna. BMC Bioinformatics, 2011.

ABSTRACT OF THE DISSERTATION

Algorithms and Methods for Characterizing Genetic Variability in Humans

by

Christine Lo

Doctor of Philosophy in Computer Science

University of California, San Diego, 2014

Professor Vineet Bafna, Chair

Characterizing genetic variation including point mutations and structural variations, is key to understanding phenotypic variation in humans. The rapid development of sequencing technology has fueled the development of computational methods for elucidating genetic variation. In this dissertation, we develop novel computational methods to mainly target two human genetic variation problems using current and emerging sequencing technology.

Capturing variation on the haplotype level is challenging with current sequencing technology as it involves linking together short sequenced fragments of

the genome that overlap at least two heterozygous sites. While there has been a lot of research on correcting errors to achieve accurate haplotypes, relatively little work has been done on designing sequencing experiments to get long haplotypes. With the development of new sequencing technology and experimental haplotyping methods, we parametrize the haplotyping problem in two contexts, strobe sequencing and clone-based haplotyping, and provide theoretical and empirical assessment of the impact of different parameters on haplotype length.

Variation in certain regions of the genome are harder to capture than others. Reconstruction of the donor genome from whole genome sequence data is either based on de novo assembly of the short reads or on mapping reads to a standard reference genome. While these techniques work well for inferring 'simple' genomic regions, they are confounded by regions with complex variation patterns including regions of direct immunological relevance such as the HLA and KIR regions. Characterizing these regions have previously relied on laboratory methods using traditional and quantitative PCR primers and probes which can be labor and time intensive. We address the problem of ambiguous mapping in complex regions by defining a new scoring function for read-to-genome matchings. This scoring function is applied to predicted sequence assemblies of the KIR region in order to determine the most likely KIR haplotype groups of the donor. In another approach, we developing a novel method based on barcoding (deriving signatures) known KIR templates in order to determine the copy number and allelic type of genes in the KIR region directly from whole genome sequencing data without assembly or mapping.

Chapter 1

Introduction

Deoxyribonucleic acid (DNA) encodes information regarding the development, function, and reproduction of living organisms by being transcribed into ribonucleic acid (RNA) which is translated into proteins that determine cell function and ultimately phenotype. Thus, knowledge of the genomic sequence encoded by DNA is motivated by the promise of personalized medicine. DNA is a molecule that is made up of nucleotides with one of four bases: adenine (A), cytosine (C), guanine (G), and thymine (T) and is tightly packed in structures called chromosomes. The human genome is made up of 22 types of chromosomes and 2 sex chromosomes, and because humans are diploid there are two copies of each chromosome type. The two chromosomes of the same type are highly homologous to each other and only differ at a small fraction of variant sites (0.1%). From a computational stand point the genomic sequence can be represented as a string of $\{A,C,G,T\}$ that is 3 billion base pairs long or 6 billion base pairs if the sequence of the two homologous chromosomes are represented separately. The algorithms and methods in this dissertation revolve around uncovering the genetic variation in humans as these variations are key to understanding phenotypic variation in people including susceptibility to disease.

1.1 Background

1.1.1 Variation in Humans

Genomic variations are often classified into two groups based on their size. Small variations (less than 1Kb in size) include *point mutations*, or *single nucleotide variations* (SNVs), which only affect a single nucleotide. Further classification of this type of point mutations are defined somewhat arbitrarily. *Single nucleotide polymorphisms (SNPs)* are point mutations that are common in a population (occurring in at least 1% of the population) while *p*rivate point mutations are unique to an individual (occurring in 1% of the population). To date, there are over 43 million SNPs registered in dbSNP for the human genome (Sherry et al. (2001)). Other types of small variations include indels (small insertions and deletions) and microsatellites. Larger variations (at least 1Kb in size) are known as *structural variations* (SV). *Copy number variations* (CNVs) are a type of structural variation that is characterized by the deletion or duplication of large sections of the genome. Other types of structural variations include genomic rearrangements such as insertions, inversions and translocations.

1.1.2 Inheritance

Humans inherits one copy of each chromosome type from the mother and the father. Cellular processes like mutation and recombination result in the inherited chromosome being different or mutated away from its parent (Figure 1.1). A *mutation* will change the allelic value at a chromosome site from parent to child. In humans, the mutation rate is estimated to be $1.3 - 1.8 \times 10^{-8}$ per base per generation (Lynch (2010)). The low mutation rate is why two chromosomes of the same type only differ at a small fraction (0.1%) of variant sites. *Recombination*

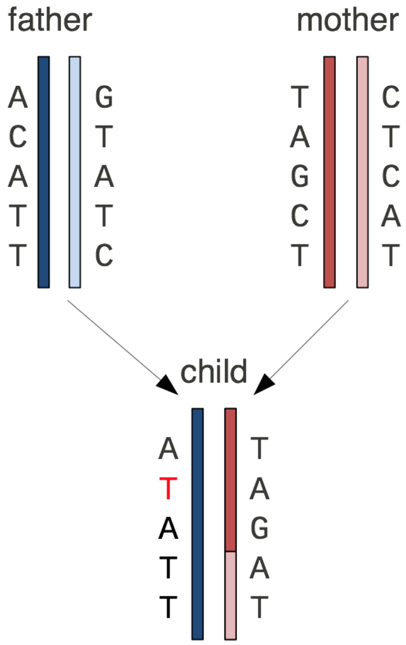


Figure 1.1: **Mutation and Recombination.** A child inherits one chromosome from the father and one from the mother. In this example, there is a mutation at the second site of the paternal chromosome ($\mathtt{C} \rightarrow \mathtt{T}$). There is also a recombination on the maternal chromosome between the third and fourth site.

occurs when the two chromosomes of the same type exchange regions of their genome during meiosis, which is the process where reproductive cells are formed in the parent. Certain areas of the genome are more prone to recombination than others and regions of sites with a high probability of recombining are known as *recombination hotspots*.

1.1.3 Capturing Variation

Sequencing platforms allows one to read or sequence the whole genome directly rather than probe at specific locations. In recent years, sequencing platforms have been rapidly developing and are becoming more cost effective than they were before. Sequencing platforms take as input a DNA sample, cut up and amplify (replicate) the DNA, read each DNA fragment, and output a set of reads- each read is the nucleotide sequence corresponding to a small portion of a chromosome from the sample.

Though there are a variety of sequencing platforms and protocols, the sequence fragments output from sequencing technologies represent only a small subsequence of the genome and require computational tools to reconstruct or infer the donor sequence. The two main computational methods for inferring the donor sequence are *de-novo assembly* of the (short) fragments and *mapping* of the fragments to a standard reference sequence. Both of these approaches are complicated by sequencing errors, alignment errors, and repetitive regions of the genome. Often the reconstructed donor sequence is left incomplete with gaps in areas where no fragments were sequenced and in areas where fragment reads were not long enough to span repeat regions. Once assembled or mapped, genotype information is relatively easy to obtain via statistical methods with several computational tools devoted to this problem (McKenna et al. (2010b); Li et al. (2009b)); however, haplotype information is still limited.

1.2 Haplotype Phasing

Haplotypes are the combination of alleles of SNPs along a single chromosome while *genotypes* are the combined haplotype information for a pair of chromosomes. Recall that humans are diploid, inheriting a pair of each chromosome, where the two copies of each chromosome are highly homologous to each other and differ in roughly 0.1% of variant sites. Characterizing variation on the haplotype level is challenging with current sequencing technology, as the similarity between the two homologous chromosomes make it difficult to elucidate the combination of alleles on a single chromosome.

Given the importance of haplotype information, a variety of computational, statistical, and experimental techniques have been developed to phase chromosomes. While we focus our discussion and work on haplotyping with sequencing data, we mention a few other strategies to put our work in context. *Population-based inference* exploits linkage disequilibrium to identify likely haplotypes from genotype data. Consider a population of individuals sampled at two sites and an individual with the genotype (A/C-T/A). If a large number of individuals carry the homozygous genotypes (A/A-T/T), we can infer that the haplotype (A-T) is common in the population, and infer the two haplotypes to be (A-T) and (C-A). Large scale studies such as the International HapMap Project (Frazer et al. (2007)) and 1000 Genomes project (Abecasis et al. (2010a)) have attempted to phase individuals using this approach; however, historical recombination events can reduce or eliminate linkage, and reliable phasing can only be achieved over short regions, 30-50Kb on the average (Reich et al. (2001); Altshuler et al. (2005)). Furthermore, phasing private mutations may not be possible. While phasing is difficult with populations, it is almost trivial if parental information is known (Marchini et al. (2006); Roach et al. (2010)). Given that only a few crossovers occur on each chromosome during meiosis, a small sampling of homozygous alleles in the parents are sufficient to phase entire chromosomes. While *family-based haplotyping* is powerful, it is not always feasible as it requires genotype or sequencing information of the parents. Several methods have recently been developed for deriving chromosome length haplotypes based on isolation of a single cell followed by physical separation of chromosome pairs during mitosis using microfluidic devices, microdissection, and fluorescence-activated cell sorting (Fan et al. (2011); Ma et al. (2010); Yang et al. (2011)). Once separated, the chromosomes are amplified and sequenced. Another recent study achieved chromosome length haplotypes by isolation of sperm cells and deconvolution of recombination breakpoints (Kirkness et al. (2013)). While these techniques achieve chromosome length haplotypes, they are often sparse due to amplification bias. Furthermore, they require custom equipment and are quite labor intensive which could add to the cost and the throughput.

Alternatively, inferring haplotypes from sequencing data is attractive due to the proliferation of inexpensive sequencing techniques that have the throughput to sequence the entire human genome (Shendure and Ji (2008); Mardis (2008)). Because a sequence fragment is sampled from one chromosome, the heterozygous variants of the fragment can be chained together to assemble haplotypes, a problem known as *haplotype assembly*. Haplotype assembly was proposed some time ago (Halldórsson et al. (2003); Bafna et al. (2005)), but the data for individuals genomes has recently become more prevalent. The first sequence of a genomic individual, J. Craig Venter, (*HuRef*) was produced in 2007 using Sanger sequencing. The sequence fragments were paired-end, spanning $2 - 150$kbp linking subreads of 2kbp. Each base sampled an average of $6\times$. The phasing was quite effective, with a 'median' haplotype (the metric is precisely defined later) of length 350kbp (Levy et al. (2007)). Sanger sequencing provides long and accurate reads but lower throughput and expensive library preparation making it less cost-effective. By contrast, next-generation technologies allow for massively parallel sequencing, but have much shorter reads, and are more error prone. Recall that haplotype assembly relies on linking together heterozygous variants from sequence fragments; therefore, the length of the sequence fragment is an important factor in achieving long haplotypes. While current computational research focuses on improving haplotyping accuracy assuming the specific technological parameters are determined by the technology, the onset of newer technologies such as strobe sequencing and clone-based haplotyping (utilizes sequencing data) allow for user specification for an experiment.

In the first half of this thesis, we parametrize two types of sequencing technologies in order to develop more cost effect methods for achieving long range haplotypes. In chapter 2, we parameterize the technology of strobe sequencing and analyze the relationship between sequencing parameters and haplotype length. *Strobe sequencing* uses single molecule sequencing technology to allow for longer insert sizes (up to 9Kb) as well as variable advance length and multiple subreads. This work is also application to other sequencing technologies such as single-read and paired-end sequencing since strobe sequencing is a generalization of the two. In chapter 3, we parameterize *clone-based haplotyping* experiments , which extract long genomic fragments between $10 - 140$Kb depending on the type of clone used (Lo et al. (2013a)) and pool clones together before sequencing such that the probability that two overlapping clones belong to different haplotypes is low. Relative to haplotyping with pure sequencing technologies, a bit of cost is absorbed both in the preparation and computational phase in order to achieve longer sequence fragments. Parameterization is also applicable to long range fragment reads (i.e. complete genomics and moleculo) which similarly extract long sequence fragments before pooling. Applying our analysis, we demonstrate improved haplotype lengths using BAC clones on PGP1, the first volunteer of the Personal Genome Project (Church (2005)).

1.3 Genomic Regions With Complex Variation

With current technology, variation in certain regions of the genome are harder to capture than others. This is due to the fact that these regions are highly variable and repetitive. One such example is the Killer Immunoglobulin-like Receptor (KIR) region on chromosome 19 spanning $100 - 200$ thousand base pairs in length. To date, 15 genes including 2 pseudogenes have been identified in this region (Middleton and Gonzelez (2010)) with haplotypes containing a select combinations of 7 to 12 genes. Although variable, the KIR region is tightly organized and haplotypes can be classified into 7 known haplotype groups based on their gene content. Furthermore, haplotype groups are a combination of 4 centromeric and 2 telomeric motifs suggesting evolutionary history via recombination maintained by balancing selection (Gendzekhadze et al. (2006)). The region is also extremely diverse in allelic variation in genes, with over 670 known alleles for the 15 genes of this region (Robinson et al. (2005)). More recently, gene copy number variation (besides gene presence and absence) in this region has been discovered (Jiang et al. (2012); Pontikos et al. (2014)). Coupled with the fact that this region is marked by segmental duplications of various length, reconstruction of this region using high throughput sequencing technology and conventional computational methods (e.g. mapping and de-novo assembly) is challenging (Lo et al. (2013b)).

Characterizing the genetic variation in KIR region is of direct immunological relevance. The genes of the KIR region play a role in regulating the immune response in humans by inhibiting or activating natural killer (NK) cells. NK cells are important mediators of early immune response by recognizing ‘abnormal’ cells’ such as tumor cells and pathogen infected cells. Activating receptors will aid NK cells in identification of ‘abnormal cells’, while inhibitory receptors aid by recognizing ‘self’ molecules preventing an auto-immune response. Interestingly, the genes present in group A haplotypes have more of an inhibitory role where as the genes present in group B haplotypes have more of an activating role (Middleton and Gonzelez (2010)). Adding a level of diversity, evidence from disease association studies suggest the co-evolution between the KIR region and the HLA region, a region on chromosome 6 that is also marked by hyper-variability and its role in regulation of the immune system (Norman et al. (2013); Gendzekhadze et al. (2009)).

Previous attempts to characterize this regions have primarily relied on polymerase chain reaction (PCR) along with sequence specific primers (SSP) and sequence specific oglionucleotide probes (SSOP) to isolate specific genes in order to determine gene content and allelic type in this region (Middleton and Gonzelez (2010)). More recent studies have utilized quantitative PCR techniques to detect the rate of amplification and calculate the gene copy number in this region (Jiang et al. (2012); Pontikos et al. (2014)). In one study, pooled fosmid clones of fragment length $30 - 50$kbp were used to obtain the unambiguous haplotype sequence of this region for 12 individuals (24 haplotypes) (Pyo et al. (2010)). However, all these approaches are both labor and time intensive and require additional work even when whole genome sequence information is available.

The last part of this thesis is devoted to efficient computational methods for characterizing the KIR region from increasingly prevalent sequencing data. While the methods developed here are used to characterize the KIR region, they can be extended to uncover variability in other complex regions of the genome such as the HLA region. Chapter 4 is the first computational attempt for identifying the two KIR haplotype groups of an individual using sequencing data. We model the problem of ambiguous mapping in repetitive regions of the genome as a variant of the standard (one-to-one) weighted bipartite matching problem, which can be solved using network flows. The resulting method provides a scoring function for predicted sequence assemblies of the whole genome sequence reads derived from this region and use this scoring function to determine the most likely KIR haplotype group. In chapter 5 we develop a fast, alignment-free method to detect copy number and allelic type directly from whole genome sequencing data. As haplotype groups are classified based on gene content and thus copy number, our method has the ability to detect haplotype groups of the individual whether they be known or novel. We applied our method to type individuals from the 1000 genomes project (Abecasis et al. (2010a)) and the Icelandic population.

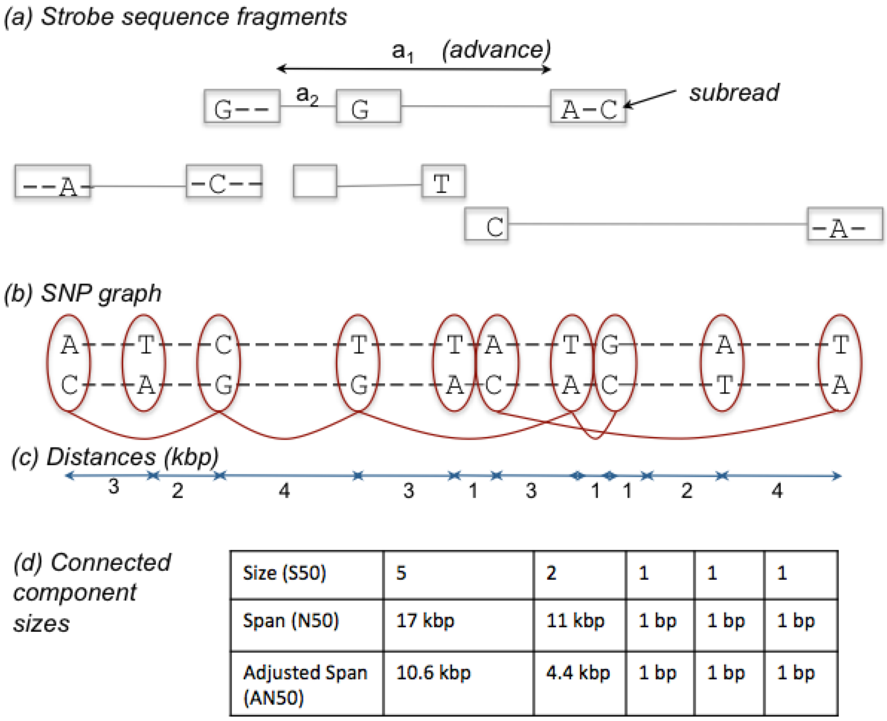
Chapter 2

Haplotyping with Strobe Sequencing Technology

2.1 Introduction

Much of the current computational research on haplotype assembly focuses on improving haplotype accuracy (Bansal et al. (2008); Bansal and Bafna (2008); He et al. (2010)). Until now, the length of the haplotypes depended upon the specific technological parameters, and was assumed to be determined by the technology. With recent developments in sequencing, the user has the ability to select different parameters for an experiment. Our paper investigates the relationship of sequencing parameters on the haplotype length.

Of particular relevance is the recent technology of *strobe sequencing*, available from Pacific Biosciences (Ritz et al. (2010)). In this technology, a genomic fragment is sequenced in a *strobed fashion* with subreads of pre-determined lengths separated by user-determined intervals (*advances*). In Figure 2.1a, we see a number of fragments with $k = 2$ strobes, and one with 3 strobes. Paired-end sequencing is



Size (S50)	5	2	1	1	1
Span (N50)	17 kbp	11 kbp	1 bp	1 bp	1 bp
Adjusted Span (AN50)	10.6 kbp	4.4 kbp	1 bp	1 bp	1 bp

Figure 2.1: **Schematic for Haplotype Assembly** (a) Strobe sequencing allows for the generation of subreads, separated by user-defined advance lengths. The reads can be mapped to the reference to detect heterozygous sites. (b) Nodes in the SNP-Graph correspond to heterozygous sites. Edges correspond to pairs of sites that are linked by a fragment. Haplotype assembly is limited to connected components of the SNP-Graph. (c) The distance between sites is used for measuring haplotype lengths. (d) The S50, N50, and AN50 measure of haplotype assembly.

analogous to strobe sequencing with $k = 2$, however it differs in that the sequenced reads must be from terminal portions of an insert which leads to reduced flexibility in selecting advance lengths. A key result of our analysis is that the choice of advance lengths can change the haplotype length by an order of magnitude for the same amount of sequencing. In fact, the best results are obtained by a complex distribution f on advance lengths. Besides k and f we also study the impact of other parameters on haplotype length. These include (a) L, the number of bp sequenced per fragment; $L = \sum_i l_i$, where l_i is the length of the i-th subread; (b) N: number of fragments sequenced; (c) A, the maximum insert size allowed. Note that because we usually fix L, the advance lengths are related to A. For example, the maximum advance length for $k = 2$ strobes is $A - L$. In addition, we usually work with coverage $c = NL/G$, which gives the number of times each bp is sampled, on average. To obtain our results, we developed a simulator that generates reads according to specific technological parameters, and constructs connected components of the SNP-Graph. The software is available upon request from the authors.

While the focus of our analysis is on designing experiments for haplotype length, we also touch upon haplotype accuracy. We use a simulator provided by Pacific Biosciences to generate strobe sequence data based on an error model having high rates (roughly symmetric) of insertions and deletions relative to miscall errors (Eid et al. (2009)). We use our previously designed tools to phase in the presence of error (Bansal et al. (2008)). Our results indicate that long and accurate haplotyping is feasible even with technology having such high error rates.

2.2 Preliminaries

We begin by formalizing the problem. Aligned fragments define a *SNP-Graph* in the individual, as shown in Figure 2.1b. Each heterozygous location corresponds to a node. When a fragment overlaps two sites, we add an edge to the corresponding nodes. It is easy to see that two sites can be phased if and only if they are connected in the SNP-Graph. Therefore the *length* of the haplotypes depend upon the size of connected components, while the *accuracy* of haplotypes depends upon the error in sequencing, depth of coverage, and computational algorithms for error correction. The quality of a haplotype is measured by metrics for length and accuracy.

Metrics for haplotype length

Given the SNP-Graph, we use three different metrics (S50, N50, AN50) to measure the median length of assembled haplotypes: S50, N50, and AN50, related to the size (number of SNPs), span (distance spanned), and adjusted span of the contigs respectively. See Figure 2.1d. Recall that the haplotyping is limited to connected components in the SNP-Graph. The length of a haplotype can be described in terms of its *size* (# of heterozygous sites), or *span* (distance between distal heterozygous sites). As the connected components can interleave, we define the *adjusted-span* of a component as the span times the fraction of sites that lie in the contig. In Figure 2.1d, we observed connected components of size 5 and 2 with spans 12kbp, and 11kbp, respectively. The adjusted spans are given by $\frac{5 \cdot 12}{8} = 7.5$kbp, and $\frac{2 \cdot 11}{5} = 4.4$kbp.

We define $S50$ (and $N50$) to be the *size* (respectively, span) such that 50% of all sites are in contigs of size (span) $S50$ (N50), or greater. As SNPs display a 'clumping' property, S50 might inflate the haplotype size. On the other hand, $N50$

tends to inflate the haplotype size when there are contigs that span a long distance, but do not phase many SNPs. The AN50, or adjusted N50 metric considers both span, and size. It is defined as the adjusted span s.t. 50% of the SNPs are in contigs with an adjusted span AN50 or larger. We will primarily use the AN50 metric. However, our results and trends remain the same for any metric.

Metrics for haplotype accuracy

Erroneous base-calls corrupt the accuracy of assembled haplotypes. In simulations, where the reference is known, we can measure the accuracy of the reconstructed haplotype as the *haplotype edit rate (HER)*, equal to the fraction of incorrectly called alleles. A second reason for incorrect haplotyping is that weak links might cause a 'switch', a crossover from one true haplotype to the other. This could potentially cause HER to be large, even though a single crossover can correct the haplotypes. See Figure 2.7. Therefore, we define another metric *switch error rate (SER)* which is the number of crossovers (per heterozygous site) in the assembled haplotypes to match the correct haplotype.

2.3 Effect on haplotype length

2.3.1 Singleton strobes

Assuming that the cost is proportional to the number of nucleotides sequenced, we compare all designs after fixing coverage c. A back-of-the-envelope calculation suggests that with long read lengths ($L \simeq 1$kbp), we should be able to link all SNPs together, given that the average pair of SNPs is 1kbp apart. The intuition is wrong because (a) a Poisson process for SNPs implies an exponential distribution of inter-SNP distance in a population- hence a long tail; and, (b) a single individual is heterozygous at only a subset of the SNPs. Indeed, the distribution of inter-SNP distances in HuRef is more consistent with the power-law (than exponential) with a long tail of large inter-SNP distances (Figure 2.2a). Therefore, we only reach an AN50=48kbp even with $L = 5$kbp and $c = 20\times$ (Figure 2.2b). Similar results can be obtained with mate pair sequencing ($k = 2$) at much lower coverage. The linking together of SNPs through subread probes is indeed the most significant parameter determining haplotype length.

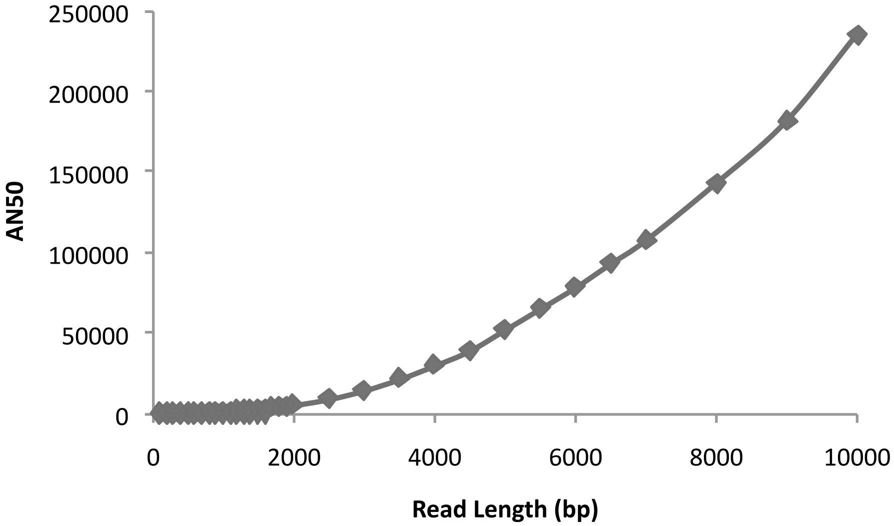


Figure 2.2: **Haplotype Assembly with Singleton Strobe** (a)Distribution of Inter-SNP Distances. The log-log plot suggests slower than exponential decay, better fit by a power-law. (b) With single strobes ($k = 1$), high coverage and very long reads are needed to achieve significant haplotypes.

2.3.2 Advance Lengths for Paired End Sequencing ($k = 2$)

We fixed the read-length $L = 900$bp as it is within the current mean length distribution reported by Pacific Biosciences (Davies (2010)). For $L = 900$bp,$c = 20\times$, and $k = 2$ subreads, choosing fixed insert sizes $A_1 = 3$kbp, $A_2 = 9$kbp results in low AN50 values 5.4kbp and 6.7kbp, respectively. However, a simple 50-50 mix of the two increases this by an order of magnitude AN50=54kbp. Clearly, variation in insert size, and thus advance length, is important. However, it is not immediately obvious what distribution of advance lengths will give the highest AN50. For example, we could consider uniformly varying advances from a minimum to a maximum length, or follow the library mix used for sequence assembly dominated by smaller advance lengths to form contigs, mixed with a smaller number of large advances to create scaffolds. To search efficiently over a large space of distributions, we used the 2-parameter β-distribution. For parameters (α, β), and maximum insert size A, define the p.d.f as

$$f(a) = \frac{a^{\alpha-1}(A-L-a)^{\beta-1}}{\int_{x=0}^{A-L} x^{\alpha-1}(A-L-x)^{\beta-1}dx} \tag{2.1}$$

where the denominator is a normalizing constant. Different choices of α, β provide a large range of distributions for $f(a)$ (beta). For example, larger α values correspond to a negative skew (longer advance lengths are preferred), while larger β correspond to a more positive skew. When $\alpha = \beta$, the distribution is symmetric. We systematically explored all α, β values in the interval $(0 - 4]$. Additionally, we implemented a simulated annealing algorithm (see Section 2.5) to identify the optimal choice of parameters.

Surprisingly, the distributions with the highest AN50 had $\alpha \in [1.0 - 3.2]$ and $\beta \in [0.3 - 0.9]$, and skewed heavily toward the longer clones. For $c = 20\times, L = 900$bp, $A = 9$kbp, $(\alpha, \beta) = (1.6, 0.5)$, we achieve an AN50$\simeq$ 151kbp (Figure 2.3). Even more surprising, distributions skewed toward smaller clone lengths $(\alpha, \beta) = (0.6, 2.3)$ had the worst performance (AN50=38kbp). Uniform $(\alpha, \beta) = (1, 1)$, and other symmetric distributions $(\alpha = \beta)$ show an intermediate performance. The bias is maintained at different values of coverage, maximum insert size, and other parameters. While there is a heavy bias towards longer clones, variation is important as well. For example, the distribution given by $(\alpha, \beta) = (4.5, 0.1)$ shows an extreme skew towards longer clone lengths so that it almost mimics a delta function at 9kbp and gives an AN50 of 45kbp. The trends do not change with a choice of other metrics S50, N50 (see Figure 2.8b-d).

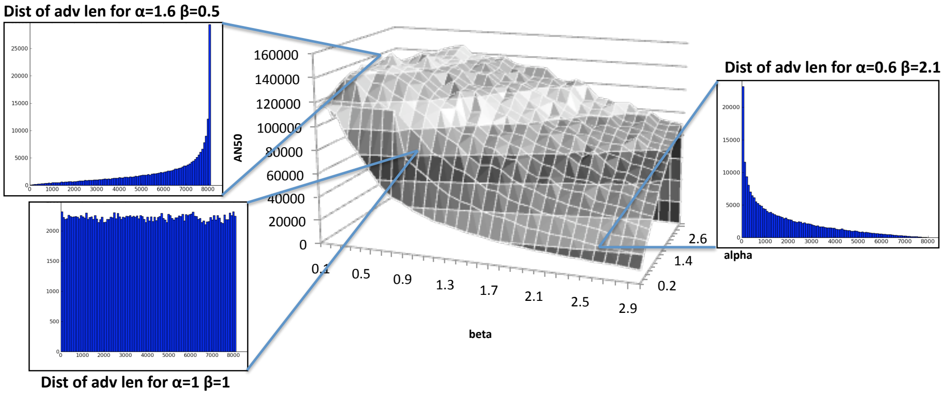


Figure 2.3: **AN50 for various α, β values.** This grid was created by simulating the first 10Mbp of chr1 (HuRef), at $L = 900$bp, $c = 20\times$, $A = 9$kbp, $\alpha \in (0, 3.4]$ and $\beta \in (0, 3]$. Each (α, β) pair was simulated 25 times, and the median AN50 was plotted.

Wasted Reads: Note that popular designs for sequence assembly emphasize short inserts (with a tight distribution of insert-lengths) mixed with a few large clones for scaffolding. By contrast, haplotype assembly is improved by focusing on larger inserts and higher variation. Figure 2.4a provides an illustration of the impact of different distributions of advance lengths on the connectivity of the SNP-Graph. A connected component with k vertices and m edges has $m - k + 1$ 'waste' edges, as only $k - 1$ 'useful' edges are needed to maintain connectivity. Due to the clustering of SNPs, a design with larger number of short advances has more wasted edges compared to a design with long advances. As each useful edge connects two previously unconnected components, it has a large impact on haplotype lengths. We computed the number of useful edges for the two designs, fixing $c = 20\times$ and varying maximum insert, A. We observe that the number of useful edges is always larger in designs with a bias towards long advance lengths (Figure 2.4b). For $A = 5$kbp, we see a 13% difference in useful edges between the two distributions.

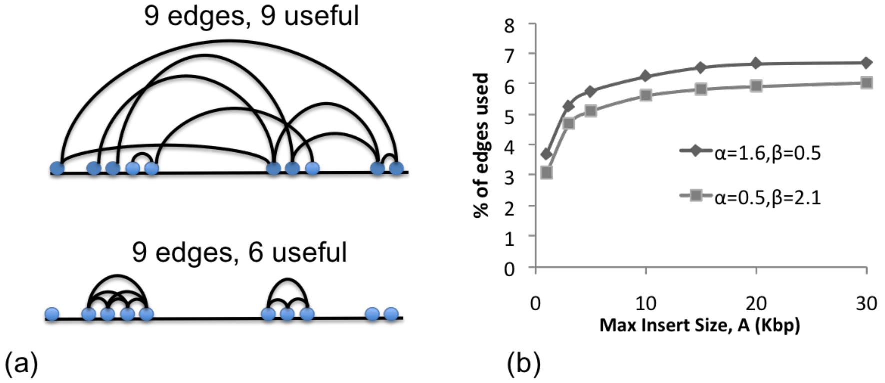


Figure 2.4: **Increase in the number of useful edges with increasing** A (a) A schematic illustration of the impact of designs on the number of useful edges. As there is a larger number of SNP pairs with small distances, designs that favor short edges tend to connect already connected components, and have less useful edges. (b) The fraction of useful edges for two designs, and for a range of values for A. Note that a large fraction of edges (reads) is wasted either because the edges do not connect two variants, or because the they connect already connected components. The design with longer advance lengths always has a larger fraction of useful edges resulting in better AN50. Simulations used $L = 900$bp, $c = 20\times$, $A = 9$kbp.

The Erdós-Renyi theory describes the evolution of a random graph from isolated components to a single component, with increasing number of edges (Erdos and Renyi (1959)). In our case, the edge probability in SNP-graphs is not initially uniform due to the clustering of SNPs (i.e. there is a bias towards proximal SNP pairs). By choosing designs with a bias towards longer advance lengths, we are essentially leveling out the probability of linking SNP pairs irrespective of their distance, leading to improved connectivity.

2.3.3 Other parameters

Maximum Insert Size, A**:** In Figure 2.5a, we plot maximum achieved

AN50 (for $c = 20\times, L = 900$bp) maximum theoretical AN50 (assuming infinite coverage) as a function of A. The achieved AN50 increases with increasing A for the same amount of sequencing ($c = 20\times$), indicating that the largest possible value of A should be chosen. Interestingly, the SA optimized parameters (α, β) remain similar as A is increased (See Table 2.2).

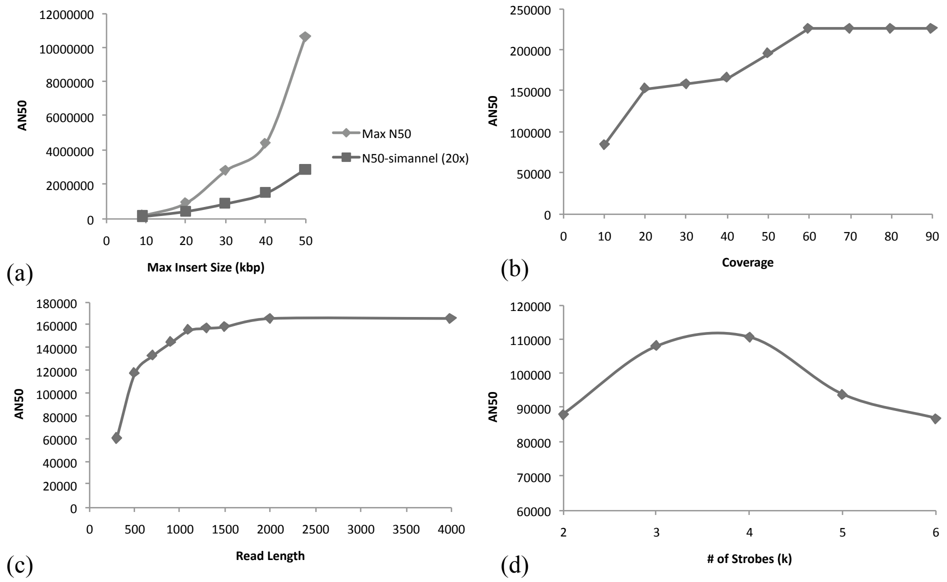


Figure 2.5: **AN50 for Other Parameters.** (a) Increasing A increases AN50. (b) Increasing c increases AN50 until saturation. (c) Increasing L increases AN50, but saturates quickly. (d) Using more strobes increases the variation of advance lengths, but decreases the size of the subreads. AN50 is maximized at $k = 3, 4$. All simulations were performed on the first 10Mbp of chr1 (HuRef) using $L = 900$bp, $c = 20\times$, $A = 9$kbp. SA based optimal (α, β) values were similar, and produced advance length distributions skewed towards longer advances. (See Table 2.2 for exact (α, β) values.)

Coverage, c: The effect of coverage on AN50 is analogous to increasing the edge probability, and we expect to see an increase in connectivity until saturation is reached. The plot in Figure 2.5b shows this for $A = 9000$bp, $L = 900$bp, and SA optimized (α, β).

Read length, L: Once A, c are fixed the impact of read-length L is minimal. Here, we assume that the subread is of minimal size (≥ 100) to permit accurate mapping. Initial improvement is seen with increasing L as the same subread captures proximal SNPs. However, the effect saturates quickly. (Figure 2.5c shows this for $A = 9\text{kbp}, k = 2, c = 20\times$ and SA optimized (α, β) values. Again, the β-distribution stays similar with changes in L. (See Table 2.2)

2.3.4 Number of strobes, k:

Besides flexibility in advance lengths, strobe sequencing allows the possibility of multiple strobes k. Figure 2.1a provides a cartoon of strobe sequencing for $k = 2$ and $k = 3$. To compare designs with different number of strobes, we fixed the subread lengths for each k to $l_k = L/k$, keeping the total read-length constant. We also fixed the maximum insert size, A. Recall from the paired-end results that longer subread lengths help cover the relatively high proportion of SNPs that are clustered close together. Therefore, increasing number of strobes helps increase the variation in advance lengths against the penalty of smaller subreads.

Optimal advance distribution for higher k

For a simulation with k strobes, we compute an optimal collection of (α_i, β_i) for $0 < i < k$ iteratively. Thus, for $k = 3$, a_1 is randomly generated with (α_1, β_1, A), and a_2 is randomly generated with (α_2, β_2, a_1). The strobed read is arranged as in Figure 2.1a with a_1 as the advance length between the subread$_1$ and subread$_3$, and a_2 as the advance length between subread$_1$ and subread$_2$. A similar pattern is used for higher k. While we see an improvement for $k = 3$ and $k = 4$, higher values of k do not help (Figure 2.5d).

The optimal distribution always skewed towards longer advance lengths. The skew towards longer advance lengths was extremely strong, and consistent among the very first set of (α, β)'s chosen, corresponding to the advance length between to two furthest strobes. For the other set of (α, β)'s, there was still a skew towards the longer advance lengths; however, the skew was not as strong and the degree of the skew was much more varied. We conclude that for the shorter advance lengths among multiple strobes, the *exact* distribution does not have a strong effect, as long as it is skewed towards longer advance lengths.

2.3.5 Regions with a high SNP density

Haplotype assembly is often applied to phase specific regions of interest. Often, these regions are gene-rich, and have a high SNP density. The HLA Region on chromosome 6, contains genes encoding cell surface antigen presenting genes and many other genes involved in the immune system. Diversity in this region is important for host defense against pathogens, and it has been implicated in susceptibility to diseases including diabetes, cancer, and various autoimmune disorders (Aversa et al. (1998); Shiina et al. (2004)). Phasing of coding SNPs could provide critical structural information, motivating the development of haplotyping techniques specifically targeted to this region (Guo et al. (2006)). We specifically looked at the region from position chr6:$29,652$K-$33,130$K, using HuRef data. While increased coverage provides modest improvement, high gains in AN50 are obtained by increasing A (Figure 2.6). At $c = 10\times$, $L = 900$bp, $A = 20$kbp we span 80% of the region with 5 haplotypes.

2.3.6 A short note on haplotype accuracy

While our focus is on the feasibility of generating long haplotypes, accuracy is also an important consideration with next generation technologies that may have undesirable raw read error rates. We used our previously developed tools, HASH, and HapCUT (Bansal et al. (2008); Bansal and Bafna (2008)) to phase haplotypes

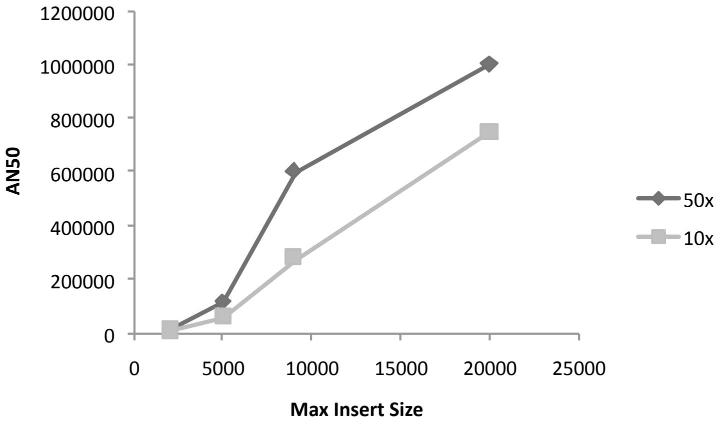


Figure 2.6: **Haplotyping the HLA Region (chr6:29,652K-33,130K) of HuRef.** AN50 increases with maximum insert, and increased coverage. For $c = 10$, $A = 20$K, 5 haplotypes cover 80% of the 3.5Mbp region.

while accounting for error. The Pacific Biosciences simulator was used to generate reads under realistic error models. The simulator takes a single parameter ε as input, reflective of the overall error rate. We chose $\varepsilon \in \{0.05, 0.1, 0.15\}$. As our subreads are long, we assumed correct alignment of all reads (see Section 2.5).

Many homozygous sites appear heterozygous due to missed base calls. For example, we observe 202K heterozygous sites at $\varepsilon = 0.05$ in a region with 936 known SNPs. Using a statistical test for filtering, only 3 of the erroneous sites remain, and none of the 'true' SNPs is eliminated. Table 2.1 summarizes the false negative and false positive rates for $\varepsilon \in \{0.05, 0.10, 0.15\}$, and $c \in \{10\times, 15\times\}$.

For $c = 10\times, \varepsilon \in \{0.05, 0.1\}$, we were able to perfectly assemble the haplotypes. Even with $\varepsilon = 0.15$, we were able to assemble haplotypes with HER= 2.25%, SER=0.76%. Increasing coverage to $c = 15\times$, we achieved HER=1.39%, SER=0.11%. As more data becomes available, we will exploit the error characteristics and related base level quality values to further improve haplotyping accuracy.

Table 2.1: **Filtering Erroneous Heterozygous Sites.** This table shows false positive rates (% of actual SNPs filtered) and false negative rates (% of erroneous sites not filtered) when sites are filtered using the likelihood ratio with a cutoff at -1.

Error Rate, Coverage	**% of False Positives**	**% of False Negatives**
$5\%, 10\times$	0	0.467
$10\%, 10\times$	0	0.197
$15\%, 10\times$	0	0.142
$5\%, 15\times$	0	0.329
$10\%, 15\times$	0	0.149
$15\%, 15\times$	0	0.114

2.4 Discussion and Conclusions

In spite of a long history and success with Sanger sequencing, the feasibility of assembling meaningful haplotypes with next generation sequencing has been questioned. Here, we demonstrate that with a judicious choice of parameters and strobe sequencing, long (and accurate) haplotypes can be effectively generated. The most important parameter appears to be the flexibility in choosing advance lengths, available with strobe sequencing. Even with only $k = 2$ strobes, and coverage $c = 10\times$, we can achieve long haplotypes. On the target HLA region, we covered 80% of the region with 5 haplotypes.

Surprisingly, the optimal design for haplotyping heavily favors longer advances, and the trend does not change with higher values of A, L, c, or number of strobes. Here, we only provide a partial explanation, suggesting that the longer advances level the probability of all edges. A rigorous explanation based on extending the Erdós Renyi theory to the interval-like SNP-Graphs will be the focus of future efforts. Other parameters influence haplotype lengths as well, and our results help determine the optimal values.

Here, we use the 'number of bp sequenced (coverage)' as the "cost" of the design, and optimized parameters after fixing coverage. However, other cost factors might be reasonable. For example, it may be more expensive to generate reads with longer inserts. Also, more biological sample is needed (and wasted) with longer inserts, and that can be a limitation when sample is limited (as in tumors). Our simulated annealing software for optimizing parameters can easily be modified to deal with a custom cost function.

Finally, while haplotype assembly can generate long haplotypes, it is not yet capable of separating entire chromosomes. However, other techniques such as chromosome dissection and amplification can generate long scaffolds connecting distal sites. Used in conjunction with Haplotype assembly on strobe sequences, chromosome level haplotyping is indeed feasible, even without familial information.

2.5 Methods

2.5.1 Data Source

The data source was derived from available human assemblies including HuRef (Levy et al. (2007)). For our simulations we used data from chromosome 1 of the HuRef Genome. While the majority of the experiments were performed on the first 10Mbp of chromosome 1, tests in other regions show similar results. For simulations in the HLA region we used a 3.5Mb interval on chromosome 6.

2.5.2 Simulator

The input to the simulator is a data source, D, containing a list of heterozygous sites and their respective coordinates, and the parameters of the reads $(L, c, A, k, (\alpha_i, \beta_i))$. The S50, N50, and AN50 metrics are output. The algorithm is described below. It simulates subreads as fixed intervals of size L/k, with advances chosen from the appropriate β-distributions. The nodes of SNP-Graph are connected by an edge if a fragment overlaps their locations. The procedure GETSUMMARY computes the different metrics at the end of the simulation.

Algorithm 1 Simulation algorithm

```
procedure SIMULATE($D, L, c, A, k, (\alpha_1, \beta_1), .., (\alpha_{k-1}, \beta_{k-1})$)
    Initialize SNP-Graph by creating a node for each SNP in $D$
    Set $N = \frac{cG}{L}$
    Repeat $N$ times
        Select random start position, $d_0$; Set $S = \phi$
        For $1 \leq i < k$
            Set advance $a_i \leftarrow \mathcal{D}(\alpha_i, \beta_i)$ (* $\beta$-dist *)
            Set $d_i = d_{i-1} + L/k + a_i$
            Add SNPs in intervals $[d_i, d_i + L/k]$ to $S$
        Add edge $(s_i, s_j)$ and edge $(s_j, s_i)$ to SNP-Graph for all $(s_i, s_j)$ in $S$
    $(S50, N50, AN50)$ = GETSUMMARY(SNP-Graph)
end procedure
```

2.5.3 Computing optimal (α, β)

We used a Simulated Annealing (SA) algorithm to compute the optimal (α, β) values. To test the performance of the SA, we also used a slower coarse-grained optimization.

Simulated Annealing (SA): We start with α, β chosen at random from the range $(0, 3.5]$. Empirically, Temperature T was selected to be $11,000$, and reduced by a fixed amount in each iteration. The neighboring solution was selected at random from $\{(\alpha \pm s, \beta), (\alpha, \beta \pm s)\}$. We set $s = 0.5$ for the first half of the iterations, and set $s = 0.1$ for the remaining to allow for finer optimization. This allows for a free exploration of the search space, followed by fine grained optimization at the end. Due to a large variation in AN50 for a fixed (α, β), we recompute AN50 values for the current solution and the neighbor, making it easier to escape an artificially high value. We maintain a list of all $(\alpha, \beta$,AN50) triples observed.

Algorithm 2 Simulated annealing algorithm

```
**procedure** SIMULATEDANEALLING($D, L, c, A, k$)
    Initialize grid, $G(\alpha, \beta)$, a list of observed $AN50$ values
    Set $(\alpha, \beta) \leftarrow (0, 4] \times (0, 4]$
    For all $1 \leq i \leq l$
        Set $s = 0.5$ if $i < I/2$; else Set $s = 0.1$
        $(\alpha', \beta') \leftarrow \{(\alpha \pm s, \beta), (\alpha, \beta \pm s)\}$
        $G(\alpha, \beta) = G(\alpha, \beta) \cup$ SIMULATE($D, L, c, A, k, \alpha, \beta$)
        $G(\alpha', \beta') = G(\alpha', \beta') \cup$ SIMULATE($D, L, c, A, k, \alpha', \beta'$)
        AN50 = median($G(\alpha, \beta)$)
        AN50' = median($G(\alpha', \beta')$)
        Set $T = T - T_0/I$; $\Delta$ =AN50'-AN50
        Move to $(\alpha', \beta')$ with probability $\min\{1, e^{-\frac{\Delta}{T}}\}$
**end procedure**
```

2.5.4 SA performance

We use an exhaustive coarse-grained optimization to check the performance of SA. Each (α, β) pair for $\alpha \in (0, 3.4]$ and $\beta \in (0 - 3]$ was chosen with step sizes of 0.2 and 0.1 respectively. For each value, we performed 25 simulations, and recorded the median AN50. We compared SA and coarse grained optimization for $c = 20\times, L = 900$bp, $A = 9$kbp to match the parameters currently available for strobe sequencing. See Figure 2.2. The coarse grained optimization entails a total of $12,750$ simulations, each about 1CPU min. on a PC. By contrast, SA achieves a finer grained optimization using only 450 simulations. The results are consistent with the two methods (Figure 2.8).

2.5.5 Calling heterozygous sites (SNPs)

After running our simulated fragments through the Pacific Biosciences error simulator and aligning the erroneous fragments (since our data is simulated, we use original fragments to perfectly align the erroneous fragments), we used statistical methods to differentiate heterozygous sites caused by true SNPs versus those caused by error. If a heterozygous site has a coverage of n (n fragments overlap the site), there are n_1 counts of the dominant allele and $n_2 = n - n_1$ counts of the minor allele.

H_0: The heterozygous site has no bias in the two alleles; the two alleles both have a 50% chance of appearing with a small probability of error.

H_1: The heterozygous site always shows one allele with a small probability of error

Let ε be the probability of a miscalled base. Then, the likelihood ratio statistic is given by

$$\Lambda = 2 \ln \frac{P(O|H_0)}{P(O|H_1)} = 2 \ln \frac{(1-\varepsilon)^{n_1} \cdot \varepsilon^{n_2}}{(\frac{1}{2}+\varepsilon)^{n}} \tag{2.2}$$

The likelihood ratio Λ asymptotically approaches the χ^2 distribution. However, we empirically selected $\Lambda = -1$ as the cut-off for calling heterozygous SNPs.

2.6 Appendix

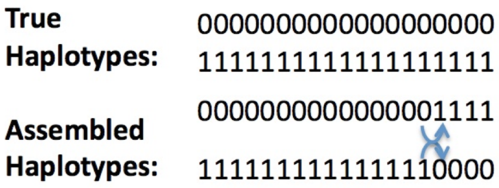


Figure 2.7: **Haplotype Accuracy.** The haplotype edit rate(HER) in this example, given by the fraction of incorrectly called alleles, is $\frac{4}{19}$ while the switch error rate (SER) defined as the number of crossovers per site required to match the correct haplotype is $\frac{1}{18}$.

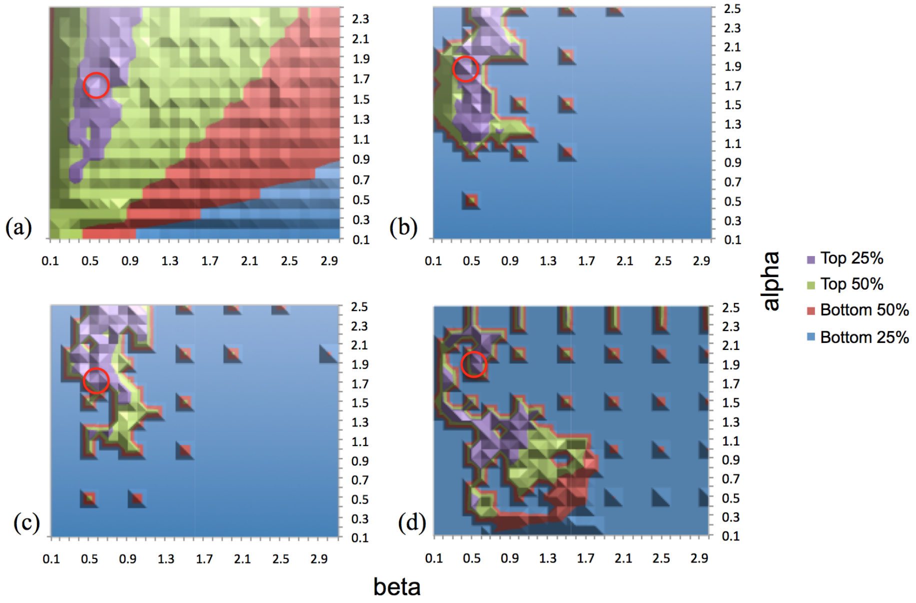


Figure 2.8: **Contour Plots.** Comparison of the contour plot of SA and Coarse grained optimization shows that optimal (α, β) range of both approaches are similar. Different metrics (S50, N50, AN50) also produce similar results. (a) Coarse grain optimization for AN50 (b) SA contour for AN50 (c) SA contour of N50 (d) SA contour of S50. Optimal (α, β) value for each is circled. All simulations were performed on the first 10Mbp of chr1 (HuRef) using $L = 900$bp, $c = 20$x, $A = 9$kbp.

Table 2.2: **Simulated Annealing Results for Figure 2.5.** The following tables show the optimal AN50 and the (α, β) values found by simulated annealing. All the optimal β-distributions are similar and skewed towards longer advance lengths.

(a) AN50 and Optimal (α, β) for Figure 2.5a

Max Adv Len (kbp)	**AN50**	(α, β)
9	161607	(1.9,0.7)
20	900068	(3.8,0.8)
30	2804943	(2.6,0.6)
40	4389053	(2.8,1.0)
50	10658401	(3.6,0.7)

(b) AN50 and Optimal (α, β) for Figure 2.5b

Coverage	**AN50**	(α, β)
10	84816	(2.6,1.0)
20	152625	(1.7,0.6)
30	158652	(2.1,0.7)
40	166004	(3.1,1.0)
50	195282	(1.6,0.5)
60	226540	(1.1,0.4)
70	226540	(1.8,0.9)
80	226540	(3.6,0.6)
90	226540	(3.2,0.9)

(c) AN50 and Optimal (α, β) for Figure 2.5c

Read Length	**AN50**	(α, β)
300	60593	(2.7,0.8)
500	117657	(3.3,0.7)
700	133155	(2.4,0.8)
900	144690	(0.9,0.5)
1100	155397	(2.3,0.6)
1300	157248	(2.5,0.6)
1500	158652	(1.4,0.4)
2000	166004	(1.4,0.5)
4000	166004	(3.1,0.5)

(d) AN50 and Optimal (α, β) for Figure 2.5d

Num of Strobes (kbp)	**AN50**	$(\alpha_1, \beta_1) \ldots (\alpha_k, \beta_k)$
2	88084	(1.8, 0.4)
3	108110	(2.7, 0.3), (2.0, 1.6)
4	110740	(2.9,0.1), (2.9, 1.0), (0.3, 0.5)
5	93870	(3.5, 0.1) (2.7, 0 .3), (2.7, 2.0), (0.8, 1.6)
6	86836	(3.1, 0.6), (3.2, 1.1), (2.9, 1.6), (2.6, 2.5), (2.5, 2.5)

2.7 Acknowledgements

Chapter 2, in part, is a reprint of the material as it appears in BMC Bioinformatics 2011. Lo C, Bashir A, Bansal V, and Bafna V. “Strobe sequence design for haplotype assembly.” The dissertation author was the primary investigator and author of this paper.

Chapter 3

Haplotyping with Clone-based Sequencing

3.1 Introduction

The basic principle behind clone-based haplotyping (Burgtorf et al. (2003)), involves constructing clone libraries that will extract long subsections of a haploid and pooling together several clones for sequencing. As long as the clones within a pool do not overlap, the clones can be computationally reconstructed from shorter sequencing reads and assembled into longer haploid sequences. Alternative implementations of the clone-based haplotyping method (Kitzman et al. (2011); Suk et al. (2011); Peters et al. (2012)) mainly differ in how clones are generated (affecting the length of the clones) and the number of pools sequenced. For example, in a study by Suk et al., fosmid clones with an average length of 40kbp were combined into 288 pools, with $5,000$ clones per pool, and the N50 length of the assembled haplotypes was 1Mbp (Suk et al. (2011)). Several conceptually similar haplotyping methods have recently been reported that fragment genomic DNA in vitro and then pool together the fragments for sequencing. From example, the long fragment read (LFR) method was used in one study (Peters et al. (2012)) to generate haploid fragments of length (L) 10k to 300kbp, which were combined into 384 pools with around 5,000 to 10,000 fragments per pool. The resulting N50 length of the assembled haplotypes ranged from 400k to 1Mbp. Another method generated haploid fragments of average length (L) = 13.8kbp, leading to haplotypes with comparable N50 values (Kaper et al. (2013)). Most recently, research using Moleculo technology (Illumina Inc., San Diego, CA, USA) reported fragments of $L = 6$k to 8kbp, possibly reaching up to 10k (Voskoboynik et al. (2013)).

The differences in the experimental designs of these studies directly affect the cost versus haplotype length trade-off. Previous clone-based haplotyping experiments did not explicitly consider how their parameter choices affect the cost versus haplotype length trade-off, and often used the same design criteria as those used for sequence assembly. Note that there is a major difference between sequence assembly and haplotype assembly; sequence assembly relies on partially overlapping short sequences of typically 20 to 70 bp in length, whereas haplotype assembly depends on multiple adjacent heterozygous variants at a typical spacing of 1.5kbp, which is a much more stringent requirement.

Here, we pursue a parameterized approach to haplotype assembly. We considered the following parameters: clone length (L); number of clones per pool (n); number of pools (p); and sequence read coverage per pool (r). The use of parameters allowed us to compare the different methods (Table 3.1) and understand the effects of haplotype length on different parameters. Please note that we use the term *clone* in this paper because we were designing explicitly for clone-based haplotyping; however, a *clone* can refer more generally to any type of haploid subsequence of the genome, regardless of how it was obtained (that is, in vitro or in vivo).

We started with the assumption that once the clone library is prepared, the cost is simply a function of the number of pools (p). However, our calculations are useful for understanding other trade-offs as well. Although intuitively, larger clone size (L) leads to longer haplotype length, the assembled haplotype length depends upon the combination of parameters L, n, p, and r in a non-trivial fashion. Connecting overlapping haploid clones generates long haplotypes. As mentioned above, haplotype assembly is unlike sequence assembly, where L only needs to be long enough to span the longest repetitive sequence. For haplotype assembly, the clones must be long enough to span adjacent heterozygous variants. Therefore, the first step in a good design is to make L as large as possible within the constraints of available technology and cost. Next, to maximize the chance of getting overlapping clones, the total clone coverage ($c = \frac{nLp}{G}$) should be maximized. At the same time, overlapping clones within a pool may lead to heterozygous calls that are not informative for haplotyping, implying that clone coverage within a pool ($c_p = \frac{c}{p}$) should be kept low. The naive way to accomplish these design objectives is to keep n low and p high, which in turn, increases the cost of the experiment.

We studied the p versus length trade-off by simulating clones under different experimental parameters, and assembling haplotypes assuming a known distribution of variants. To experimentally validate the effect of using larger clone size (L), we performed an experiment with $p = 24$ pooled bacterial artificial chromosome (BAC) clones from a Caucasian male sample, NA20431, of the Personal Genome Project (designated PGP1). BAC clones are longer ($L = 100$k to 300kbp) (Shizuya et al. (1992)) than fosmid-based clones ($L = 40$kbp) and LFRs ($L = 60$kbp). To keep sequencing costs low, we used additional reads from single low-pass WGS in addition to modest coverage in each pool. Even with low sequencing cost (p and sequence coverage), we identified the assembly of accurate haplotypes that were more than 2.5 times longer than previously reported ones. Our results also suggest important design principles for clone-based haplotype assembly: for long haplotyping, L should be as high as possible within the bounds of technology; it is possible to achieve long haplotypes with a much smaller p (and hence lower experimental cost) than was implemented in the previous efforts; and, there is a direct trade-off between depth of sequencing per pool (r), and the length and resolution of haplotypes (fraction of variants phased), but modest sequencing depth is sufficient.

In addition to length, we also took into account the accuracy of the generated haplotypes. The two types of errors that can arise in haplotyping are mismatches and switches. Mismatches are defined as single nucleotide differences between the assembled haplotype and the true haplotype, and are probably caused by erroneous base calls. Switch errors are defined as positions where a crossover in haplotype orientation is needed to recover the true phase. To test the accuracy of the haplotypes, we need to compare the generated haplotypes with the true haplotypes (that is, haplotypes from trio data). However, the true haplotypes are not always known, and without knowledge of the ground truth, the best we could do is compare with haplotypes obtained via other methods.

3.2 Results

3.2.1 Design of experiment for clone-based haplotyping

To assemble long and accurate haplotypes, the clone coverage, c, (that is, the average number of clones covering each genomic position) must be high enough so that overlapping clones from different pools can be assembled to form longer haplotypes. For a genome of length G, the expected clone coverage is given by $c = \frac{nLp}{G}$. Assuming that clones of fixed length L arrive at random and overlapping clones come from different pools, the overlapping clones assemble into longer contigs. The Lander and Waterman estimate (Lander and Waterman (1988)) for expected length of a contig is given by:

$$E(\text{contig length}) = \frac{L(e^c - 1)}{c} \tag{3.1}$$

3.2.2 Effect of clone length (L)

The Lander and Waterman estimate allows for a quick comparison of different strategies, and suggests that increasing clone coverage can compensate for low L (see Figure 3.6). However, this does not model an important aspect of both sequence and haplotype assembly. In sequence assembly, L must be long enough to span repeats in order to permit unambiguous assembly. Once L exceeds the length of known repetitive sequence (about 10kbp for humans in order to span long interspersed nuclear elements (LINEs)), increasing L further has diminishing returns (Chaisson et al. (2009)). However, in haplotype assembly, overlaps are informative for phasing only when they cover heterozygous variants. If two adjacent heterozygous variants are further apart than the length of a clone, they cannot be linked into a single haplotype. Based on the observed variant distribution in the human genome, no saturation is seen even with very high values of L (Figure 3.1a). Thus, L must be chosen as high as is technologically possible. We show the affect of long clone size on haplotype assembly by using available BAC clones (140kbp), which are longer than clones from previous approaches (Table 3.1).

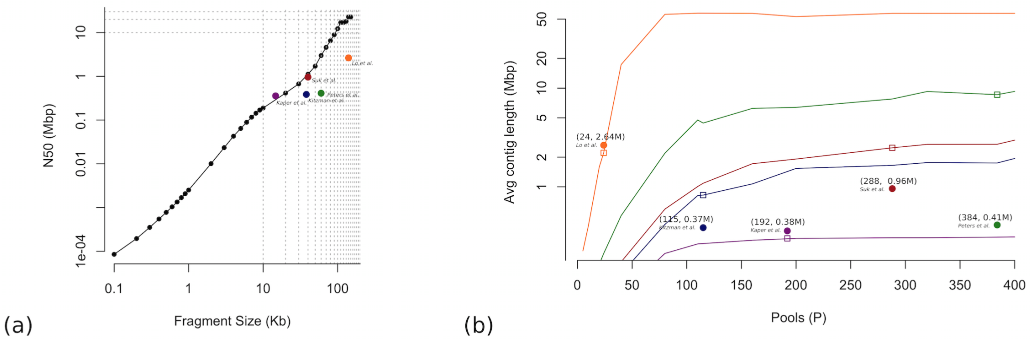


Figure 3.1: **Expected contig length for various clone-based haplotyping designs.** (a) Log-log plot of the maximum achievable haplotype N50 length for different values of clone length (L) (assuming a distribution of heterozygous variants obtained from Complete Genomics Institute (CGI) whole genome sequencing (WGS) on chromosome 1 of sample NA20431 of the Personal Genome Project (designated PGP1). This plot suggests a power law relationship between haplotype N50 length (N50) and clone length (L), which is characterized by N50 being approximately $L^{1.42}$. Note that achieved haplotype lengths (filled circles) may not reach the maximum length, owing to smaller numbers of pools or low fraction of the variants recovered. (b) Simulated haplotype length versus the number of pools (p) for given values of L (shown in different colors). In all cases, except one (magenta), the number of clones per pool (n) is 5,000 (n = 16,800 for magenta). The curves reach saturation when all variants that are less than distance L apart are connected in a contig. Simulations are performed using the distribution of heterozygous variants obtained from CGI WGS on chromosome 1 of PGP1. The squares represent the simulated estimate given parameter settings of several clone-based haplotyping experiments, while the circles show the reported N50.

3.2.3 Effect of pool number (p)

The clone coverage must be high in order to form long contigs, but overlapping clones within a pool result in heterozygosity, and are not informative for haplotyping. Denoting the coverage per pool as $c_p = \frac{c}{P}$, the probability of overlap for a clone is given by:

$$P_o = 1 - e^{-2c_p} \tag{3.2}$$

Previous clone-based methods, (Kitzman et al. (2011); Suk et al. (2011); Peters et al. (2012)) (Table 3.1) all kept c high and c_p low by keeping p high and nL low. As each pool must be sequenced independently, the cost increases linearly with p. To keep sequencing costs low, we considered the effect of overlaps within a pool explicitly. As a first approximation, we simply discard clones that overlap within a pool. Thus, the number of clones per pool is reduced to $n' = n(1 - P_o)$, yielding a new coverage of $c' = \frac{n'Lp}{G}$. Figure 3.1b shows this "p (or cost) versus contig length" trade-off, and clearly shows that the previous approaches (denoted by circles) used many more pools than necessary for their specific clone length choices. Here, we worked with a relatively low value of $p = 24$, which kept costs low. We additionally improved haplotype contig lengths by not discarding overlapping clones in a pool, but separating them computationally (see Methods below).

Another consideration in the design is the recovery of heterozygous variants. For haplotyping, the heterozygous variants of the individual must be linked, and therefore the variant must be sampled from both parental chromosomes. By contrast, the homozygous variants (reference or non-reference) can be filled in subsequently, and it is only necessary to sample the variant on one chromosome. The expected percentage of heterozygous variants that are sampled by clones from both chromosomes is given by:

$$p_v = 1 - 2e^{\frac{-v}{2}} \tag{3.3}$$

We worked with low values of $c = 6$x, thus we expected only 90% of the heterozygous variants to be recovered. To recover more heterozygous variant locations, we augmented the detected heterozygous variants by using additional WGS data of the same individual.

3.2.4 Effect of sequence read coverage per pool (r)

The final parameter of interest is the read coverage per pool, r. Increasing r increases the sequencing cost per pool, but low values of r can affect clone reconstruction, and thus haplotype length and resolution. For example, low values of r decrease the resolution of the clones (that is, not all bases spanned by a clone will be covered by a read) and make it difficult to detect clone boundaries. At the same time, increasing r has diminishing returns for increasing cost. In particular, assuming a Poisson distribution with parameter r, the probability that a position is covered by k reads is given by:

$$p_r \approx \frac{e^{-r} r^k}{k!} \tag{3.4}$$

For $r = 6$x, Equation 4 suggests that 84% of the base pairs spanned by a clone are covered by four or more sequence reads. However, the actual coverage (see Figure 3.7) suggested that the coverage distribution is not Poisson. In fact, only 65% of the base pairs spanned by a clone were covered by four or more sequence reads. The bias in coverage could be attributable to a variety of factors, such as amplification bias and filtering of reads in order to control for repeats. To capture all these factors (including r, amplification bias, filtering of reads, and variants), we worked directly with the parameter f (the fraction of heterozygous variants recovered in a clone). In our experiments, r was 6x and f was 65%. We studied the effects of f on haplotype length and haplotype-resolution via simulations (see Figure 3.8, 3.9), and our results showed that modest values of f (or sequencing depth r) and p could be used to achieve long haplotypes with high resolution.

3.2.5 BAC pool construction

Our analysis suggested that using a limited number of pools of larger clones can lead to longer haplotypes than using many pools of smaller clones or fragments. To provide experimental support for this prediction, we implemented the clone-based haplotyping strategy using a set of BAC clones. We started with existing BAC clones constructed from high molecular weight PGP1 genomic DNA and individually maintained in 384-well plates for other purposes. Given that the mean length of a BAC clone is 140 kbp (L), 384 BAC clones in one plate amount to 54 Mbp, approximately 1.7% of the 3.2 Gbp human haploid genome. Additional pooling of fourteen 384-well plates, containing a pool of 5,376 (n) BAC clones, would be expected to cover about a quarter of the human genome. With a total of 24 (p) pools, the expected clone coverage (c) was 6x. The PGP1 BAC library was constructed for multiple purposes and maintained as one clone per well in 384 wells, which involved a high cost (>US $\$50,000$) for handling individual wells. For haplotyping purposes, we estimated that making pooled BAC libraries from genomic DNA without individual colony picking and maintenance would costs approximately US$5,000, and preparing DNA from each pool would cost roughly US$20. Therefore, to implement this BAC-based approach routinely, the total cost involved in BAC library construction and preparing DNA from 24 BAC pools

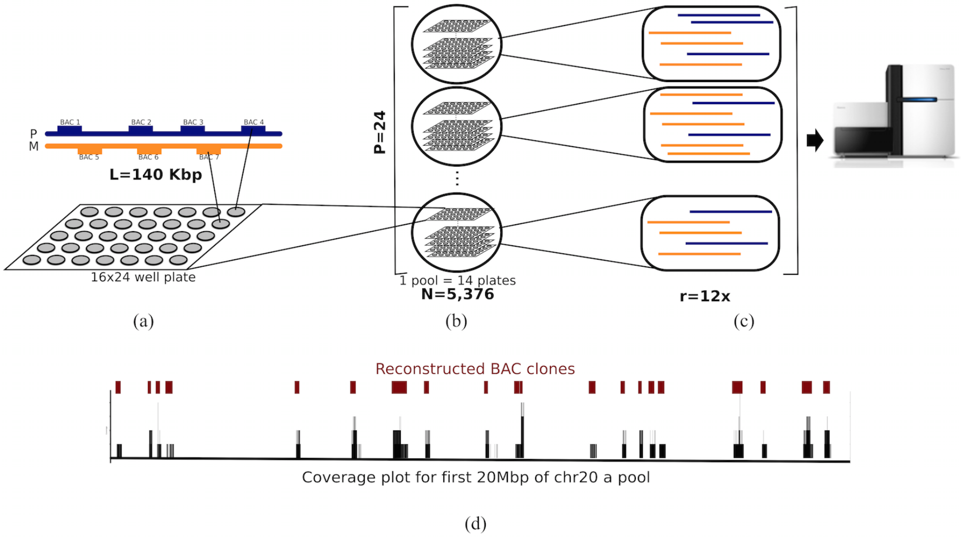


Figure 3.2: **Haplotyping with bacterial artificial chromosome (BAC) clones.** (a) Constructing the BAC library. DNA was extracted from PGP1 (NA20431, Personal Genome Project) and BAC clone libraries with clone length ($L = 140$kbp. (b) Forming pools of BAC clones. The number of pools (p) formed was 24, with each pool consisting of fourteen 16 24-well plates, so that there was a total of $n = 5,376$ clones per pool. (c) Sequencing and mapping each pool. Sequencing libraries were prepared for each pool with a read coverage of $L = 6$x. After sequencing, reads were mapped to hg19. (d) Reconstructing BAC clones. Clones were reconstructed from the mapped reads of each pool using coverage-based techniques (clones detected in region of chromosome 20).

would be roughly US$5,480.

Following this design (Figure 3.2), we constructed 24 sequencing libraries from the 24 pools, which collectively contained 129,024 BAC clones. A total of 2 billion pair-end 100 bp reads were generated for these libraries, with an average of approximately 74 million reads for each pool. Of these, roughly 47 million reads (63.5%) were uniquely aligned to the genome, giving an effective read coverage (r) of 6x per pool.

3.2.6 Reconstruction of BAC clones

In each pool, the boundaries of BAC contigs were determined by detecting regions of enriched read coverage after the reads of the pool had been mapped to the genome (Kitzman et al. (2011)) (see Methods). If clones in a pool do not overlap, a BAC contig will contain only one BAC clone, and the boundary of a BAC contig will be the boundary of a BAC clone. However, given that $c_p = 0.25$ in our experiment, we estimated that the percentage of BAC contigs containing more than one clone would be $P_o = 39.34\%$. When clones overlap, it is not possible to assume that the consensus sequence of the BAC contigs provides haplotype information.

With the goal of maintaining the haploid nature of each pool, we developed a computational approach to detect and remove regions covered by more than one clone (see Methods). Previous clone-based methods detected and removed overlaps by finding heterozygous variants and either removing the whole contig (Kitzman et al. (2011)) or breaking the contigs at those locations (Suk et al. (2011)). These methods were sufficient for previous methods because P_o was relatively low (Table 3.1). However, we developed a more sophisticated method to detect and remove only the overlapping regions of a contig. Briefly, our method first detects the boundaries of overlapping regions by searching for bulges in coverage. Using these boundaries, we removed regions of the contigs that contain a significant fraction of heterozygous variants, as these probably represent regions of overlapping clones.

Before removing the overlap regions, we detected a total of 92,937 BAC contigs with an average length of 161,397 bp (N50 = 199,744 bp). This is consistent with estimates derived from Lander and Waterman statistics for the expected number and length of contigs (100,396 contigs and 159,127 bp). After removing the overlap regions, there were a total of 85,445 reconstructed clones with an average

Table 3.1: **Comparison of different clone-based haplotyping protocols.**

	Kitzman et al. (Fosmid)	Suk et al. (Fosmid)	Peters et al. (LFR)	Kaper et al.	Lo et al. (BAC)
N: Number of clones in pool	5,000	5,000	5,000-10,000	16,377	5,000
L: Exp(clone length) (Kbp)	37	40	60	13.8	140
P: Number of pools	115	288	384	192	24
c: Exp(clone coverage) = $\frac{nLp}{G}$	7.1 1	9.2	57.6	14.5	6.0
c_p: Exp(clone coverage per pool) = $\frac{nL}{G}$	0.06	0.07	0.15	0.075	0.25
P_o: overlap probability = $1 - e^{-c_p}$	11.31%	13.06%	25.92%	13.93%	39.35%
Exp(haplotype length) (bp)	2.05E7	4.37E10	5.30E16	4.89E9	3.42E5
Simulated haplotype length (bp)	825,046	2,486,692	8,585,663	300,336	2,210,343
Actual haplotype length (bp)	386,000	959,175	411,000	358,000	2,640,036

length of 140,777 bp (N50 = 161,300 bp). Note that some contigs had to be removed completely because the non-overlapping portions of the clones could not be recovered (see Figure 3.10 for the distribution of lengths for the final reconstructed BAC clones).

3.2.7 Variant detection

To recover variants, we pooled together all the sequence data from the 24 pools and called variants using BWA/ GATK software (McKenna et al. (2010a)) (see Methods). A crucial part of phasing is variant calling, and more specifically, differentiating heterozygous and homozygous variants. However, our method could call heterozygous variants only where both haploids were covered by BAC clones. For instance, if only the haplotype with the reference allele of the variant was covered, the variant would not be called. Likewise, if only the haplotype with the non-reference allele of the variant was covered, the variant caller would not be able to confirm its zygosity, and it would be called homozygous by default and be discarded for phasing. Of the 2,906,810 variants recovered, 1,287,220 variants were called as heterozygous and 1,619,590 were called as (non-reference) homozygous.

To overcome the challenges caused by low clone coverage, we augmented the recovered variants by using existing Complete Genomics Institute (CGI) WGS data of PGP1 [13,20]. A total of 3,283,326 variants were called by CGI (Drmanac et al. (2010)). Of these variants, 2,086,302 were heterozygous and 1,196,934 were homozygous. A total of 3,208,817 variants were recovered by augmenting the variants detected by the pooled BAC data with those detected by CGI WGS data, and of these, 1,942,116 were classified as heterozygous (see Methods) and used in the haplotype assembly. When compared with dbSNP135, 3,083,460 (4%) of the 3,208,817 variants were found to be novel. The percentage of novel variants, the number of variants, and the homozygous to heterozygous ratio were comparable with other individuals of European descent (Table 3.2, 3.4).

Table 3.2: **PGP1 statistics.** Variant statistics for PGP1 compared to others individuals of European descent.

	Total Variants	Hom/Het Ratio	% Novel Variants
PGP1 (BAC Pools)	2,906,810	1.26	1%
PGP1 (CGI WGS)	3,283,236	0.57	4.9%
PGP1 (BAC + CGI WGS)	3,208,817	0.71	4%

3.2.8 BAC haplotype assembly

Haplotype contigs were assembled by chaining together heterozygous variants that were connected by a BAC clone; the more overlapping clones (that is, higher clone coverage) present, the longer would be the expected haplotype length. Given the number of reconstructed BAC clones (n' = 85,445), and their average length (L' = 140,777), the effective clone coverage, $c' = \frac{n'L'p}{G}$, was 4x. Previous methods report clone coverage of 6.6x (Kitzman et al. (2011)), 12.56x (Suk et al. (2011)), and $38-56$x (Peters et al. (2012)). Although the clone coverage for this BAC haplotyping experiment was lower, we achieved longer haplotypes because of the longer length of the BAC clones. In total, 2,379 haplotype contigs were assembled to form haplotypes with an N50 length of $2,640,036$ bp. The chromosome level breakdown of the number of contigs and N50 lengths is shown (Table 3.3), and the distribution of the haplotype lengths is provided (see Figure 3.11). The longest haplotype contig spanned over 14Mbp.

Table 3.3: Chromosome level breakdown of haplotype statistics.

Chrom.	# Clones	# Het. variants (BAC + CGI WGS)	Fraction of het. variants phased	# Contigs	N50 Haplotype Length
1	6,892	148,691	0.973	219	2,166,488
2	7,632	156,474	0.978	195	2,905,522
3	6,677	147,498	0.985	146	3,015,997
4	6,937	159,704	0.982	106	4,113,256
5	6,101	127,746	0.983	117	2,700,783
6	5,818	142,863	0.981	119	3,005,140
7	4,781	113,746	0.974	150	2,621,008
8	4,589	98,664	0.982	114	2,413,181
9	3,485	80,480	0.966	125	2,276,528
10	4,104	99,111	0.969	109	2,264,959
11	4,233	90,557	0.979	113	3,368,062
12	4,150	93,795	0.979	116	2,641,808
13	3,395	75,463	0.984	51	2,945,506
14	2,846	62,844	0.967	71	2,067,670
16	2,031	59,080	0.956	123	1,252,516
17	1,965	47,238	0.96	116	1,485,600
18	2,547	56,853	0.981	43	3,345,667
19	1,155	32,849	0.956	110	660,242
20	1,649	39,551	0.958	66	1,608,505
21	1,170	34,095	0.958	36	3,226,907
22	768	19,020	0.956	65	1,057,117
TOTAL	85,445	1,942,116	0.975	2,379	2,640,036

3.2.9 Accuracy

We used the minimum edit score (MES) to measure the accuracy between two independently derived haplotypes. The MES takes into account the two common error modes for haplotype assembly- mismatches and switch errors. When comparing two haplotypes, an error can be classified as either a mismatch or a switch. Given the cost of a mismatch (c_m), the cost of a switch (c_s), the number of mismatches (m), and the number of switches (s) the total cost is given by

$$\frac{mc_m + sc_s}{\#\ \text{var}} \tag{3.5}$$

Under the MES criterion, the objective is to classify each error as a mismatch or a switch, such that the total cost is minimized. For example, if there are 10 consecutive errors, these can either be classified as 10 mismatches or 1 switch. If $c_s < 10 * c_m$, then under the MES objective, these errors would be classified as one switch. The classification of errors as mismatches or switches will depend on the cost. In the calculations of this paper, we used $c_m = 1$ and $c_s = 1$.

We tested the accuracy of our haplotypes (henceforth referred to as BAC haplotypes) by comparing them with the haplotypes of PGP1 constructed using the LFR clone-based method (LFR haplotypes) and a population-based method. The population-based haplotypes were computed with BEAGLE (Browning and Browning (2009))using CGI WGS genotype data of PGP1 and population data from the 1000 genomes project (Abecasis et al. (2010b)). In chromosome 1, the MES between the BAC and LFR haplotypes was relatively lower (0.003) than the MESs involving population-based haplotypes (LFR = 0.012, BAC = 0.017) (Figure 3.3). The small discrepancy between two haplotypes could be an error in either the LFR or in the BAC haplotypes. Specifically, haplotype errors are caused by the improper linking of heterozygous variants or errors in variant calling, both of which can happen if there are not enough clones spanning a particular site.

We computed the clone coverage at discrepant sites and found that 95% of the discrepant sites were covered by three or more clones, and there was no correlation between discrepancy and clone coverage (see Figure 3.3). Furthermore, we compared the accuracy between the BAC haplotypes and BAC clones. Of the $358,697$ overlapping variants, there were $1,486$ mismatches and 353 switches, giving an MES of 0.005. The small percentage of mismatches (0.41%) and switches ($< 0.1\%$) can be attributed to sequencing error and errors in clone reconstruction, respectively. Owing to the clone coverage of 6x, most of the errors are recovered during haplotype assembly (Figure 3.4). Our results therefore suggest high accuracy for the computed BAC haplotypes.

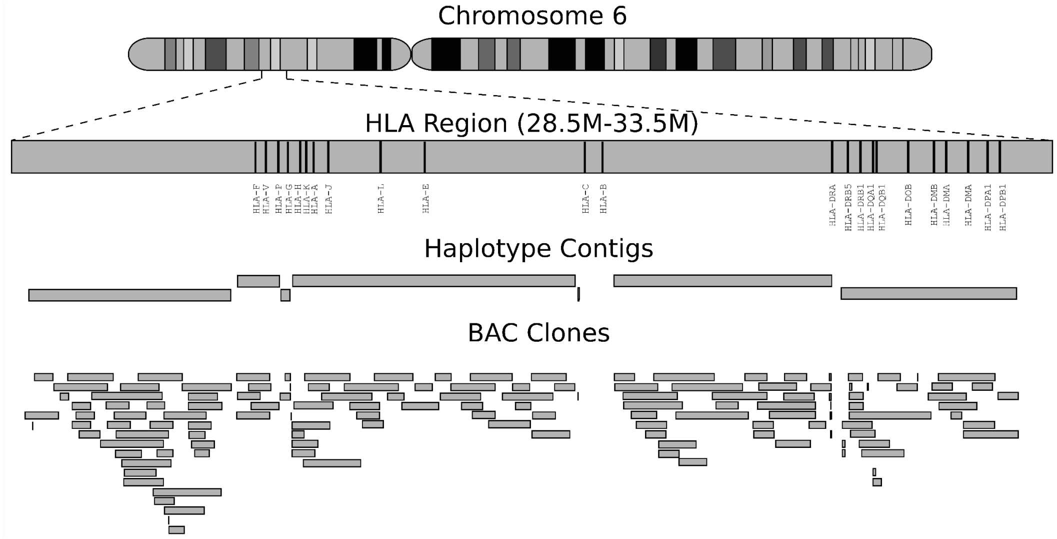


Figure 3.3: **Accuracy comparison between bacterial artificial chromosome (BAC), long fragment read (LFR), and population-based haplotypes.** The MES score is given by the classification of errors as mismatches or switches such that $c_m = c_s = 1$.

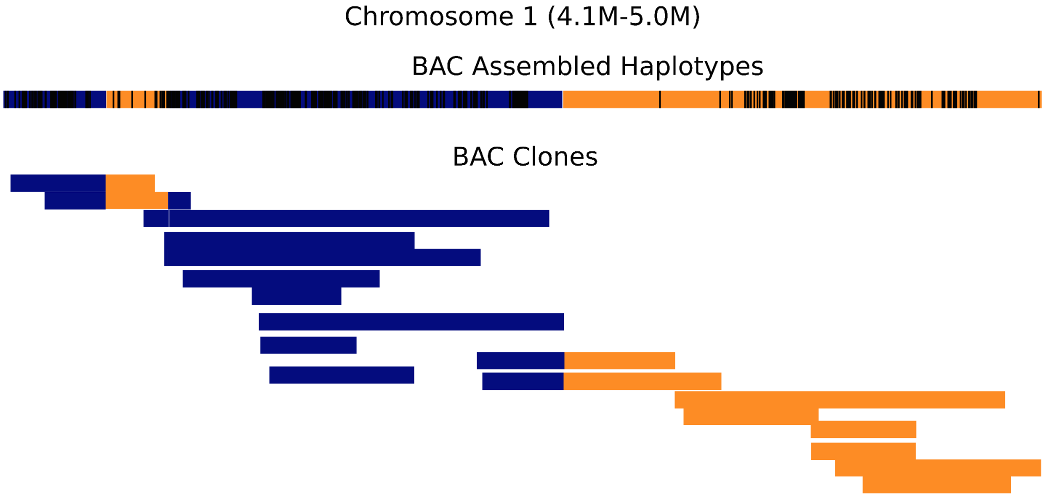


Figure 3.4: **Consistency between BAC-assembled haplotypes and BAC clones.** A snapshot of a 1Mbp region on chromosome 1 illustrating three switch errors between BAC-assembled haplotypes and the population-based haplotypes (indicated by three color changes along the haplotype). At all three switches, 100% of the BAC clones that span this switch are consistent with the BAC-assembled haplotypes. The heterozygous variants that are phased are represented as black vertical lines in the BAC-assembled haplotypes.

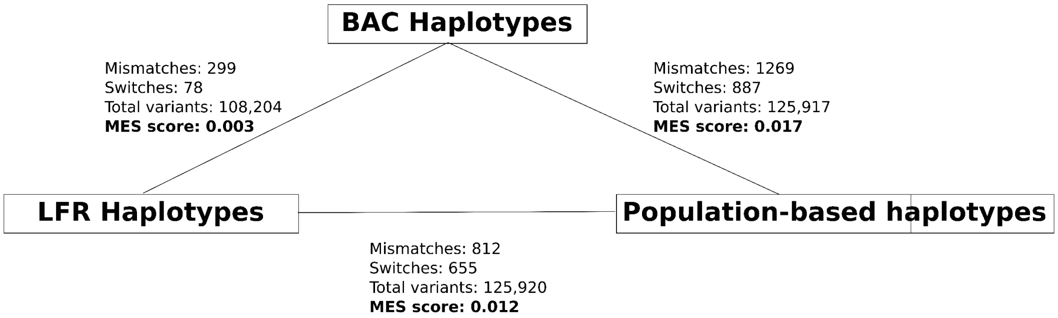


Figure 3.5: **Haplotypes of the human leukocyte antigen (HLA) region.**

3.2.10 Haplotyping the HLA region

The human leukocyte antigen (HLA) region is a 5Mbp region on chromosome 6 that contains many genes that have important regulatory roles in the immune system. Haplotype information of the HLA regions is medically relevant because the specific combination of certain alleles is known to be linked with several autoimmune and other diseases. Owing to the repetitive nature of the HLA region, the haplotypes here are difficult to obtain with current next-generation sequencing technology. However, the length of BAC clones can be used to span over these repetitive regions and connect many more genes, achieving long, accurate haplotypes. In our experiment, the 5Mbp HLA region was covered by 145 BAC clones, which assembled into 7 haplotype contigs, similar to a previous study (Kaper et al. (2013)). More than 90% of the entire HLA region was spanned by six contigs, and the longest haplotype contig in this region spanned 1.37Mbp ($N50 : 1.1$Mbp). Figure 3.5 shows the BAC clone coverage of this region. Of 23 HLA genes, 20 were spanned by BAC clones; 18 of these were phased completely ($> 90\%$ of variants phased in 1 haplotype block), and 2 were partially phased. In addition, 96.7% ($11,861$ of $12,272$) of the heterozygous variants in this region were phased (see Table 3.5).

3.3 Discussion and Conclusions

In our parameterized analysis of clone-based haplotyping methods, the current bottleneck for achieving long haplotypes was the clone lengths (L). Because of the distribution of variants, adjacent variants that are longer than L can never be spanned, and thus the haplotype lengths saturate when all variants within a distance greater than or equal to L from each other are connected.

The importance of L is illustrated in Figure 3.1a, which shows a power law relationship between haplotype length and L. Furthermore, it illustrates the current gap between in vitro technologies for isolating DNA and clone-based methods. As shown, when $L = 10$kbp (current limit on reported length of Illumina's Moleculo technology (Voskoboynik et al. (2013))), the maximum achievable haplotype length is 188kbp. Meanwhile, clone technologies have the potential to achieve significantly longer haplotypes with N50 lengths of 1.12Mbp ($L = 40$kbp, fosmid clones) to 22.8Mbp ($L = 140k$ bp, BAC clones). The importance of L is not just limited to clone-based and dilution-based methods; haplotyping using sequence reads can be modeled using our framework by setting L as the read length, $n = 1$, and p as the number of reads sequenced. For example, long reads were used to assemble haplotypes on the HuRef genome (Levy et al. (2007)). The HuRef genome used a more complex paired-end Sanger sequencing ($c = 7.5$x) protocol with mixed insert sizes (L) and achieved haplotype lengths of N50 = 350kbp. More recent methods (Schadt et al. (2010)) use a single molecule approach to achieve long reads. The importance of L is further illustrated in Figure 1b, as other clone-based methods are well into the saturation levels of their corresponding expected contig-length curve. We concluded that it was more effective to increase L and use a moderate p. In our experiment ($L = 140$kbp, $p = 24$), we achieved haplotypes that had comparable accuracy to leading clone-based methods, and were more than twice as long, with an N50 length of 2.6Mbp and the longest haplotype spanning over 14Mbp. By contrast, the LFR haplotypes, derived from shorter clones ($L(N50) = 60$kbp) and more pools ($p = 384$), had an N50 length of $411k$ bp for the same individual (Peters et al. (2012)).

By reducing p, the total cost of sequencing and clone library construction was reduced, but clone coverage was also decreased. Although we were able to compensate for the small clone coverage in terms of haplotype length by using larger L, our lower coverage recovered fewer variants compared with WGS experiments from other individuals of European descent (see Table 3.4). This low clone coverage also decreased the probability of recovering a heterozygous variant (p_v) and may explain the higher homozygous to heterozygous ratio for BAC pool data. However, the variants could be augmented by acquiring WGS data from the same individual.

The final parameter affecting haplotype resolution and thus length is f, the fraction of variants recovered per clone, which is affected by many other factors such as read coverage per pool (r), amplification bias, and the filtering protocol for reads and variants. The discrepancy between simulated and actual haplotypes lengths in Figure 1b may be due to different values of f (see Figure 3.8,3.9). For example, it is not surprising that the most discrepant results are from the LFR experiment, which used low values of sequencing coverage ($r < 2$x, in contrast with our protocol where $r = 6$x), causing a smaller value of f, which in turn decreases haplotype lengths.

To test accuracy, we performed a three-way comparison between the BAC, LFR, and population-based haplotypes (Figure 3.3). The high concordance between the BAC and LFR haplotypes suggests that both methods have similar accuracy. The higher MES between the clone-based and population-based haplotypes could be due to a variety of factors, including limited population sample size and limited burn-in iterations run by the algorithm due to limited computational resources. Furthermore, population-based haplotypes have difficulty phasing rare, individual-based, and somatic variants. Upon further examination of the population-based haplotypes, we found that the positions of the switch errors correlated with positions where the BEAGLE algorithm had difficulties deciding which phase assignment to choose. The biological implications of these regions have not yet been studied and could possibly represent undiscovered recombination hot spots, or simply areas where the population data are weak. In summary, clone-based haplotypes can be used to provide accurate, megabase-long haplotypes.

Through the integration of statistical modeling and experimental validation, we found that long-range connectivity encoded in large clones or DNA fragments is crucial for constructing long haplotypes. We also provide a practical guideline on the parameter choices and expected haplotype sizes for further design and development of haplotyping methods.

3.4 Methods

3.4.1 BAC library construction and pooling strategy

BAC libraries were constructed with an average length of approximately 140 kb from genomic DNA of the PGP1 sample by Amplicon Express (Pullman, WA, USA). BAC clones were grown in separate wells on 16 24-well plates. The 384 clones on a plate were then combined to form a mini-pool via a two-dimensional pooling strategy, as described by Oeveren et al. (van Oeveren et al. (2011)). Briefly, the strategy combines clones on a plate by rows and columns using a liquid handling robot (Biomek 2000; Beckman Coulter, Brea, CA, USA). Super-pools were formed by further combining clones from 14 mini-pools so that each super-pool contained a total of 5,376 BAC clones. On average, 150 to 250 ng of high-quality DNA was purified in each super-pool by applying DNA isolation via a modified alkaline lysis DNA extraction protocol (Maniatis (1982)). In total, 24 super-pools were constructed.

3.4.2 Construction of sequencing library and variant calling

DNA derived from an individual super-pool was precipitated with ethanol and dissolved in water, then used (10 ng) for random fragmentation by Tn5 transposon based fragmentation method (Epicenter, Madison, WI, USA). Fragmented DNA was purified by Ampure XP beads (Beckman Coulter) and attached with illumina adaptors by PCR amplification to construct sequencing libraries. Barcoded libraries were pooled for sequencing using a HiSeq 2000 instrument (Illumina).

Sequencing libraries were constructed for 24 pools. The resulting sequencing data were processed for variant calling using an established pipeline based on BWA/GATK, following the GATK best practices instructions (version 3). All raw sequencing data have been deposited to NCBI Sequence Read Archive under the project number SRP029150.

The goal for variant calling is to recover all the heterozygous variants for phasing. In particular, a heterozygous variant can fall into one of four categories: 1) both alleles are sampled by clones, 2) only the non-reference allele is sampled by clones, 3) only the reference allele is sampled by clones, and 4) none of the alleles are sampled. We focused on determining the heterozygous variants that fell into the first three categories, as no clones covered those the fourth category and their phase was non-determinable with the sampled BAC clones. In the previous paragraph, we described how heterozygous variants from category 1) are recovered.

To recover variants from the second and third category, we used CGI WGS data of PGP1 (Peters et al. (2012)). The CGI WGS reads were mapped to hg19/b37 reference genome for variant calling using the CGI proprietary algorithm. For the heterozygous CGI variants that were not recovered using BAC pool data, we needed to verify that at least one clone covered the variant For instance, if a CGI heterozygous variant is called homozygous in using pooled BAC reads, it falls into category 2) and we can phase it. If a CGI heterozygous variant is not called using pooled BAC reads but is covered by at least eight reads from the pooled BAC data, we consider it a heterozygous variant from category 3) and recover it.

3.4.3 Reconstructing BAC clones from sequencing reads

After mapping the sequence reads in a pool to the reference genome, we identified regions of enriched coverage (that is, BAC contigs) by using targetcut in the SAMtools library (Li et al. (2009a)). targetcut identifies regions of enriched coverage by calculating read depth for 1kbp windows and then looking for consecutive regions where two-thirds of the windows have a read depth above the predicted background level (95th percentile of read depths, if reads were distributed uniformly across the genome). The regions are then appropriately trimmed to find the first and final base pair read in each region.

$$G(x) = -\frac{x}{\sqrt{2\pi}}e^{\frac{x^s}{2}} \tag{3.6}$$

To recover the non-overlapping portions of a BAC clones, we looked for significant changes in coverage using a method similar to those for detecting changes in copy number variation (Lee et al. (2011)). This algorithm (see Figure 3.13) begins by obtaining the read count for non-overlapping windows of 100 bp within the boundaries of a BAC contig. It is assumed that the read count of a window with overlapping clones has low variance. Therefore, when there is a significant increase in the read count, this indicates that more than one clone is covering the area. Let $R(x)$ be the resulting window versus read count function, where x is the window number (in genomic position order), then, to detect significant changes in read count, we convolute $R(x)$ with the derivative Gaussian function,

The breakpoints of the BAC contig are indicated by positions where the convoluted function reaches above or below a certain threshold ($|t| = 30$). After breaking up the contigs at the breakpoints, the resulting regions are either non-overlapping potions of a BAC clone or the overlapping portion.

3.4.4 Assembling haplotypes from BAC clones

We used a generalized version of HapCUT (Bansal and Bafna (2008)) to assemble BAC clones into haplotypes. Following the notation of Bansal and Bafna (Bansal and Bafna (2008)), the input to HapCUT can be represented as a matrix, X, where each row represents a BAC clone and each column represents a heterozygous variant. It is assumed that all heterozygous sites are bi-allelic, as there are only two haplotypes, and thus the alleles are arbitrarily relabeled as 0 and 1. An entry in the matrix, $X[i][j]$, is either 0, 1 or $-$ depending on the allelic value of position j in BAC clone i. The goal is to partition the rows (clones) of the matrix into two disjoint sets corresponding to the two haplotypes. If the fragments are error-free, the columns of each set are homozygous. However, sequencing errors, for instance, can produce errors in the fragments, and perfect bi-partitions cannot be achieved. Therefore, the goal is to partition the clones such that error correction is minimized-this is also known as the minimum error correction (MEC) objective.

The generalized version of HapCUT takes as input the BAC clones represented in the matrix form described previously. The algorithm starts by assigning a random haplotype configuration and iteratively improves it by finding positive cuts in the graph representation of the matrix and current haplotype configuration. In the graphical representation, the nodes are the variants and there is an edge between two variants if at least one clone covers both variants. The weight of edge (i, j) is the number of clones that are inconsistent with the current phase of i and j, subtracted by the number of reads that are consistent with the current phase of i and j scaled by some factor. If switching the phase for variants on one side of the cut will improve the MEC score, the haplotype configuration is updated. The algorithm iteratively finds positive cuts and updates the haplotype configuration until the MEC score does not improve.

3.5 Appendix

Table 3.4: **More PGP1 statistics.** More variant statistics for PGP1 compared to others individuals of European descent.

	Total Variants	Hom/Het Ratio	% Novel Variants
HuRef (Levy et al. (2007))	3,213,401	0.82	15%
JDW (Wheeler et al. (2008))	3,322,090	0.79	18%
NA07022 (Drmanac et al. (2010))	3,076,870	0.61	10%
20 Genomes (Pelak et al. (2010))	3,473,639 (avg)	0.593 (average)	13% (avg, compared to dbSNP129)
MP1 (Suk et al. (2011))	3,258,774	0.59	8% (compared to dbSNP129)

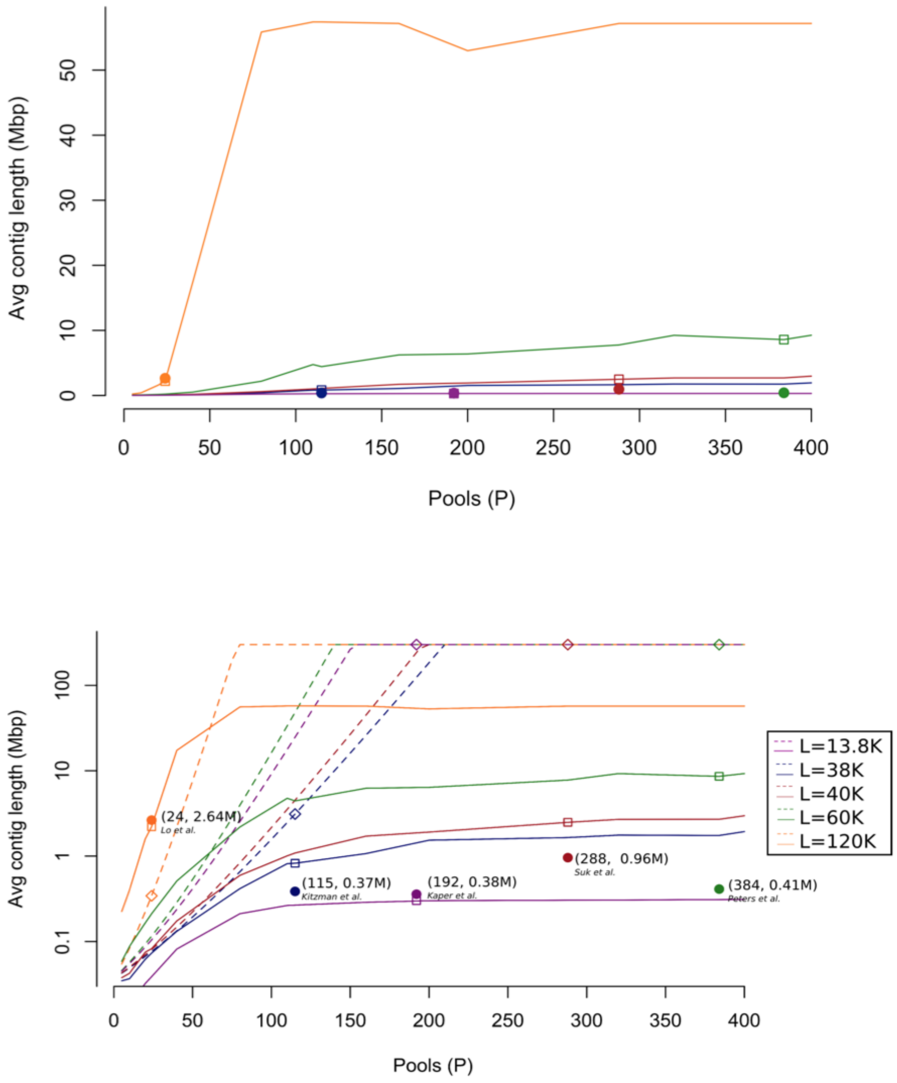


Figure 3.6: **Theoretical contig length of various clone-based haplotyping designs.** (a) Average contig length vs number of pools given read length. This figure shows Figure 3.1 with the log scale removed. (b) The dotted lines show the Lander and Waterman estimates for haplotype length capped at 300 Mbp, the length of chromosome 1. The solid curves show the simulated contig lengths given the actual distribution of heterozygous variants on chromosome 1 obtained from CGI whole genome sequencing data of PGP1. In contrast, the diamonds and squares represent the Lander Waterman estimate and simulated estimate respectively.

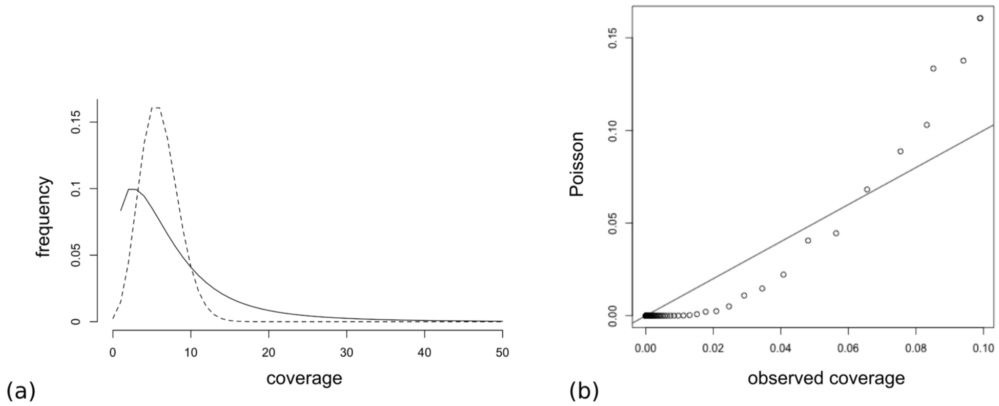


Figure 3.7: **Effect of amplification bias on sequencing coverage.** (a) Actual and ideal distribution of the sequencing coverage. The solid line shows the actual distribution and the dashed line shows the ideal Poisson distribution. Using a nave 4-read coverage rule for variant calling, under the idealized settings (no bias), we expect to see a variant recovery rate of 84%. However, under the observed bias on sequencing coverage, the 4-read coverage rule would yield a variant recovery rate of 61%. Indeed, using the GATK filter protocol for calling variants, the observed f is 65%. (b) Q-Q plot of the actual read depth distribution and the idealized Poisson distribution. Comparing the trend of the points to the y=x line illustrate the difference between the two distributions.

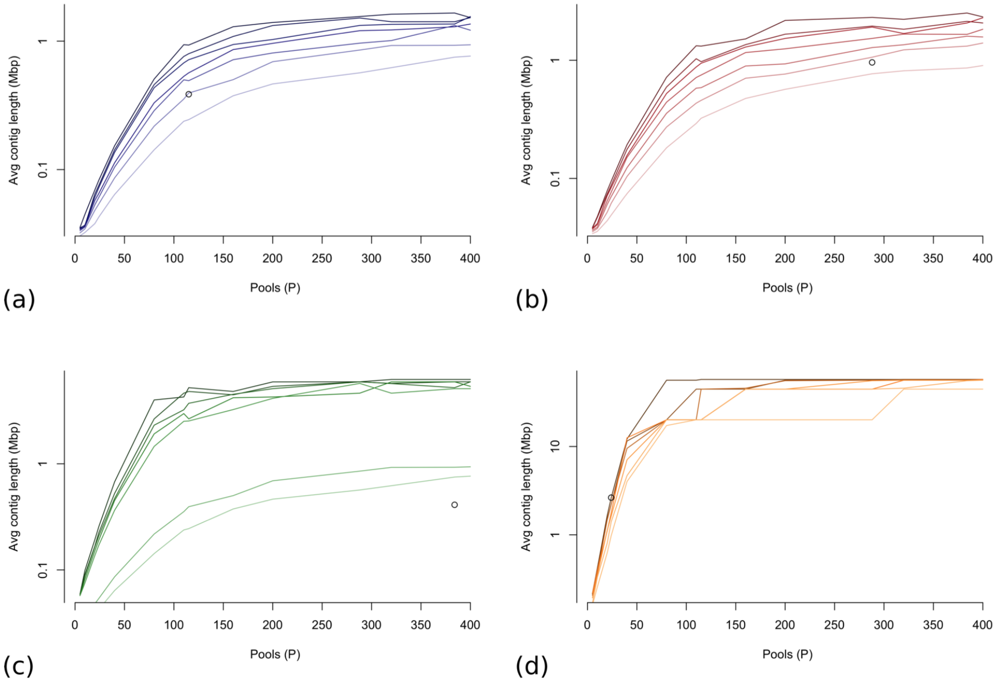


Figure 3.8: **Effect of the fraction of variants recovered on haplotype length.** Simulation results that show how haplotype length is affected by f, the fraction of variants recovered per clone. In each subfigure, L is set to (a) 37 Kb, (b) 40 Kb, (c) 60 Kb, or (d)140 Kb while n is fixed at 5000 and p and f are varied. Each curve in a subfigure represents simulations under a different value of f (1.0, 0.75, 0.65, 0.55, 0.45, 0.35, or 0.25); the darker color indicates higher f value. The circle dots represent the actual reported N50 length for (a) Kitzman et al., (b) Suk et al., (c) Peters et al., and (d) our BAC clone haplotypes.

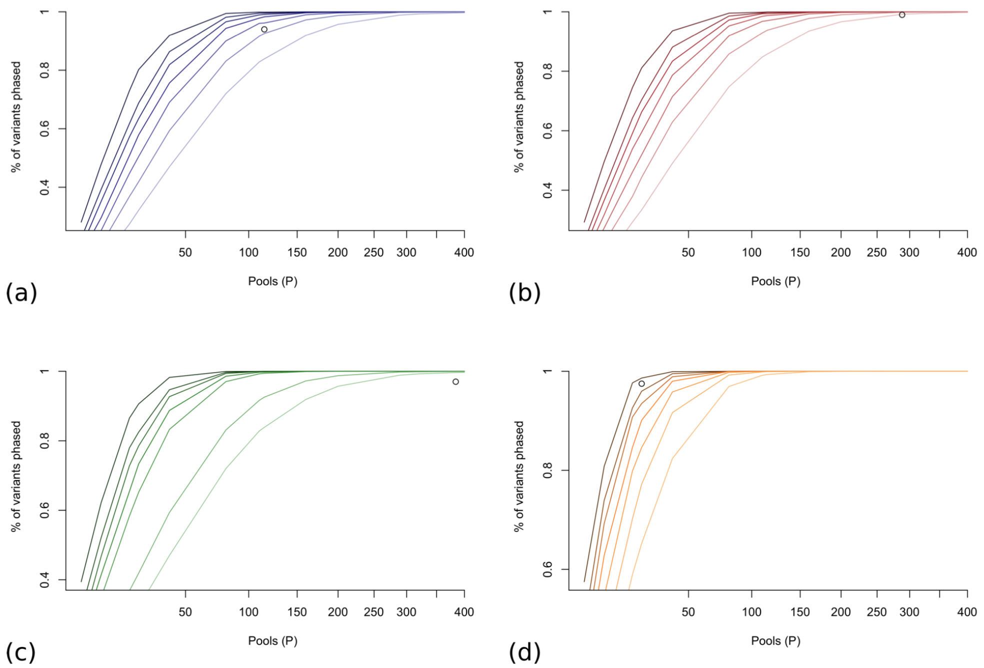


Figure 3.9: **Effect of the fractions of variants recovered on haplotype resolution.** Simulation results that show how haplotype resolution is affected by f, the fraction of variants recovered per clone. In each subfigure, L is set to (a) 37 Kb, (b) 40 Kb, (c) 60 Kb, or (d)140 Kb while n is fixed at 5000 and p and f are varied. Each curve in a subfigure represents simulations under a different value of f (1.0, 0.75, 0.65, 0.55, 0.45, 0.35, or 0.25); the darker color indicates higher f value. The circle dots represent the reported haplotype resolution for (a) Kitzman et al., (b) Suk et al., (c) Peters et al., and (d) our BAC clone haplotypes. The actual haplotype resolution of our BAC haplotypes (97.5%) is slightly higher than the simulated haplotype resolution at f=0.65 (93.5%). The value of f is not reported in the other studies.

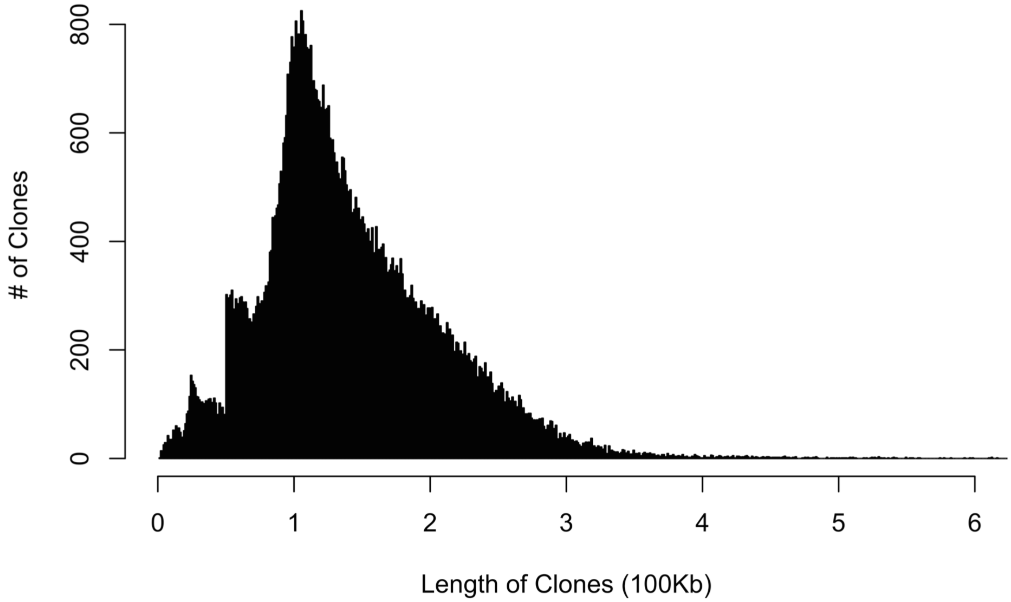


Figure 3.10: **Distribution of reconstructed clone lengths.**

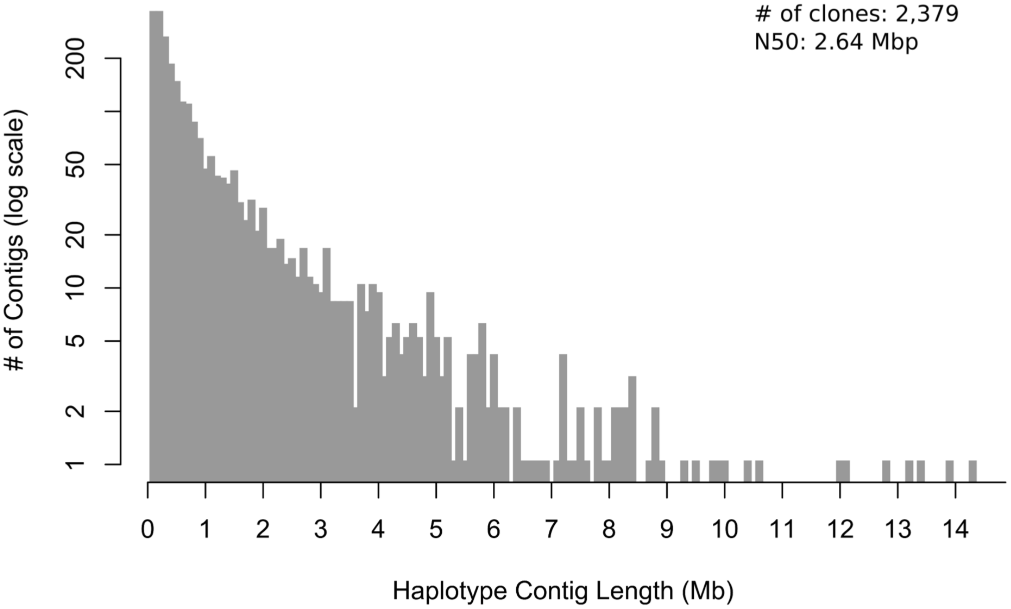


Figure 3.11: **Distribution of the haplotype lengths.**

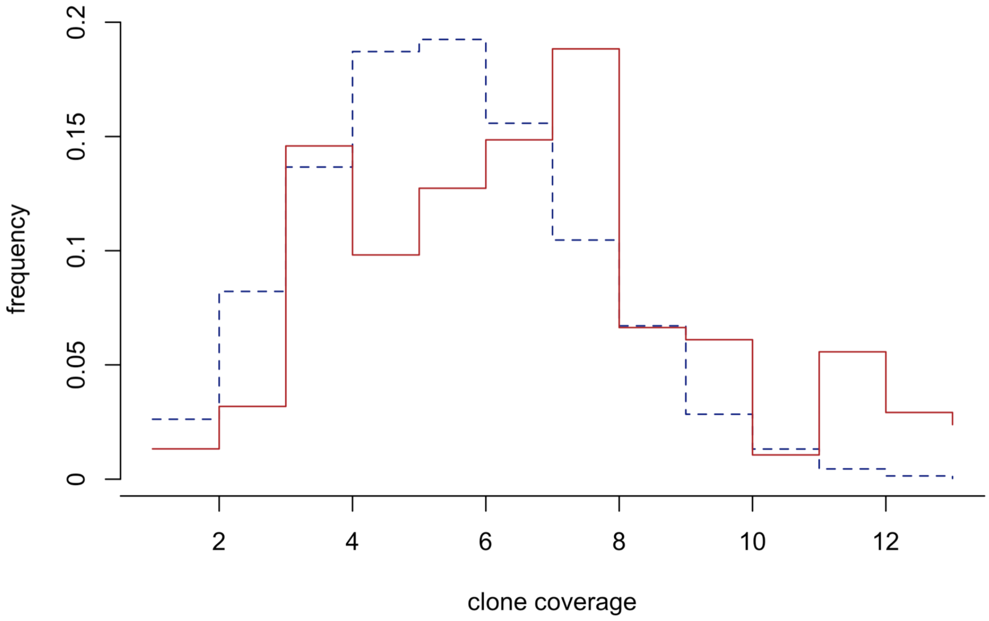
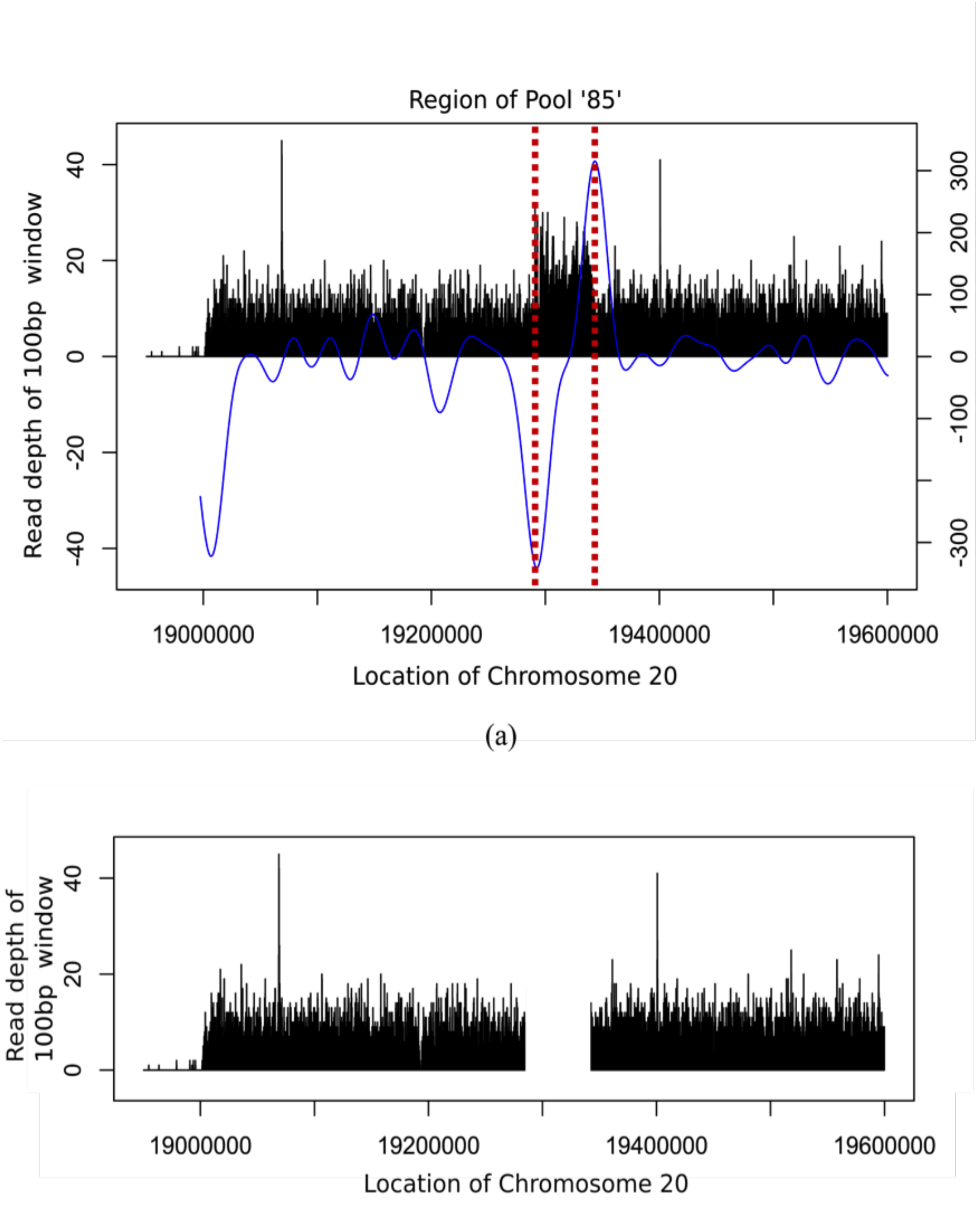


Figure 3.12: **Distribution of clone coverage.** Clone coverage at discrepant locations (red, solid) and match locations (blue, dashed) when comparing BAC haplotypes with LFR haplotypes. The average clone coverage at discrepant locations is 6.2 while the average clone coverage at match locations is 5.0. Furthermore, 95% of the discrepant locations are covered by three or more clones, indicating high confidence in our calls.



(a)

(b)

Figure 3.13: **Illustration of clone reconstruction method.**(a) Read depth of a region on chromosome 20 in pool "85". The read depth is convoluted with the derivative Gaussian function to determine the boundaries (dotted red lines) of the overlapping region. (b) The two reconstructed BAC clones after removing the overlapping region.

Table 3.5: HLA statistics.

Gene	Total num. of clones	Number of haplotype blocks	% of bases covered by clones	Variants detected (Het. and Hom. variants)	Variants phased	% of variants phased
			HLA Class I Genes			
HLA-A	7	1	100%	105	104	99%
HLA-B	0	0	-	-	-	-
HLA-C	0	0	-	-	-	-
HLA-E	2	1	100%	3	3	100%
HLA-F	5	1	100%	15	15	100%
HLA-G	1	1	100%	31	29	94%
HLA-H	4	1	100%	63	63	100%
HLA-J	6	1	100%	3	3	100%
HLA-K	5	1	100%	53	53	100%
HLA-L	5	1	100%	13	13	100%
HLA-P	1	1	100%	30	29	97%
HLA-V	1	1	100%	23	23	100%
			HLA Class II Genes			
HLA-DRA	2	1	100%	53	53	100%
HLA-DRB1	7	1	100%	480	477	99%
HLA-DRB5	0	0	-	-	-	-
HLA-DPA1	5	1	100%	1	1	100%
HLA-DPB1	4	1	100%156	154	99%	
HLA-DQA1	8	1	100%	186	185	99%
HLA-DQB1	7	1	100%	123	121	98%
HLA-DMA	2	1	100%	14	5	36%
HLA-DMB	1	1	100%	28	5	18%
HLA-DOA	4	1	100%	14	14	100%
HLA-DOB	4	1	100%	15	15	100%

Simulation of haplotype lengths In order to simulate haplotype length given clone length (L), number of pools (p), and number of clones per pool (n), the following approximations were made:

- We used the distribution of heterozygous variants of PGP1 determined by CGI whole genome sequencing (Peters et al.). Using the exact distribution of variants is essential in modeling overlapping clones that are useful for phasing PGP1 and ultimately in determining haplotype length.

- Modeling overlapping clones in a pool. Due to high clone coverage within a pool, the probability that a clone overlaps with another clone in the same pool (P_o) is greater than 0. While we use more sophisticated methods to deal with overlaps within a pool, for simulations we use a first order approximation and assume that overlapping clones within a pool are thrown out. Therefore, the effective number of clones in a pool is $N? = N(1 - P_o)$

- We modeled the sequencing read coverage per pool (r) by only recovering a fraction of the variants spanned by a clone (f). We calculated the fraction of variants recovered in each clone for our BAC data and found that $f = 65\%$. We note that this fraction is probably higher in Kitzman et al. and Suk et al. because they have higher read coverage and thus a higher chance of recovering variants, and probably lower in Peters et al. as r is very low. Figure 3.9 shows these designs using different fractions of recovered variants in a clone.

The simulator for haplotype length and resolution is available upon request.

The actual reported N50 is close to the simulated N50. But we note that discrepancies may arise due to the assumptions we made. For example, if the distribution of variant distributions is more/less sparse than PGP1?s, the simulated haplotype lengths will differ from the actual haplotype lengths. Different methods for dealing with overlapping clones and different read coverage (r) may also cause discrepancy between simulated and actual haplotype lengths. Natural noise in the data will also cause discrepancy between simulated and actual haplotype lengths.

Derivation: Equation 3.2 The probability that a clone does not overlap with a given clone is $1 - \frac{2L}{G}$. Thus, the probability that a clone overlaps with a given clone in the same pool is given by

$$P_o = 1 - (1 - \frac{2L}{G})^{NP} = 1 - (1 - \frac{2L}{G})^{c_p \frac{2G}{2L}} \approx 1 - e^{-2c_p} \tag{3.7}$$

Derivation: Equation 3.3 Given the length of the diploid genome is G, the total length of the haploid genome is given by $2G$. Let x be the probability that a particular position on the haploid genome is not covered by any clone.

$$x = (1 - \frac{L}{2G})^{nP} = (1 - \frac{L}{2G})^{c\frac{2G}{2L}} \approx e^{\frac{-c}{2}} \tag{3.8}$$

In order to recover a heterozygous variant, both copies of the variant must be covered by at least one clone each. Therefore, the probability that a heterozygous variant is given by

$$p_v = 1 - 2x = 1 - 2e^(\frac{-c}{2}) \tag{3.9}$$

3.6 Acknowledgements

Chapter 3, in part, is a reprint of the material as it appears in Genome Biology 2013. Lo, Christine; Liu, Rui; Lee, Jehyuk; Robasky, Kimberly; Byrne, Susan; Lucchesi, Carolina; Aach, John; Church, George; Bafna, Vineet; Zhang, Kun.“On the design of clone-based haplotyping.” The dissertation author was the primary investigator and author of this paper.

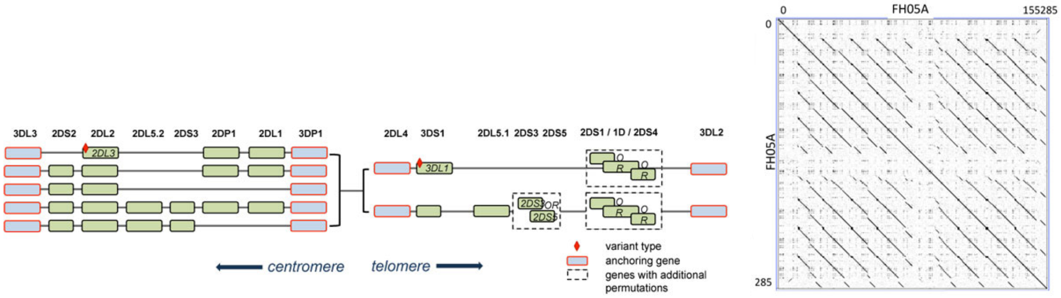
Chapter 4

Template identification for complex regions.

4.1 Introduction

The inexorable drop in costs and rise in throughput of DNA sequencing is driving a future in which every individual person will have their genome sequenced, perhaps multiple times in their lifetimes (Hall (2011)). Current high throughput technologies produce sequenced read fragments from donor genomes, which are then used for inferring the complete genomic sequence. The main algorithmic approaches for inferring a donor genome from a set of its sequenced reads are either based on *de novo assembly* (Li et al. (2010); Pevzner et al. (2001)), i.e. producing a parsimonious super-string that approximately contains most reads as its substrings, or based on *mapping* approaches (Havlak et al. (2004); Kidd et al. (2008); Mills et al. (2011)), in which the algorithm takes the read set and a previously sequenced reference genome (or a set of reference genomes), maps the reads to the reference, and uses the identified similarities and variations in order to predict the donor genome.

While the accuracies of sequencing technologies keep improving and their usage costs keep decreasing, many of them still produce reads of relatively short lengths. Reconstruction of repetitive genomic regions using the mentioned approaches is considered more challenging, due to the fact that short reads may be de-novo assembled, or mapped to the reference, in multiple ambiguous manners. The difficulty even increases for diploid genomes, limiting the investigation of many important genomic regions, such as the killer cell immunoglobulin like receptor (KIR) region (located in humans within the 1Mb Leucocyte Receptor Complex 19q13.4, see Fig. 4.1b), the 3.6Mbp Human Leucocyte Antigen (HLA) region and others, which exhibit highly repetitive sequences and extensive polymorphisms.



(a)

(b)

Figure 4.1: **KIR Region.** (a) Variability of gene architecture in KIR haplotypes (Hsu et al. (2002)). (b) Complex repeat structure in a KIR haplotype, as observed by a dot-plot of FH05A against FH05A. The different genes all show significant sequence similarity. Dot-plot prepared using Gepard (Krumsiek et al. (2007)).

Here, we address the problem of *assessing the quality* of a donor genome prediction given the set of its sequenced reads, confronting difficulties related to genomic regions of repetitive nature. We present in Section 4.3 a prediction quality measure which is independent of the approach used for generating the prediction. It combines scoring penalties related to both (a) imperfect alignments of the reads to the predicted region, and (b) deviations between the expected and actual read coverage of segments of the region. Our tool differs from previous ones which compare predictions to a known reference. For example, tools that evaluate the quality of de-novo assemblies (Salzberg et al. (2012)) rely on comparing assembled genomes to known references. Mapping tools (Langmead and Salzberg (2012); Li and Durbin (2010)) can be used to provide a naive scoring function comparable to SAGE by summing up the best alignment score of each read. This naive scoring function only optimizes the alignment of the reads and does not take into account read coverage. In Section 4.4 we show the advantage of simultaneously optimizing the combined alignment and coverage score by comparing our tool to the naive approach.

In order to evaluate the new cost function, we applied it in Section 4.4 to the KIR region, a hyper-variable region known to be important for the immediate immune response in humans and higher mammals (Hsu et al. (2002)). The KIR region is challenging to reconstruct from sequence read fragments due to its variable gene architecture (Figure 4.1a) and repetitive nature (Figure 4.1b). We show that our scoring function allows us to correctly identify KIR haplotype templates in diploid genomes, differentiating correct predictions form incorrect ones based on their computed score, while the naive approach fails in many cases to predict the correct template.

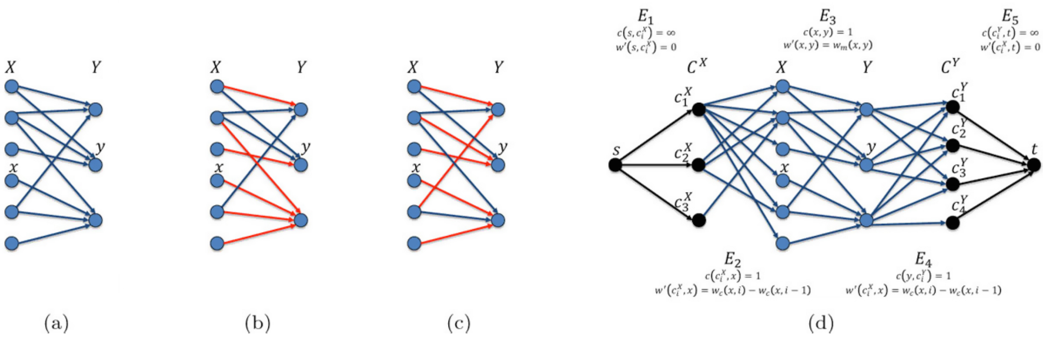
Our cost function for evaluating donor genome predictions is based on a new variant of a bipartite matching problem, entitled *Coverage Sensitive many-to-many min-cost bipartite Matching* (CSM), which is a many-to-many generalization of the classical min-cost (or max-weight) bipartite matching problem (Edmonds and Karp (1972); Lovász and Plummer (1986)). The formal definition of the CSM problem is given in Section 4.2. While in general CSM is NP-Hard (see Appendix), we show a special "convex" case for which CSM can be efficiently solved by reducing it to a network flow problem, similar to many other variants of bipartite matching problems (Lovász and Plummer (1986)). Optimal matching/flow algorithms were recently used by several related works to predict structural variations between genomes. Examples to such works include (Medvedev et al. (2010)), in which min-cost flow was used to call copy number variations between a reference and a donor genome, (Hajirasouliha et al. (2010)), which used maximum-weight matching in order to reconstruct breakpoint sequences in long genomic insertions, and (Hormozdiari et al. (2011)), which used maximum-flow in order to apply a post-process refinement of simultaneous detection of structural variations in multiple genomes.

4.2 Coverage Sensitive many-to-many min-cost bipartite Matching (CSM)

The CSM problem is a many-to-many generalization of the classical min-cost bipartite matching problem (Lovász and Plummer (1986)). We describe the problem in an abstract setting, and cast it to a read alignment problem in Section 4.3.

Consider arbitrary sets X and Y. A *many-to-many matching* (henceforth a *matching*) between X and Y is a set M of pairs $\{(x, y) \in X \times Y\}$ (see Figure 4.2, (a), (b), and (c)). The *coverage* of an element $x \in X$ with respect to a matching M is $c_M(x) = |\{y : (x, y) \in M\}|$. Symmetrically, $c_M(y) = |\{x : (x, y) \in M\}|$ for $y \in Y$.

A *coverage sensitive matching cost function* (henceforth a *cost function*) w for X and Y assigns *matching costs* $w_m(x, y)$ for every pair $(x, y) \in X \times Y$, and *coverage costs* $w_c(z, i)$ for every $z \in X \cup Y$ and every integer $i \geq 0$. The *cost* of a



(a) (b) (c) (d)

Figure 4.2: **Matching instance and its reduction to a cost flow network.** (a) A bipartite graph corresponding to sets X and Y. In our particular application, X represents a set of reads and Y represents a set of genomic segments, where the expected coverage of each read is one and segments are expected to be uniformly covered. Each read $x \in X$ potentially maps to multiple segments, illustrated by the edges in the graph. An edge (x, y) has the weight $w_m(x, y)$, reflecting the best similarity between read x and a substring of of the genome starting at segment y. (b) and (c) depict two possible *matchings*. In (b), one of the y segments is covered by four reads, while the other two segments are covered by one read each. In (c), each segment is covered by two reads. It is possible that the matching in (b) is better in terms of sequence similarity, though is unrealistic in terms of segment coverage, which would make the matching in (c) preferable. (d) The corresponding network. Each pair of consecutive layers is a bipartite graph with capacities c and costs w' as described.

matching M between X and Y with respect to w is given by

$$\sum_{(x,y)\in M} w_m(x,y) + \sum_{z\in X\cup Y} w_c(z, c_M(z)) \tag{4.1}$$

Note that CSM is a generalization of classical problems in combinatorics. For example, consider the problem of finding a maximum (partial one-to-one) matching on a bipartite graph G with vertex shores X, Y, and an edge set E. This problem can be solved by solving CSM on the input X, Y using the following costs: set $w_c(z,0) = w_c(z,1) = 0$, and $w_c(z,i) = \infty$ for all $z \in X \cup Y, i > 1$; set $w_m(x,y) = -1$ for $(x,y) \in E$ and otherwise set $w_m(x,y) = \infty$. Similarly, CSM can also be used for solving the minimum/maximum weight variants of the bipartite matching problem. However, CSM is NP-hard in general (see Section 4.6), and therefore we do not expect to solve the general instance efficiently.

4.2.1 CSM with convex coverage costs

Let (X, Y, w) be a matching instance. We say that w has *convex* coverage costs if for every element $z \in X \cup Y$ and every integer $i > 0$, $w_c(z,i) \leq \frac{w_c(z,i-1)+w_c(z,i+1)}{2}$. We show here that *CSM* with convex coverage costs can be reduced to the poly-time solvable *min-cost integer flow* problem (Edmonds and Karp (1972).

For $x \in X$, denote $d_x = |\{y : w_m(x,y) < \infty\}|$, and similarly $d_y = |\{x : w_m(x,y) < \infty\}|$ for $y \in Y$. Denote $d_X = \max\limits_{x\in X} d_x$ and $d_Y = \max\limits_{y\in Y} d_y$. The reduction builds the flow network $N = (G, s, t, c, w')$, where G is the network graph, s and t are the source and sink nodes respectively, and c and w' are the edge capacity and cost functions respectively. The graph $G = (V, E)$ is defined as follows (Figure 4.2d).

- $V = X \cup Y \cup C^X \cup C^Y \cup \{s, t\}$, where the sets $C^X = \{c_1^X, c_2^X, \ldots, c_{d_X}^X\}$, $C^Y =$

$\{c_1^Y, c_2^Y, \ldots, c_{d_Y}^Y\}$, and $\{s, t\}$ contain unique nodes different from all nodes in X and Y. Note that we use the same notations for elements in X and Y and their corresponding nodes in V, where ambiguity can be resolved by the context.

- $E = E_1 \cup E_2 \cup E_3 \cup E_4 \cup E_5$, where
  - $E_1 = \{(s, c_i^X) : c_i^X \in C^X\}$,
  - $E_2 = \{(c_i^X, x) : c_i^X \in C^X, x \in X, d_x \leq i\}$,
  - $E_3 = \{(x, y) : x \in X, y \in Y, w_m(x, y) < \infty\}$,
  - $E_4 = \{(y, c_i^Y) : y \in Y, c_i^Y \in C^Y, d_y \leq i\}$, and
  - $E_5 = \{(c_i^Y, t) : c_i^Y \in C^Y\}$.

The capacity function c assigns infinity capacities to all edges in E_1 and E_5 and unit capacities to all edges in E_2, E_3 and E_4. The cost function w' assigns zero costs to edges in E_1 and E_5, costs $w_c(x, i) - w_c(x, i-1)$ to edges $(c_i^X, x) \in E_2$, costs $w_c(y, i) - w_c(y, i-1)$ to edges $(y, c_i^Y) \in E_4$, and costs $w_m(x, y)$ to edges $(x, y) \in E_3$. For $E' \subseteq E$, denote $w'(E') = \sum\limits_{e \in E'} w'(e)$. An *integer flow* in N is a function $f : Earrow\{0, 1, 2, \ldots\}$, satisfying that $f(e) \leq c(e)$ for every $e \in E$ (*capacity constraints*), and $\sum\limits_{u:(u,v) \in E} f(u, v) = \sum\limits_{u:(v,u) \in E} f(v, u)$ for every $v \in V \setminus \{s, t\}$ (*flow conservation constraints*). The cost of a flow f in N is defined by $w'(f) = \sum\limits_{e \in E} f(e) w'(e)$.

In what follows, let (X, Y, w) be a matching instance where w has convex coverage costs, and let N be its corresponding network. Due to the convexity requirement, for every $x \in X$ and every integer $i > 0$, $w'(c_{i+1}^X, x) - w'(c_i^X, x) = (w_c(x, i+1) - w_c(x, i)) - (w_c(x, i) - w_c(x, i-1)) = w_c(x, i+1) + w_c(x, i-1) - 2w_c(x, i) \geq 0$. Similarly, for every $y \in Y$ and every integer $i > 0$, $w'(y, c_{i+1}^Y) - w'(y, c_i^Y) \geq 0$, and we get the following observation:

Observation 1. *Series of the form* $w'(c_1^X, x), w'(c_2^X, x), \ldots$ *and* $w'(y, c_1^Y), w'(y, c_2^Y), \ldots$ *are non-decreasing. Consequentially, for every* $E' \subseteq \{(c_i^X, x) : x \in X, 1 \leq i \leq d_x\}$ *and* $E'' = \{(c_i^X, x) : x \in X, 1 \leq i \leq |E'|\}$, $w'(E'') \leq w'(E')$, *and similarly for* $E' \subseteq \{(y, c_i^Y) : y \in Y, 1 \leq i \leq d_y\}$ *and* $E'' = \{(y, c_i^Y) : y \in Y, 1 \leq i \leq |E'|\}$.

Given a flow f in N, define the matching $M_f = \{(x, y) : (x, y) \in E_3, f(x, y) = 1\}$. Denote $E_x^f = \{(c_i^X, x) : f(c_i^X, x) = 1\}$ and $E_y^f = \{(y, c_i^Y) : f(y, c_i^Y) = 1\}$. Since for edges $e \in E_1 \cup E_5$ we have that $w'(e) = 0$, and since for edges $e \in E_2 \cup E_3 \cup E_4$ we have that $f(e) \in \{0, 1\}$ (due to capacity constraints), we can write

$$\begin{aligned} w'(f) = \sum_{e \in E} f(e) w'(e) = \sum_{\substack{e \in E_2 \cup E_3 \cup E_4 \\ f(e)=1}} w'(e) \\ = w'(M_f) + \sum_{x \in X} w'(E_x^f) + \sum_{y \in Y} w'(E_y^f). \end{aligned} \tag{4.2}$$

Given a non-infinity cost matching M between X and Y, define the flow f_M in N as follows:

- For every $(x, y) \in E_3$, $f(x, y) = 1$ if $(x, y) \in M$, and otherwise $f(x, y) = 0$;
- For every $(c_i^X, x) \in E_2$, $f(c_i^X, x) = 1$ if $c_M(x) \leq i$, and otherwise $f(c_i^X, x) = 0$;
- For every $(y, c_i^Y) \in E_4$, $f(y, c_i^Y) = 1$ if $c_M(y) \leq i$, and otherwise $f(y, c_i^Y) = 0$;
- For every $(s, c_i^X) \in E_1$, $f(s, c_i^X) = |\{x : f(c_i^X, x) = 1\}|$;
- For every $(c_i^Y, t) \in E_5$, $f(c_i^Y, t) = |\{y : f(y, c_i^Y) = 1\}|$.

It is simple to assert that f_M is a valid flow in N (satisfying all capacity and flow conservation constraints), and that $M_{f_M} = M$.

Claim 1. *For every flow f in N, $w'(f_{M_f}) \leq w'(f)$.*

Proof. From flow conservation constraints $|E_x^f| = |E_x^{f_{M_f}}| = c_{M_f}(x)$ for every $x \in X$, where in particular by definition we have that $E_x^{f_{M_f}} = \{(c_i^X, x) : 1 \leq i \leq c_{M_f}(x)\}$. Therefore, it follows from Observation 1 that $w'(E_x^{f_{M_f}}) \leq w'(E_x^f)$ for every $x \in X$, and similarly it may be shown that $w'(E_y^{f_{M_f}}) \leq w'(E_y^f)$ for every $y \in Y$. Hence,

$$\begin{aligned} w'(f_{M_f}) &\overset{\text{Eq.4.2}}{=} w'(M_{f_{M_f}}) + \sum_{x \in X} w'(E_x^{f_{M_f}}) \\ &+ \sum_{y \in Y} w'(E_y^{f_{M_f}}) \\ &\leq w'(M_f) + \sum_{x \in X} w'(E_x^f) + \sum_{y \in Y} w'(E_y^f) \\ &\overset{\text{Eq.4.2}}{=} w'(f). \end{aligned}$$

□

Denote $\Delta = \Delta(X, Y, w) = \sum_{z \in X \cup Y} w_c(z, 0)$, and note that Δ depends only on the instance (X, Y, w) and not on any specific matching.

Claim 2. *For every matching M between X and Y, $w'(f_M) = w(M) - \Delta$.*

Proof. For $x \in X$, we have that $w'(E_x^{f_M}) = w'(c_1^X, x) + w'(c_2^X, x) + \ldots + w'(c_{c_M(x)}^X, x) = (w_c(x,1) - w_c(x,0)) + (w_c(x,2) - w_c(x,1)) + \ldots + (w_c(x, c_M(x)) - w_c(x, c_M(x) - 1)) = w_c(x, c_M(x)) - w_c(x, 0)$, and similarly $w'(E_y^{f_M}) = w_c(y, c_M(y)) - w_c(y, 0)$ for $y \in Y$. Therefore,

$$\begin{aligned}
w'(f_M) &\overset{\text{Eq.4.2}}{=} w'(M) + \sum_{x\in X} w'(E_x^{f_M}) + \sum_{y\in Y} w'(E_y^{f_M}) \\
&= w'(M) + \sum_{x\in X} \left(w_c\left(x, c_M\left(x\right)\right) - w_c\left(x,0\right)\right) \\
&\quad + \sum_{y\in Y} \left(w_c\left(y, c_M\left(y\right)\right) - w_c\left(y,0\right)\right) \\
&= \sum_{(x,y)\in M} w_m\left(x,y\right) + \sum_{z\in X\cup Y} w_c\left(z, c_M\left(z\right)\right) \\
&\quad - \sum_{z\in X\cup Y} w_c\left(z,0\right) \\
&\overset{\text{Eq.4.1}}{=} w(M) - \Delta
\end{aligned}$$

□

Claim 3. *Let f^* be a minimum cost flow in N. Then, M_{f^*} is a minimum cost matching between X and Y, and $CSM(X,Y,w) = w'(f^*) + \Delta$.*

Proof. Since f^* is a minimum cost flow in N, $w'(f^*) \leq w'(f_{M_{f^*}}) \overset{\text{Clm.1}}{\leq} w'(f^*)$, thus $w'(f^*) = w'(f_{M_{f^*}})$. Let M be a matching between X and Y. Again, from the optimality of f^*, $w'(f^*) \leq w'(f_M)$ and so $w(M_{f^*}) - \Delta \overset{\text{Clm.2}}{=} w'(f_{M_{f^*}}) = w'(f^*) \leq w'(f_M) \overset{\text{Clm.2}}{=} w(M) - \Delta$, and in particular $w(M_{f^*}) \leq w(M)$. Thus, M_{f^*} is a minimum cost matching for (X,Y,w), and so $CSM(X,Y,w) = w(M_{f^*}) \overset{\text{Clm.2}}{=} w'(f^*) + \Delta$. □

4.3 Constructing CSM instance from read mapping data

Consider a set of reads and a prediction of the genomic sequence (henceforth, the "prediction") from which the reads were extracted. It is assumed that the sequencing procedure produces reads with some sequencing error probability, and that read extraction positions along the genome adhere to some expected distribution. The probability for extracting a read starting at a given position may depend on the sequential context at this position and its location along the genome. Given such probabilities, it is possible to compute for a given segment of the prediction an expected amount of extracted reads starting within this segment. Such an amount of expected reads will be referred to here as the *expected coverage* of the segment. Hence, we can argue that the reads *well support* the prediction in case it is possible to assign to each read a position within the prediction, from which it was presumably extracted, in a manner that (a) each read sequence approximately matches the substring of the prediction starting at the assigned position, and (b) for every segment of the prediction, the amount of reads assigned to positions within this segment does not deviate significantly from the expected coverage of the segment. On the other hand, when no such position assignment can be found, it is suggestive that the prediction exhibits some variation with respect to the true genome.

Given a predicted region, a *mapping* between the reads and the prediction is a function that assigns to each read a set of positions in the region from which it is possible to extract the read (with some allowed amount of sequencing errors). Software tools for producing such mappings exist (e.g. Bowtie (Langmead and Salzberg (2012)) and are widely used. Ideally, if the prediction is in fact the correct genomic sequence from which the reads were extracted, and this region is non-repetitive, it is expected that a mapping would assign to each read a unique position that is the true position from which it was extracted. Nevertheless, when the sequence contains repeats, and sequencing errors are not negligible, it is expected that some of the reads will be mapped to multiple positions (due to the repeats), while others may not be mapped to any position (due to sequencing errors). Given a mapping between the reads and the region, we define a *read-to-genome matching* as a function that selects for each read at most one corresponding position among its set of positions given by the mapping, from which it was presumably extracted. A read-to-genome matching better supports the prediction the more reads it match to the genome, the higher the similarity is between the reads and the chosen matching position, and the smaller the deviation is between the expected coverage and the coverage implied by the matching positions.

The quality of a read-to-genome matching can be naturally evaluated using the CSM formalism described in the previous section. A matching instance (X, Y, w) can be generated, choosing X to be the set of reads, and Y to be a partition of the prediction into segments (where each element in Y corresponds to a segment in the partition). For each read $x \in X$ and each segment $y \in Y$, $w_m(x, y)$ is set to the best sequence similarity score between x and a substring of the prediction starting at y (such similarity scores may be generated by tools such as Bowtie (Langmead and Salzberg (2012)), or set to ∞ if no substring starting at y is similar to x. The coverage cost function for a read $x \in X$ sets $w_c(x, 0)$ to some penalty added to the score in case x is unmatched, sets $w_c(x, 1)$ to 0 (no penalty is added when x participates in the matching), and $w_c(x, i)$ for $i > 1$ to ∞ (a matching in which a read is assigned to more than one position is illegal, and has an infinite cost). For a segment $y \in Y$, it is possible to compute the expected coverage c_y of y, and generate a convex score function $f(i)$ whose minimum point is at $i = c_y$, and set $w_c(y, i) = f(i)$ for every nonnegative integer i. The cost of an optimal matching for this instance can then serve as a quality measure for the prediction.

Implementation: We implemented the CSM algorithm as a java based tool named SAGE, a **S**coring function for **A**ssembled **GE**nomes, freely available upon publication. The inputs to SAGE are a set of reads, R, mapped to a genomic template, G, in the BAM format (samtools.sourceforge.net) along with a parameter file containing alignment costs, unmatched read penalty, genome segmentation, expected segment coverage values, and a choice of coverage cost functions (currently linear and polynomial cost functions).

4.4 Results

We tested SAGE on the hypervariable KIR region. The KIR region, while variable, is tightly organized and contains between 8 and 14 genes, and 2 pseudogenes (Figure 4.1a) (Middleton and Gonzelez (2010). The genes are organized into two adjacent regions, each bordered by two anchoring genes/pseudo-genes: KIR3DL3 and 3DP1 for the centromeric region; 2DL4 and 3DL2 for the telomeric region. Variability within KIR is expressed in the form of changing gene numbers, gene-copy numbers, and gene polymorphisms. There are two broad types of KIR haplotypes- Type A and Type B- that are distinguished by their gene content. Type A haplotypes are characterized by the absence of the following genes: {KIR-2DL5, -2DS1, -2DS2, -2DS3, -2DS5, -3DS1}, while Type B haplotypes contain one or more of these genes (Marsh et al. (2003). Type B haplotypes can be split further into different sub-types, characterized by the gene content on the centromeric-side and telomeric-side. The various (sub-)types of KIR haplotypes are denoted by {A, AB, BA1, BA2, BA2X, Bdel, B}. However, the typing is incompletely developed, and is likely to change as more data is acquired.

To test the effectiveness of SAGE on a variety of haplotype types, we simulated reads from 27 known KIR haplotypes using GemSIM (McElroy et al. (2012b) with an error model learned from paired-end (100 × 2)bp reads generated by Illumina GA IIx with TrueSeq SBS Kit v5-GA(McElroy et al. (2012b). The 27 haplotype templates were taken from the IPD-KIR database (Robinson et al. (2005). The sequences of these templates were obtained experimentally by first separating the two haplotypes of an individual using fosmid-pools, determining the gene content and architecture of each haplotype using STS assays, and then finally sequencing the individual fosmids (Pyo et al. (2010).

Before we ran SAGE, we mapped each read set, R, back to each template, G, using Bowtie. We ran Bowtie under the "-a" option with all other parameters set to the default, in order to obtain a set of all possible mapping locations and their corresponding alignment costs for each read, which was used as input into SAGE. The mapping position of a paired-end read was set to be the genomic index to which the first character of the first sub-read was aligned. The alignment cost for a complete (100 × 2)bp paired-end read varied between 0 and 180, with 0 corresponding to identity. When two paired-ends mapped in a concordant manner, the total alignment cost for the read was calculated by adding the alignment cost of both paired-ends. When a paired-end did not have a concordant mate, suggestive of incorrect architecture, the alignment cost was further penalized by adding a cost of 90, which is the maximum penalty for one paired-end.

The unmatched read penalty was constant for all reads and set to 100, allowing for a progressive reduced penalty for matching, scaled by alignment costs.

On the other side, the genome G was partitioned to segments of fixed length of 1000bp (except for the last segment which may be shorter than 1000bp), with expected coverage per segment given by $\lambda = 1000\frac{|R|}{|G|}$ (with the appropriate adjustment for the last segment), where $|R|$ and $|G|$ denote the number of reads and the length of the genome, respectively. While coverage varies due to natural biases in sampling, there are technical challenges in maintaining convexity and computational efficiency for sophisticated coverage cost functions. Therefore, segment coverage cost function was chosen to be the quadratic function $f(i) = (\lambda - i)^2$.

To the best of our knowledge, SAGE is the first tool that scores templates given a set of reads. As there is no competing tool, we compared SAGE results against a naive approach that ignores coverage and sums up the best alignment score for each read to obtain a total score for each read set and template. The scores obtained by this approach will be referred to as the *Bowtie scores* below.

Haploid templates

As a first pass, we tested SAGE's ability to score haploid templates. We scored each of the 27 read sets against each of the 27 templates using SAGE . A visualization of the scores are shown in Figure 4.3a, where the templates are organized by sequence similarity so that templates of the same type/sub-type are clustered together. Note that the matrix is not symmetric. Each row corresponds to the scores of a single read data set against a collection of haploid templates. As can be seen, SAGE always gets the top-score for the correct template. Moreover, the other templates from the same sub-type get progressively weaker scores. Major haplotypes fall within distinct blocks, but the scores also suggest a hierarchy within the subtypes that can be studied further.

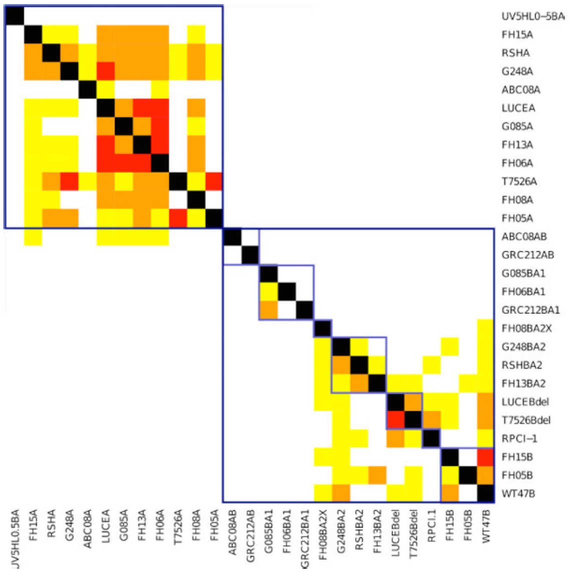
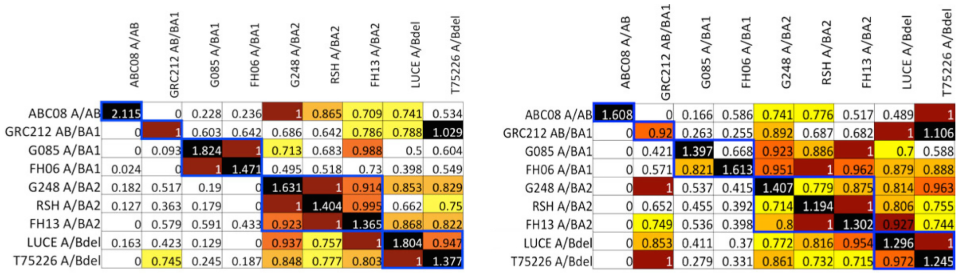


Figure 4.3: **Scoring simulated reads against haploid templates using SAGE .** Each row contains the color-coded percentage from the top-score of a read-set mapped against 27 genomic templates. Black: top-score; Red: within 5% of top-score; Orange: $\leq$ 10% ; Yellow: $\leq$ 20%; White: $>$ 20% below top-score. Sequences are ordered along the rows and columns so that sequences with the same (sub-)type are adjacent to each other. Templates of the same type are indicated by the blue boxes, and those of the same sub-type by light blue boxes.

Diploid templates

To test scoring on more realistic templates, we simulated reads from 9 diploid individuals whose pair of haploid templates were obtained experimentally in Pyo et al. (2010) and are in the IPD-KIR database (Robinson et al. (2005)). The 9 diploid templates from this study fell into one of 6 combination of sub-types. We scored each of the 9 simulated read sets against each of the 9 diploid templates using SAGE. In all but one case, SAGE (Figure 4.4a) and Bowtie (Figure 4.4b) predicted the correct diploid template of the donor.



	ABC08 A/AB	GRC212 AB/BA1	G085 A/BA1	FH06 A/BA1	G248 A/BA2	RSH A/BA2	FH13 A/BA2	LUCE A/Bdel	T75226 A/Bdel
ABC08 A/AB	2.115	0	0.228	0.236	1	0.865	0.709	0.741	0.534
GRC212 AB/BA1	0	1	0.603	0.642	0.686	0.642	0.786	0.788	1.029
G085 A/BA1	0	0.093	1.824	1	0.713	0.683	0.988	0.5	0.604
FH06 A/BA1	0.024	0	1	1.471	0.495	0.518	0.73	0.398	0.549
G248 A/BA2	0.182	0.517	0.19	0	1.631	1	0.914	0.853	0.829
RSH A/BA2	0.127	0.363	0.179	0	1	1.404	0.995	0.662	0.75
FH13 A/BA2	0	0.579	0.591	0.433	0.923	1	1.365	0.868	0.822
LUCE A/Bdel	0.163	0.423	0.129	0	0.937	0.757	1	1.804	0.947
T75226 A/Bdel	0	0.745	0.245	0.187	0.848	0.777	0.803	1	1.377

(a)

	ABC08 A/AB	GRC212 AB/BA1	G085 A/BA1	FH06 A/BA1	G248 A/BA2	RSH A/BA2	FH13 A/BA2	LUCE A/Bdel	T75226 A/Bdel
ABC08 A/AB	1.608	0	0.166	0.586	0.741	0.776	0.517	0.489	1
GRC212 AB/BA1	0	0.92	0.263	0.255	0.892	0.687	0.682	1	1.106
G085 A/BA1	0	0.421	1.397	0.668	0.923	0.886	1	0.7	0.588
FH06 A/BA1	0	0.571	0.821	1.613	0.951	1	0.962	0.879	0.888
G248 A/BA2	0	1	0.537	0.415	1.407	0.779	0.875	0.814	0.963
RSH A/BA2	0	0.652	0.455	0.392	0.714	1.194	1	0.806	0.755
FH13 A/BA2	0	0.749	0.536	0.398	0.8	1	1.302	0.927	0.744
LUCE A/Bdel	0	0.853	0.411	0.37	0.772	0.816	0.954	1.296	1
T75226 A/Bdel	0	1	0.279	0.331	0.861	0.732	0.715	0.972	1.245

(b)

Figure 4.4: **Scoring simulated reads against diploid templates.** Each row of a matrix represent scores from the same read sets mapped to different prediction templates. The scores are normalized so that the second best score in each row is equal to 1 and the worst score is equal to 0. Furthermore, the entries are color-coded accordingly- Black: top-score; Red: second top-score; Orange: within 10% of second top-score; Light Orange: $\leq 20\%$; Yellow: $\leq 30\%$; White: $> 40\%$ below top-score. Both matrices are ordered according to template sub-types. Templates of the same type are indicated by the blue boxes. (a) SAGE scores (b) Bowtie scores

Furthermore, SAGE is better at predicting the sub-type of the donor template than Bowtie. When the donor template is not in database, as is usually the case in practice, SAGE will give a better score to templates that are more similar to the donor while Bowtie may not. For example, row 3 of Figure 4.4 show the scores when the donor template is of type A and BA1. Both SAGE and Bowtie correctly gave the best score to the diploid template G085-A/BA1. However, the template with the next best SAGE score was also of sub-type A/BA1, while the template with the next best Bowtie score was of sub-type A/BA2.

In general, coverage plays an important role in determining the correct haplotype. Figure 4.5b-e show the coverage plots when reads from donor template G085-A/BA1 are mapped to a template of the same sub-type (F06-A/BA1) and a template of a different sub-type (FH13-A/BA2) using SAGE and Bowtie. When mapped to templates of the same sub-type (Figure 4.5(b, d)), the coverage plots for both SAGE and Bowtie show less variance when compared to the coverage plots of the other templates (Figure 4.5(c, e)). Bowtie does not take into account variance of coverage and scores the template of a different sub-type (FH13-A/BA2) higher than the template of the same sub-type (F06-A/BA1). On the contrary, SAGE penalizes for the variance in coverage, and correctly predicts the sub-type of the donor.

Furthermore, if several possible mappings of a read are given, SAGE can be used to determine the best mapping. In Figure 4.5(b, c), we see less variability in the coverage plots from SAGE's matching compared against those of Bowtie's matching (Figure 4.5(d, e)). Therefore, even if Bowtie is able to determine the correct donor template, it may not output the correct mapping.

Running time: For a data-set with n reads and a total of m read mapping locations, SAGE scales as $O(nm + n^2 \log n)$. Thus, on our data-sets with haploid genomes of average length 166Kbp (166 1000bp-segments), and $\sim 24,900$ reads, SAGE ran in 21 seconds. The running time increased to 210 seconds for the average diploid genome ($\sim$ 332 1000bp-segments, $\sim 49,800$ reads). Running times were recorded using a 4 core Intel 2.66GHz processor with 9Gb of RAM.

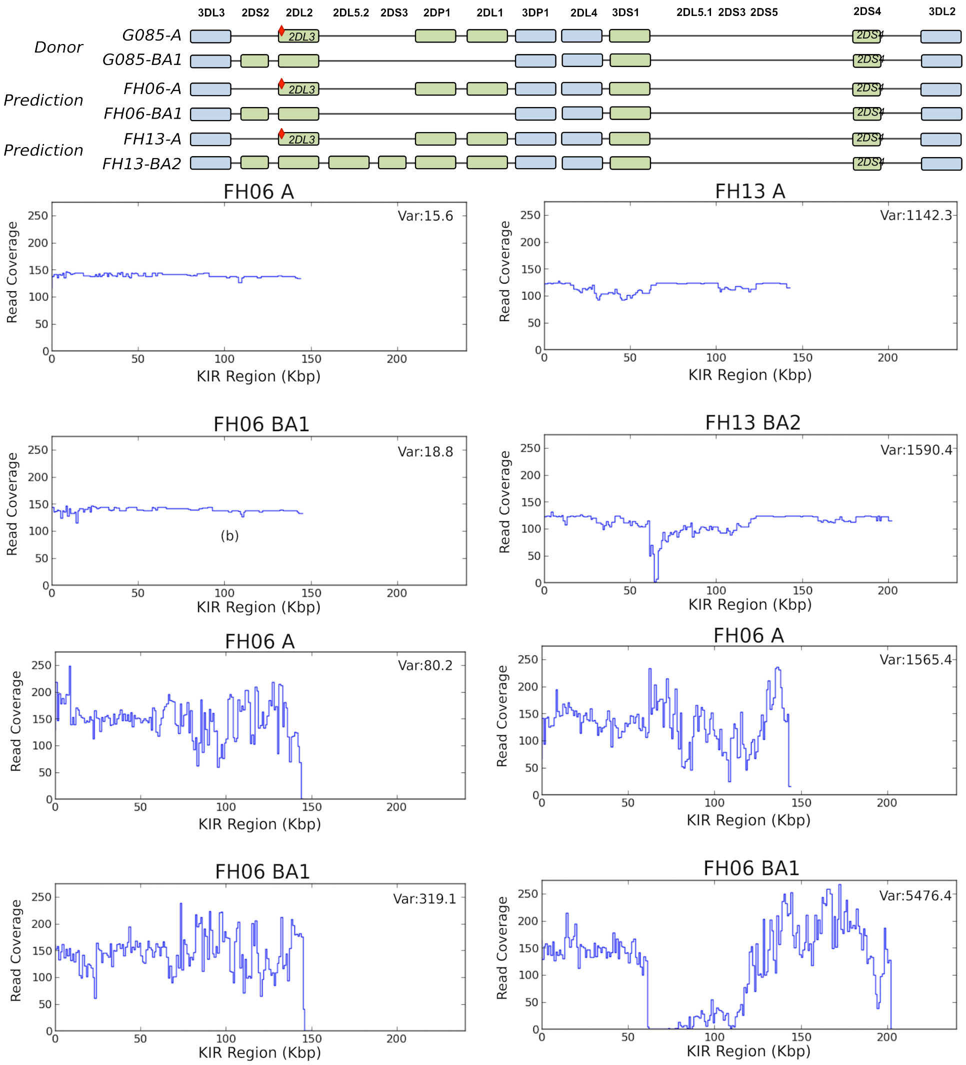


Figure 4.5: Coverage plots for reads sampled from G085-A/BA1 templates. (a) genomic architecture of of G085-A/BA1, FH06-A/BA1, and FH13-A/BA2. SAGE coverage plots when reads are extracted from G085A/BA1 and mapped to (b) FH06-A/BA1 and (c) FH13 A/BA2. Bowtie coverage plots when reads are extracted from G085A/BA1 and mapped to (d) FH06-A/BA1 and (e) FH13-A/BA2.

4.5 Discussion and Conclusions

To the best of our knowledge, SAGE is the first tool that scores predicted donor templates given a set of sequenced reads. Our results on the KIR region show that SAGE can be used to predict the sub-type of the donor KIR template, and can be directly used for haplotyping this region. Furthermore, SAGE scores the correct template higher than even templates of the same sub-type. Haplotype analysis of the KIR region is medically motivated due to the region's role in the human immune system. However, the genomic complexity (i.e. repetitive nature and variable gene architecture) of this region makes it difficult to do a complete analysis. Indeed, the possible sub-types of this region have not been completely characterized. Thus, reconstruction of this region and other complex regions of the genome, remains a worthwhile problem. Here we took the first step in the reconstruction.

While we focused our attention on the KIR region, SAGE is general enough to be applied to any complex region. It is also possible to implement many different scoring functions, which would allow the user to obtain optimal matchings according to his own custom scores. For example, read un-matching penalties may be constant for all reads, or may be read-specific. A motivation for read specific costs is in the case where the sequencing phase produces some sequencing qualities for reads, and it is possible to "pay" less when not matching reads of lower sequencing quality. Similarly, it is possible to choose a segmentation of the prediction in which all segments are of the same length, and uniform coverage is assumed, or one with variable segment lengths and possibly different coverage cost functions for each segment. A motivation to such complex segmentation is e.g. in the case where one tries to identify a specific structural variation, such as a deletion of a segment of specific length around a specific region of the prediction. Setting lengths of segments in the examined region to the expected deletion length can increase the likelihood that an optimal matching would not add artifact matchings of reads to a long segment spanning the deleted segment, in order to compensate for low coverage of the deleted segment. Lastly, by using different coverage cost functions, it is possible to decide the rate in which penalty increases due to deviations of expected coverage, which may grow linearly, polynomially, exponentially, or based on other probabilistic models, as long as the function satisfies the convexity requirement.

Future work would involve extending the use of SAGE on real data. Some challenges in dealing with real data include obtaining the set of reads extracted from the region of interest (especially when sequencing data is likely taken from the whole genome) and providing the expected coverage. If we know the parameters of the sequencing run, we could use the target read coverage as the expected coverage; however, if that is unknown, we may be able to estimate the expected coverage from the number of reads we need to map to the region. For example, if we assume a uniform distribution of coverage, then the expected coverage is simply the total length of the reads over the length of the genome.

Currently, SAGE provides a scoring function for predicted templates based on their similarity to the true donor. Therefore, it might be possible to obtain a complete reconstruction of the donor genome by iteratively refining predicted donor templates until SAGE scores are optimized. Furthermore, SAGE can also be applied for scoring *de-novo* assemblies and for comparing the accuracies of different assemblers. Indeed, this tool is the first that scores predicted donor templates given a set of sequenced reads and can be used as the first step in reconstructing complex regions of the genome.

4.6 Appendix: NP-Hardness of CSM

In this section we show that for general (non-convex) scoring functions, the CSM problem is NP-Hard. This is proven by reducing the NP-Hard problem SAT to CSM.

Let $B = \{b_1, b_2, \ldots, b_n\}$ be a set of boolean variables. An *assignment* for B is a function $A : Barrow\{true, false\}$. A *CNF clause* ϕ over B is a boolean clause of the form $\phi = (x_1 \vee x_2 \vee \ldots \vee x_k)$, where each literal x_i is either some variable $b \in B$, or a negation $\neg b$ of some variable $b \in B$. For an assignment A for B and a variable $b \in B$, define $A(\neg b) = \neg A(b)$. The clause ϕ is *satisfied* by A if $A(x) = true$ for at least one literal x appearing in ϕ. A *CNF formula* ψ over B is of the form $\psi = \phi_1 \wedge \phi_2 \wedge \ldots \wedge \phi_m$, where each ϕ_j appearing in ψ is a CNF clause. The CNF formula ψ is *satisfied* by an assignment A if all clauses in ψ are satisfied by A. Say that ψ is *satisfiable* if there exists some satisfying assignment for ψ. The CNF-SAT problem is, given a CNF formula ψ over a set of variables B, to decide whether ψ is satisfiable. CNF-SAT is a well known NP-Complete problem Cook (1971); Garey and Johnson (1979). Next, we show that CNF-SAT can be reduced to CSM in a polynomial time, proving NP-Harness of CSM.

Given a CNF formula $\psi = \phi_1 \wedge \phi_2 \wedge \ldots \wedge \phi_m$ over the set of variables $B = \{b_1, b_2, \ldots, b_n\}$, the reduction constructs the matching instance (X, Y, w) as follows:

- The set X contains an element for each literal of B, i.e. $X = \{b_1, \neg b_1, b_2, \neg b_2, \ldots, b_n, \neg b_n\}$.
- The set Y is the union of two subsets $Y^b = \{y^b_1, y^b_2, \ldots y^b_n\}$ and $Y^\phi = \{y^\phi_1, y^\phi_2, \ldots y^\phi_m\}$. Each element $y^b_i \in Y^b$ corresponds to a variable $b_i \in B$, and each element $y^\phi_j \in Y^\phi$ corresponds to a clause ϕ_j of ψ.

- The cost function w defines the following matching and coverage costs:
  - For each $1 \leq i \leq n$, set $w_m\left(b_i, y_i^b\right) = w_m\left(\neg b_i, y_i^b\right) = 0$, and $w_m\left(x, y_i^b\right) = 1$ for $x \notin \{b_i, \neg b_i\}$. In addition, set $w_m\left(x, y_j^\phi\right) = 0$ if the literal x appears in ϕ_j, and otherwise set $w_m\left(x, y_j^\phi\right) = 1$.
  - For each $x \in X$, let d_x be the number of clauses in ψ containing the literal x. Set $w_c(x, 0) = w_c(x, d_x + 1) = 0$, and $w_c(x, a) = 1$ for $a \notin \{0, d_x + 1\}$. In addition, $w_c\left(y_i^b, 1\right) = 0$ and $w_c\left(y_i^b, a\right) = 1$ for $a \neq 1$, and $w_c\left(y_j^\phi, 0\right) = 1$ and $w_c\left(y_j^\phi, a\right) = 0$ for $a \geq 1$.

Claim 4. *ψ is satisfiable if an only if $CSM(X, Y, w) = 0$.*

Proof. For the first direction of the proof, assume that ψ is satisfiable, and let A be a satisfying assignment for ψ. We show that in this case $CSM(X, Y, w) = 0$.

Construct the matching M between X and Y by adding for each $x \in X$, such that $x = b_i$ or $x = \neg b_i$ and $A(x) = true$, the pair (x, y_i^b), as well as every pair (x, y_j^ϕ) such that x appears in ϕ_j. By the reduction design, for each $(x, y) \in M$, $w_m(x, y) = 0$. In addition, it is straightforward to observe that for every $x \in X$ we have that either $c_M(x) = 0$ (when $A(x) = false$) or $c_M(x) = d_x + 1$ (when $A(x) = true$), and therefore from the reduction design $w_c(x, c_M(x)) = 0$. Moreover, for each $y_i^b \in Y^b$ we have that $c_M\left(y_i^b\right) = 1$ (since M contains exactly one pair among (b_i, y_i^b) and $(\neg b_i, y_i^b)$, and no other pair in which y_i^b participates), and for each $y_j^\phi \in Y^\phi$ we have that $c_M\left(y_j^\phi\right) \geq 1$ (since ϕ_j is satisfied by A and thus M contains at least one pair of the form (x, y_j^ϕ)). Therefore, from the reduction design, $w_c(y, c_M(y)) = 0$ for every $y \in Y$, and we get that $w(M) = \sum_{(x,y) \in M} w_m(x, y) + \sum_{z \in X \cup Y} w_c(z, c_M(z)) = 0$, and in particular $CSM(X, Y, w) \leq 0$. Since all costs defined by w are either 0 or 1, it is clear that $CSM(X, Y, w) \geq 0$, and thus $CSM(X, Y, w) = 0$.

For the other direction of the proof, assume that $CSM(X, Y, w) = 0$, and let M be an optimal matching between X and Y for which $w(M) = 0$. We show that in this case ψ is satisfiable.

Construct the assignment A for B, where $A(b_i) = \mathit{true}$ if and only if $(b_i, y_i^b) \in M$. Since $w(M) = \sum_{(x,y)\in M} w_m(x, y) + \sum_{z\in X\cup Y} w_c(z, c_M(z)) = 0$, we have that $w_m(x, y) = 0$ for every $(x, y) \in M$, and $w_c(z, c_M(z)) = 0$ for every $z \in X \cup Y$ (since all matching and coverage costs defined by w are nonnegative). In particular, M contains only pairs of the form (b_i, y_i^b), $(\neg b_i, y_i^b)$, and (x, y_j^ϕ) such that x appears in ϕ_j (all other pairs have a matching cost of 1). In addition, for every $y_i^b \in Y^b$, in order to get $w_c\left(y_i^b, c_M\left(y_i^b\right)\right) = 0$ it must hold that $c_M\left(y_i^b\right) = 1$ (from the definition of w_c) and thus M contains exactly one of the pairs (b_i, y_i^b) or $(\neg b_i, y_i^b)$. This implies that for $x = b_i$ or $x = \neg b_i$, $(x, y_i^b) \in M$ if and only if $A(x) = \mathit{true}$. Next, for every $y_j^\phi \in Y^\phi$, in order to get $w_c\left(y_j^\phi, c_M\left(y_j^\phi\right)\right) = 0$ it must hold that $c_M\left(y_j^\phi\right) \geq 1$ (from the definition of w_c), therefore M contains at least one pair of the form (x, y_j^ϕ) such that x appears in ϕ_j^2. For such a literal x, $c_M(x) > 0$, and to obtain $w_c(x, c_M(x)) = 0$ it must hold that $c_M(x) = d_x + 1$. Hence, M must contain (x, y_i^b) and all d_x pairs of the form $(x, y_{j'}^\phi)$ such that x appears in $\phi_{j'}$, and in particular $A(x) = \mathit{true}$, and ϕ_j is satisfied by A. As all clauses in ψ are satisfied by A, ψ is satisfiable.

□

It is immediate to observe that the reduction described above is polynomial, and since CNF-SAT is NP-Hard it follows that CSM is NP-Hard. It is also simple to formulate CSM as a decision problem (asking whether (X, Y, w) has a matching with cost of at most k for some argument k) and to design a non-deterministic polynomial time algorithm for it (which choses a matching M at random and checks whether $w(M) \leq k$), proving that CSM is NP-Complete.

4.7 Acknowledgements

Chapter 4, in full, is a reprint of the material as it appears in BMC Bioinformatics 2013. Lo, Christine; Kim, Sangwoo, Zakov, Shay; Bafna, Vineet. “Evaluating genome architecture of a complex region via generalized bipartite matching.” The dissertation author was the primary investigator and author of this paper.

Chapter 5

Barcode-based characterization of the KIR Region

5.1 Introduction

Natural killer (NK) cells are important mediators of the early immune response, by recognizing ‘abnormal’ cells such as tumor cells and pathogen infected cells. They express a broad array of inhibitory and activating receptors that balance the input from multiple incoming signals (Hsu et al. (2002)) . In humans and other primates, these receptors are encoded by genes of the killer cell immunoglobulin like receptor (*KIR*)) region, located in 19q13.4 as part of the larger Leukocyte Receptor Complex. This hyper-variable region encodes Ig-like receptors that respond to cytokine/chemokine signals from pathogen recognizing NK cells. Interestingly, while there are other genes with similar functionality in other mammals, the KIR gene cluster itself has not been observed in rodents. Comparative genomics suggests that gene duplication events led to the generation of the KIR genomic region 30-45M years ago. Rapid expansion during primate divergence led to high variability even within the primates. The receptors encode both inhibitory and activating receptors. The inhibitory receptors appear to be activated by recognizing “self” molecules and preventing an auto-immune response. In contrast, activating genes may function by providing NK cell activation in response to pathogens. Genomic analysis of the KIR region in individuals is fundamental to understanding their response to pathogens, success of transplantation, and other resources.

Typical whole genome or targeted sequencing approaches produce sequences of small lengths (100-150bp) where the sequences are paired-ends of larger inserts (300-500bp). Predicting the diploid KIR subtypes directly from whole genome sequencing (WGS) reads is increasingly motivated by the increasing prevalence of WGS data. However, the problem poses a serious computational challenge. Unlike other hypervariable regions which predominantly express allelic variation due to point mutations, the KIR region also shows variability in differing numbers of genes, with an individual carrying anywhere from 4 to 14 genes, and 2 pseudogenes (Figure 4.1A) (Middleton and Gonzelez (2010)) . Adding allelic variation, the total variability within KIR can be expressed in the form of variable gene content, gene-copy numbers, and gene polymorphisms. This complexity of variation is compounded by the observation that many distinct KIR genes are paralogous duplications and share significant sequence similarity. Thus, even mapping a KIR read to the correct location in light of duplications and sequence variation can be a challenge. For this reason, there are no existing tools to sub-type KIR based on WGS data.

In previous approaches, KIR sub-types have been determined by PCR amplifying and sequencing of specific target regions (Hsu et al. (2002)) . Quantitative PCR helps identify gene number. However, this approach is labor intensive, and imperfect in that it can only type known variants, and the experiments need to be done even when the complete genome sequences are available.

Recently, we proposed one of the first algorithms for KIR sub-typing (Lo et al. (2013b)) , in which reads mapping to the KIR region are *re-mapped* to known KIR templates (or haplotypes) using an algorithm that penalizes for deviation in coverage, as well as for sequence dis-similarity. The intuition is that if the template has the correct copy number of some gene, the re-mapped reads will have the smallest deviation in coverage as well as the smallest number of alignment edits. We used an algorithm based on combinatorial flow to optimize the cost function (Lo et al. (2013b)) . While the approach works well on simulated data, it turned out to be computationally expensive to adapt to WGS data. First, all reads must be mapped, and it is still difficult to separate repetitive sequence from the rest of the genome as belonging to KIR. A search of WGS Illumina 100bp reads based on mapping to the repeat-masked KIR region resulted in a huge number of reads that also mapped to other regions of the genome. Down-weighting or elimination of repetitive reads is possible but has the danger of removing truly paralogous KIR gene sequences which also map to multiple locations.

In this manuscript, we use a combination of techniques to resolve the KIR region on copy-number and allelic variation. We also use a model based approach that exploits the tight organization of the KIR region. The KIR region is organized into two distinct genomic regions. The centromeric regions anchored by the genes, 3DL3 and 3DP1, and the telomeric region is anchored by the genes 2DL4, and the pseudogene 3DL2. We also note, but do not enforce, that a haplotype contains exactly one of genes 2DL2 and 2DL3, and exactly one of 3DS1 and 3DL1 on the telomeric side, suggesting that the genes are allelic variants from a common gene, but diverged enough to be different in sequence. On the other hand, the centromeric and telomeric sides can each contain 0 or 1 copy of the genes 2DS3, 2DS5, and the genes are highly similar to be difficult to differentiate the reads. Recent studies have suggested that KIR typing is incompletely developed, and is likely to change as more data is acquired. However, even in these studies showing over 50 KIR haplotypes based solely on variation in gene copy number, these rules appear to be observed. Thus, any analysis of the KIR region must take these rules into account, but allow for the possibilities of novel haplotypes.

To resolve these challenges, we use an alignment-free approach based on read counts of fixed length-indicator strings (k-mers). First, we mark the order of genes based approximately on genomic location as:

$$
\begin{gathered}
E = (\text{3DL3, 2DS2, 2DL2, 2DL3, 2DL5, 2DS3/5, 2DP1,} \\
\text{2DL1, 3DP1, 2DL4, 3DL1, 2DS1, 2DS1, 2DS4, 3DL2)}
\end{gathered}
$$

Table 5.1: **Copy number representation of different KIR types.**

KIR Type	**Copy number representation**
A	[1,0,1,0,0,0,1,1,1,1,0,1,0,1,1]
AB	[1,0,1,0,1,1,1,1,1,1,1,0,1,0,1]
BA1	[1,1,0,1,0,0,0,0,1,1,0,1,0,1,1]
BA2X	[1,0,1,0,1,1,1,1,1,1,0,1,0,1,1]
BA2	[1,1,0,1,1,1,1,1,1,1,0,1,0,1,1]
Bdel	[1,1,0,1,1,1,0,0,1,1,1,0,1,0,1]
B	[1,1,0,1,2,2,1,1,1,1,1,0,1,0,1]

Thus $e_0 = \text{3DL3}$ and so on. Note that $e_5 = 2DS3/5$ represent copies that may occur on either the centromeric and/or telomeric side. Counts of indicator strings (k-mers) are used to determine copy numbers of genes in the sampled genome as well as presence and absence of KIR genes E. Specifically, we output a vector $\mathbf{C}$, where $\mathbf{C}_i$ gives the number of copies of e_i. The copy numbers corresponding to common KIR haplotypes and are given in Table 5.1. While the combination of gene copy numbers is unique enough to differentiate among haploid KIR types, a few pairs (diploid) KIR types are equivalent on the gene copy number level. For example, the diploid copy number representations of (A, B) and $(AB, BA2)$ are both $[2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 2]$. Similarly, (A,Bdel) and (AB,BA1) are equivalent and (Bdel,BA2) and (BA1,B) are equivalent at the gene copy number level. In our approach, we predict only the diploid copy numbers, treating all equivalent haplotype pairs as equivalent, or inferring likely ones based on frequency in the population. Additional long range haplotype information, (not available in short-read sequencing) can later be used to phase to the correct haplotype pairs, but is not part of the tool.

Once the diploid copy numbers have been computed, we use sequence information to determine the allelic sub-types for each copy in all genes. For each gene, we pre-compute a list of polymorphic genomic locations, and known allelic variants at that location. Each allelic variant is a selection of a specific allele at each variant locus. For any sample, we recruit and align reads that map to all allelic copies of the gene. We use the mappings to call variants, and sub-type the alleles.

The KIR is just one of many hypervariable regions, which include the 1Mb LRC region on Chr19, the 3.6Mbp Human Leucocyte Antigen (HLA) region, and others. Automated tools that can mine genome sequence data to sub-type these complex regions has direct implications for many autoimmune diseases, susceptibility to other disorders including diabetes and cancer, as well as the success of transplantation (Rajagopalan and Long (2005); De Re et al. (2011)) .

5.2 Results

5.2.1 Data sets

KIR Templates. To simulate data-sets for tests, we used the Immuno Polymorphisms database (IPD) (Robinson et al. (2013)) that lists 23 complete KIR haplotypes (denoted as the set T_h), each classified into one of 7 sub-types as follows:

$$\{A(10), AB(1), BA1(3), BA2(3), BA2X(1), Bdel(2), B(3)\}$$

Each sub-type contains a unique combination of copy numbers of the genes in the KIR region, and can be represented as a vector of copy numbers ordered according to a given gene order (see Table 5.1).

KIR Alleles. The IPD-KIR database (Robinson et al. (2013)) (Release 2.5.0 October 2013) lists a total of 678 KIR alleles for all the genes in the KIR region. The information per gene is summarized in Table 5.2. The complete genomic sequence of many of the alleles in the database is not available and those are not used for allelic typing in this paper. Table 5.2, Column 3 lists the different allelic types used by us. It is worth noting that we treat the genes as independent, and choose the right combination from an enormous number of possibilities. Kit also accepts a user-supplied database of known alleles.

Simulated Data. We use the notation $\mathcal{G}[t]$ to denote the hg19/b37 reference genome $\mathcal{G}$, with template t replacing the KIR region in $\mathcal{G}$. The **scaled-haploid** data-set consisted of $\mathcal{G}[t]$ sampled with reads at a uniform coverage of $c = 15\times$ with no errors or variants for all $t \in T_h$. We also paired the haplotypes to create the set T_d of $\binom{23}{2} + 23 = 276$ distinct template pairs. The **scaled-diploid** data-set

Table 5.2: **KIR Alleles.** This table lists for each gene the number of alleles in the IPD-KIR database. Alleles which we do not have the complete genomic sequence for are filtered out so only a subset of alleles are used for determining allelic type. All alleles are aligned to a gene reference sequence which is the gene sequence in b37/hg19 if available or a randomly selected allele of the gene. Polymorphic sites are determined from the multiple alignment of these alleles.

Gene	**IPD-KIR**	**Typing**	**Polymorphic sites**	**Gene Reference**
KIR3DL3	108	23	95	hg19
KIR2DS2	22	12	16	2DS2*0010101
KIR2DL3	44	16	197	2DL3*0010101
KIR2DL2	29	14	39	2DL2*0010101
KIR2DL5B	43	23	127	2DL5A*001010101
KIR2DS3B	33	12	355	2DS3*0010301
KIR2DP1	28	21	169	hg19
KIR2DL1	44	20	227	hg19
KIR3DP1	23	22	36	hg19
KIR2DL4	52	23	88	hg19
KIR3DS1	17	4	209	hg19
KIR3DL1	78	21	424	hg19
KIR2DS1	15	6	11	2DS1*0020101
KIR2DS4	30	17	349	hg19
KIR3DL2	86	22	172	hg19

consisted of $G[\tau]$ (for all $\tau \in T_d$), sequenced at perfectly uniform coverage of $c = 30\times$ with no errors or variations.

To test if our method is robust to deviations in coverage and errors in WGS experiments as well as point mutations, we introduced variants in G using an in-house genome simulator to create two haplotypes, G_1, G_2. Single nucleotide polymorphisms (SNPs) from dbSNP were induced by applying an average SNP rate of 0.001. Furthermore, the ratio of homozygous/heterozgyous variants SNVs was set to 0.33 based on previous observations (Kim et al. (2013)) . From the germline diploid, we generated a somatic diploid genome by inducing SNVs at random locations with an average rate 0.00001. Finally the generated germline diploid contained 280,473 SNVs compared to the reference genome (224,536 heterozygous and 55,936 homozygous SNVs) and 30,995 somatic variants were added. For a **simulated-haploid** data-set, 2 × 100bp paired-end reads were generated from $G_1[t]$ and $G_2[t]$ for each $t \in T_h$ at $5\times, 15\times, 30\times$, and $50\times$ coverage using GemSIM under the Illumina GA error model (McElroy et al. (2012a)) . Likewise, a **diploid-simulated** data-set was constructed by combining reads from two haplotypes for a total coverage of $10\times, 30\times, 60\times$, and $100\times$.

Real Data. For assessment of our method on real data, we used whole genome sequencing data for 5 parent-offspring trios from different populations in the 1000 Genomes Project. The **1000G** data-set included the CEU trio (NA12877, NA12878, NA12882) sequenced at 250× coverage, CEU trio (NA12889, NA12890, NA12877) (13× coverage), PUR trio (HG00731, HG00732, HG00733) (7× coverage), and KHV trio (HG02026, HG02025, HG02024) (5× coverage). The sequence reads were obtained from NCBI SRA (Kodama et al. (2012)) (Table 5.7). Finally, we tested on an **Icelandic** dataset comprising of 2649 whole genome sequenced individuals among which contained 289 trios.

5.2.2 An overview of the method

The main result of the paper is a novel barcode-based method Kit, to identify KIR types directly from paired-end WGS Illumina reads without explicit read mapping or assembly. An overview is presented here, with details in Section 5.2.4.

Let G be the ordered set of KIR genes numbered $1 \ldots |G|$. Recall that the variability in the KIR region is mediated by changes in copy numbers of each of the genes, as well as allelic variation in the gene sequences. Correspondingly, Kit has two parts: Kit-CN outputs for each sample, the vector

$$\vec{p} = [p_1, p_2, \ldots, p_{|G|}]$$

where p_g describes the copy number of each gene $g \in G$. Second, for each gene g, where $1 \leq p_g \leq 2$, Kit-AT outputs allelic sub-types of each of the gene copies.

The first part, Kit-CN, is based on the notion of *barcodes*. For gene g, a barcode is constructed with respect to a set of indicator strings F_g, chosen from the set of k-mers ($k = 50$) from the gene in all known KIR templates. Genomic sequence fragments from a sample S are used to construct a barcode, B_g^S by counting the number of occurrences of each fragment in F_g in S. Similarly, a barcode is constructed corresponding to a single copy of the gene, B_g^1. We use B_g^S, and B_g^1 to compute a *scaling-factor* s_g for each gene. In general, $s_g \simeq cp_g$, where c is the global sequence coverage (constant for all genes), and p_g is the number of copies of gene g in the sample. Kit-CN estimates the copy number that minimizes a penalty function (See Section 5.2.4)

$$\vec{p}^* = \arg\min_{c,\vec{p}} \mathcal{D}(\vec{s}, c, \vec{p}) = \arg\min_{c,\vec{p}} \sum_g \mathcal{D}(s_g, cp_g)$$

Note that the integer array of copy numbers does not give us haplotype pair information. We infer the diploid KIR type by comparing to the integer copy number array of known KIR type pairs. A match is conservatively assigned to the known pair.

The second part, Kit-AT, determines the allelic type of each copy of each gene, also with a maximum likelihood computation. It recruits all reads that map to a specific gene, and aligns them using the aligner bwa. The set of nucleotides mapping to a specific position is used to compute the most likely pair from a known set of alleles in IPD. In the current classification scheme (Robinson et al. (2013)) , allelic types have up to seven digits. The first three digits are used to distinguish alleles that differ in the sequence of the encoded protein. The fourth and fifth digit distinguish between alleles with synonymous differences in the sequence of their coding region while the last two digits distinguish allelic variants in the non-coding region. The resolution of allelic typing can be adjusted based on the user's preference, but a five digit resolution is considered state-of the art, and most commercial typing goes up to a three digit resolution. Kit-AT predicts the alleles for each gene separately and outputs a ranking of allelic types sorted by likelihood (see Section 5.2.4).

5.2.3 Copy number validation

We validated the copy number and allelic typing separately. Kit-CN was applied to scaled-haploid and scaled-diploid data-sets and all samples were correctly typed. Next, we tested on simulated-haploid, and simulated-diploid data-sets. As sequence variation can influence barcodes, we tested performance with sequence coverage chosen from $\{5\times, 15\times, 30\times, 50\times\}$ for haploid case, and $\{10\times, 30\times, 60\times, 100\times\}$ in the diploid case. Table 5.3 summarizes the results. Except for a few erroneous calls in the low coverage diploid data sets, all haploid samples and higher coverage diploid samples were typed correctly.

Kit-CN makes predictions by minimizing a penalty function described in Section 5.2.4. The penalty score of the prediction can be used as a level of confidence for the prediction. Figure 5.1a shows the score distributions of the various datasets.

For the available data-sets (1000G, and Icelandic), the true copy numbers are not known, but parent-child trios can be tested for consistency. In each trio that could be resolved into known haplotypes, we verified if the child had one haplotype

Table 5.3: **Typing the KIR region.** Accuracy of Kit-AT on simulated data.

Dataset	**Correctly typed**
Haploid 5x	23/23
Haploid 15x	23/23
Haploid 30x	23/23
Haploid 50x	23/23
Haploid Scaled	23/23
Diploid 10x	265/276
Diploid 30x	276/276
Diploid 60x	276/276
Diploid 100x	276/276
Diploid Scaled	276/276

that matches the type of the mother and the other haplotype was transmitted from the father. For a novel copy number vector, the child would have no more copies of a gene than both parents combined in order for the trio to be consistent. All 4 1000G trios and 288 of the 289 Icelandic trios were consistent (Tables 5.4).

Table 5.4: **Typing trios from 1000 Genomes.**

	Sample	**Score**	**Predicted type**
Father	NA12877	1.467	A,BA1
Mother	NA12878	2.419	AB,BA2
Child	NA12882	1.507	A,AB
Father	NA12889	2.414	A,BA1
Mother	NA12890	1.972	A,BA1
Child	NA12877	1.467	A,BA1
Father	HG00731	3.376	2, 2, 0, 2, 1, 1, 1, 1, 3, 3, 1, 2, 0, 2, 2
Mother	HG00732	4.444	A,AB
Child	HG00733	1.914	2, 1, 1, 1, 1, 1, 2, 2, 3, 3, 1, 2, 0, 2, 2
Father	HG02024	4.359	2, 1, 2, 1, 1, 1, 1, 2, 2, 2, 1, 1, 1, 1, 2
Mother	HG02025	3.279	B,Bdel
Child	HG02026	5.029	2, 1, 1, 1, 2, 2, 2, 2, 2, 2, 1, 1, 2, 1, 2

Inheritance patterns can also be used to distinguish between type pairs that are copy number equivalent. For example, for the CEU trio, the KIR type of

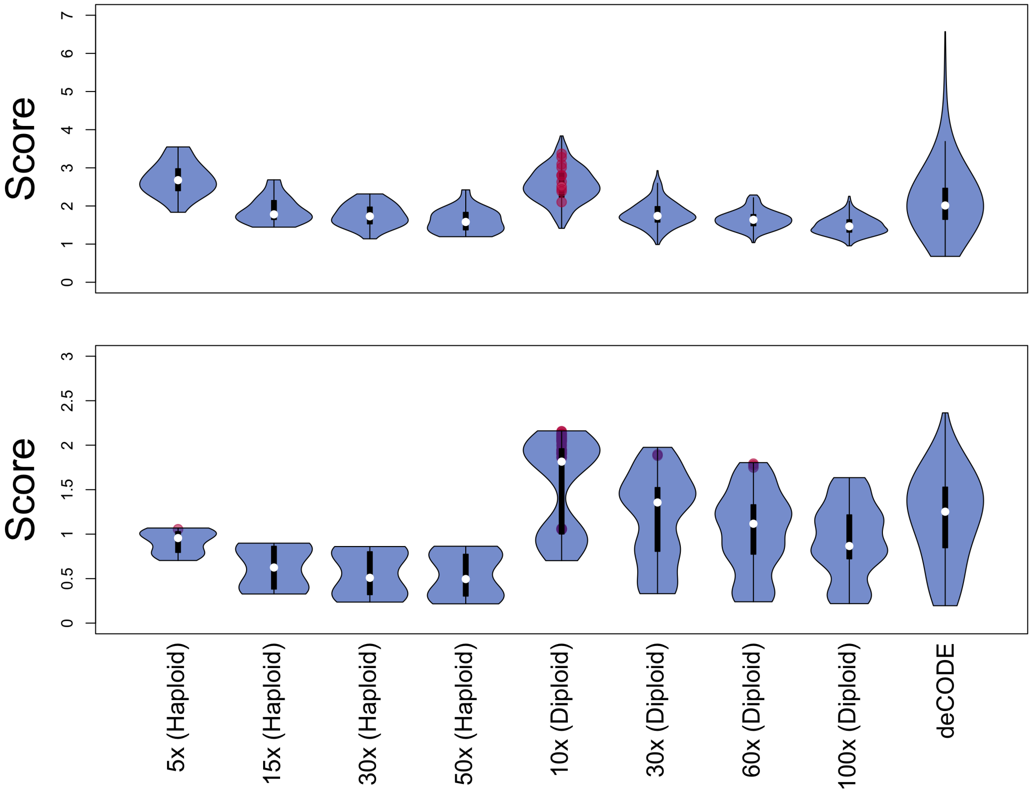


Figure 5.1: **Score distribution of Kit.** (A) Kit-CN score distribution on various datasets. In the haploid-simulated datasets there were 23 samples. In the simulated diploid datasets there were 276 samples. The Icelandic dataset contains 2649 individuals. In the simulated data sets, the red marks indicate the scores for the incorrectly typed samples. (B) Kit-AT score distribution for gene KIR3DL1 on various datasets. In the haploid-simulated datasets there were 17 samples with KIR3DL1 and in the diploid-simulated datasets there were 255 samples with KIR3DL1. In the Icelandic dataset, 2547 samples with KIR3DL1. Again, the red marks in the simulated data sets indicate scores for incorrectly typed samples.

the father (NA12877) is $(A, BA1)$, while the mother (NA12878) can be resolved into (A, B) or $(AB, BA2)$. The KIR type of the child is (A, AB) (Table 5.4) is used to resolve the mother to $(AB, BA2)$ and phases the child as inheriting the A haplotype from the father and AB from the mother.

Table 5.5 shows the haplotype frequency of the unrelated individuals in the Icelandic dataset. In previous studies, it was estimated that the A haplotype has a frequency of 55% in Caucasian populations (Parham (2003)). The higher frequency of A haplotypes in Icelandic population may be attributed to the relative genetic homogeneity of the population (Helgason et al. (2003)). Of the 2086 unrelated individuals (known children not included), 274 were of novel type.

Table 5.5: **Frequency of known types in Icelandic population.**

Type	**Frequency**
A	2553
AB	628
BA1	552
BA2	350
B	61
Bdel	28
Novel Pairs	274
Total	**4172 + 548**

5.2.4 Allele typing

The allelic type for 237 genes of the KIR templates were previously determined using experimental methods (Pyo et al. (2010)) and provide a platform for validating the accuracy of our computational method. We called the allelic types on the known gene sequences in each template and compared to the types obtained previous via laboratory methods.

On the scaled-haploid data-set, Kit-AT typed all but 1 genes correctly (at the highest, 7-digit, resolution). The gene that was inconsistent with the experimentally known type was KIR2DL5B in sample FH08BA2X. It was typed previously as KIR2DL5B*00601, but further examination showed that among the 127 polymorphic sites of this gene, the filtered reads differed from KIR2DL5B*00601 at 42 of sites, while it only differed from KIR2DL5B*003 at 6 of the sites, suggested that the computationally typed allele matches the known genomic sequence of FH08BA2X better than the previously typed allele. To test our method on simulated data, we used the computationally typed alleles from the scaled-haploid data set as the gold standard.

Kit-AT was applied to the simulated-haploid and simulated diploid datasets. Not surprisingly, the accuracy increases with increasing coverage (Table 5.6). At 30x coverage, 11 of the 15 genes have a true positive rate higher than 95%. Furthermore, the scores themselves give a level of confidence in the prediction as incorrectly typed genes tended to have a higher score (see Figure 5.1b).

Table 5.6: **Allele Typing on Simulated Data.** Kit-AT was used to type the alleles of the 15 KIR genes on samples simulated with various sequencing coverage depth. This table shows the accuracy of the predictions at 5-digit resolution.

	Haploid 5x	Haploid 15x	Haploid 30x	Haploid 50x	Diploid 10x	Diploid 30x	Diploid 60x	Diploid 100x
3DL3	0.95	1	1	0.95	0.91	0.96	0.96	0.92
2DS2	1	1	1	1	1	1	1	1
2DL3	1	1	1	1	1	1	1	1
2DL2	1	1	1	1	1	1	1	1
2DL5B	0.9	0.9	0.9	0.9	0.74	0.76	0.77	0.84
2DS3B	1	1	1	1	1	1	1	1
2DP1	1	1	1	1	0.95	0.92	0.94	0.93
2DL1	1	1	1	1	0.82	0.88	0.83	0.75
3DP1	1	1	1	1	0.98	1	1	1
2DL4	1	1	1	1	1	1	1	1
3DS1	1	1	1	1	1	1	1	1
3DL1	0.94	1	1	1	0.8	0.99	0.98	1
2DS1	1	1	1	1	1	1	1	1
2DS4	1	1	1	1	1	1	1	1
3DL2	0.95	0.95	0.95	0.95	0.84	0.9	0.9	0.91

We also used Kit-AT to predict the allelic type of the Icelandic population. The 289 trios provided a level of validation. Genes where the child had copy number 1 or 2 were typed. Among the 2541 genes typed in the Icelandic trio dataset, only 19 genes among 15 trios were inconsistent with inheritance patterns.

5.2.5 Running times

The implementation of our barcode-based typing method runs very efficiently. Both Kit-CN and Kit-CT scale linearly to the number of reads. The majority of the runtime for Kit-CN is taken up by the barcode generation which needs to parse each read, while the majority of the runtime for Kit-AT is taken up by filtering of the reads. See Figure 5.2 for the runtime on various sized bam files on a 4 core Intel 2.66GHz processor with 9Gb of RAM.

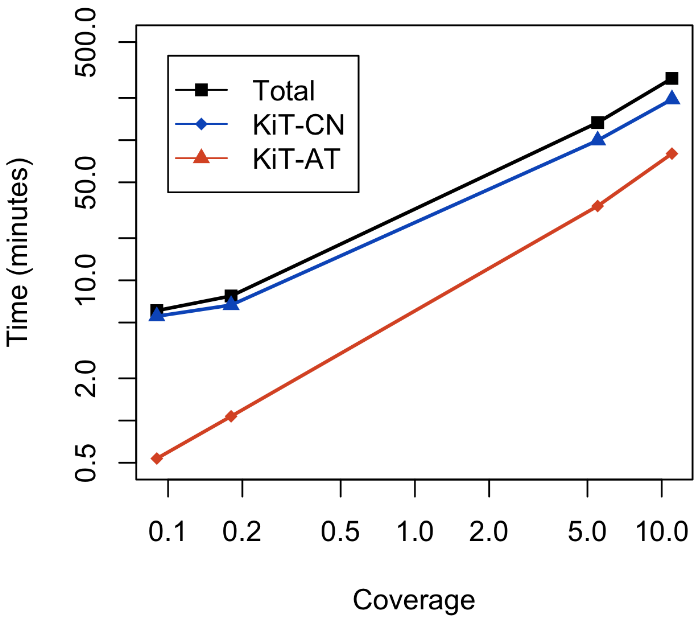


Figure 5.2: **Kit Runtime.** This log-log plot shows the runtime of Kit given sequence reads (in fastq format) at various coverage. Both Kit-CN and Kit-AT scale linearly with respect to the size of the input bam file. With 5x WGS reads, Kit will take approximately 2 hours to run. All runtimes are recorded on a 4 core Intel 2.66GHz processor with 9Gb of RAM.

5.3 Methods

5.3.1 Copy number inference with Kit-CN

Pre-processing. The first step in Kit-CN is the generation of *barcodes*, described in the next section. The barcodes are generated based on indicator strings constructed from available KIR sequences in a pre-processing step, described here.

While Kit-CN can accept any database of templates, we used the set of 23 full-length template sequences from the IPD database (Pyo et al. (2010)) , and denote the set as T. As mentioned earlier, we denote the entire reference genome as the string $\mathcal{G}$. For for any template sequence $t \in T$, we define $\mathcal{G}[t]$ as the reference sequence with the KIR region replaced with the sequence of KIR template t. Let G be the set of KIR genes. Note that each gene g may exist in multiple copy numbers within T, and the different copies may have different allelic variation (typically small nucleotide variation). We construct indicator strings for each gene g using the following procedure.

1. For each gene, $g \in G$, a collection of *candidate-indicator* strings are created as length k strings that appear in g in any template in T.
2. A candidate-indicator in gene $g_i \in G$ is selected as an *indicator* if it appears at least once in $\mathcal{G}[t]$ for any $t \in T$ that contains gene g, and does not appear in $\mathcal{G}[t']$ for all $t' \in T$ that do not contain gene g. In an effort to reduce the noise from repetitive regions of the genome (i.e ALU, LINE, SINE elements), candidates that appear more than 35 times (empirically chosen) in $\mathcal{G}[t]$ are discarded from the set of indicators. The set F_i denotes the set of indicator strings for $g_i \in G$. Note that the set is dependent on T, but we omit T for ease of exposition.

Generating barcodes. The general definition of a barcode is as follows: For sample (a collection of DNA sequences) S, the *barcode* with respect to an ordered set of *indicators* $F = (f_1, f_2, \ldots,)$ is defined as vector of integers B^S where for each i, $B^S[i]$ is the count of f_i in S.

To allow for fast barcode construction, we implemented the Aho-Corasick pattern matching algorithm (Aho and Corasick (1975)) , which takes a dictionary of words and generates a trie-like data structure with fail transitions in linear time with respect to the total dictionary length. This data structure allows, given a string s, to report all pairs (v, w) such that the w-th word in the dictionary appears as a substring of s starting at position v. The running time for analyzing s is linear in the length of s and number of output pairs (v, w). Following this, we construct a trie for each set of indicator strings (F_g for gene $g \in G$).

As part of the preprocessing step, barcodes are generated for each template in T. Let $\mathcal{G}$ be the reference genome. Denote t^* as the string encompassing the KIR region (chr19:55,235K-55,379K in b37/hg19) in $\mathcal{G}$. We use the trie for gene g to search $\mathcal{G}$ and construct $B_g^{\mathcal{G}}$, $B_g^{t^*}$, and for each template $t \in T$, B_g^t. For each template $t \in T$, KIR gene $g \in G$, we construct the barcode:

$$B_g^{\mathcal{G}[t]} = B_g^{\mathcal{G}} - B_g^{t^*} + B_g^t$$

Next, for each gene, we use the template set T to construct a barcode corresponding to a single copy for the gene. Let $p_g(t)$ denote the number of times gene $g \in G$ occurs in template $t \in T$. Define

$$B_g^1[i] = \frac{\sum\limits_{t \in T} B_g^{\mathcal{G}[t]}[i]/p_g(t)}{\#\{t : p_g(t) > 0\}}$$

Inferring scaling factors. For donor sample S, and KIR gene $g \in G$, and assuming uniform sequence coverage, we expect that B_g^S is a scaled version of B_g^1. Define the *scaling-factor* s_g as The scaling factor s^g between B_g^S and B_g^1 is computed as

$$s_g = \frac{\sum_i B_g^S[i]}{\sum_i B_g^1[i]}$$

and is used to create the following vector of scaling factors, $\vec{s} = [s_1, s_2, \ldots,]$ for all genes $0 \leq g \leq |G|$. Note that if the sequenced sample contains p_g copies of gene g, we expect the scaling factor $s_g = cp_g$ for some constant c related to sequencing coverage. Furthermore, c, is expected to be constant for all genes. Therefore, it is expected that $\vec{s} = c\vec{p} = c[p_1, p_2, \ldots, p_{|G|}]$. The discrepancy due to sequencing errors and variation in coverage sequencing is measured by a likelihood based distance function.

Computing barcode distance function. While many distance measures can be used, we report one that exhibits high classification success rates over simulated and real data (see Results). Our similarity measure is derived from the Normal distribution. Specifically, for gene that appears with a count p_g, and is sequenced with coverage c in a sample, we assume that s_g is Normally distributed with mean and variance both equal to cp_g. Thus, the p.d.f is given by

$$f(s_g|cp_g) = \frac{1}{\sqrt{2\pi cp_g}} e^{\frac{-(s_g - cp_g)^2}{2cp_g}}$$

We model the corresponding *penalty*, $\mathcal{D}$, for observing s_g as

$$\mathcal{D}_g(s_g|cp_g) = \frac{(s_g - cp_g)^2}{2cp_g} - \log \frac{1}{\sqrt{2\pi cp_g}}$$

Correspondingly,

$$\mathcal{D}(\vec{s}, c\vec{p}) = \sum_g \mathcal{D}_g.$$

Thus, Kit-CN takes sample sequence S as input, computes barcodes and $\vec{s}$, and returns

$$\arg\min_{c,\vec{p}} \mathcal{D}(\vec{s}, c\vec{p})$$

From practical reasons, for the case where $s_i = 0$ we replace the 0-variance by $\epsilon > 0$, in order to avoid infinite penalties. In order to make scores across samples comparable the scaling factor vector $\vec{s}$ is normalized by multiplying by the constant $\frac{200}{\sum_{1 \leq i \leq |G|} s_i}$.

Computing optimal copy numbers. If c is given, it is easy to find the integer barcode $\vec{p}$ that minimizes $\mathcal{D}(\vec{s}, c\vec{p})$. We use a simulated annealing procedure to estimate the optimal c and corresponding integer barcode $\vec{p}$.

```
procedure CopyNumberInference($\vec{s}$)
    1: $c = \frac{s^0}{2}$.
(* We start by assuming that there are two copies of the first framework gene. *)
    2: $\vec{p}_{best}$ = getCopyNumbers($\vec{s}$,$c$)
    3: For $1 \leq g \leq T$
        3.1: If $g < \frac{T}{2}$, $c_{new} = c \pm 0.05$; else $c_{new} = c \pm 0.01$
        3.2: $\vec{p}_{new}$ = getCopyNumbers($\vec{s}$,$c_{new}$)
        3.3: Set $\delta$ = distance($\vec{s}, c_{new}\vec{p}_{new}$) - distance($\vec{s}, c_{best}\vec{p}_{best}$)
        3.4: if $\delta > 0$, set $\vec{p}_{best} = \vec{p}_{new}$
        3.5: With probability $\max(1, e^{-\delta \frac{10i}{I}})$, set $c = c_{new}$
    4: Return $\vec{p}_{best}$.
end procedure
```

5.3.2 Allele typing with Kit-AT

Recruiting reads. Reconstruction of the genomic sequence in the KIR region is challenging with conventional assembly and mapping methods due to the high level of polymorphism and repetitive nature of the region. To overcome these obstacles, KIT-AT employs a pre-processing step using *unique indicators* to extract reads from the gene region of interest. A *unique indicator* of a gene g is a k-mer that appears only in g, and does not appear in the non-gene sequence of any other KIR template or the reference genome.

The set of unique indicators is derived from the known KIR templates T and $\mathcal{G}$, which we selected as the b37/hg19 reference sequence. As in KIT-CN, we used the Aho-Corasick (Aho and Corasick (1975)) trie to compute unique-indicators efficiently.

Selecting gene reference for allele typing. For most genes, we used the gene region of b7/hg19 as the reference. KIT-CN typed b37/hg19 to be of KIR type A. Therefore, the reference sequence for genes not in KIR haplotype A (e.g. KIR2DS2, KIR2DL5A/B, KIR2DS3/5, and KIR2DS1) were chosen arbitrarily among the known alleles (Table 5.2). Furthermore, the weak signal for KIR2DL3 in b37/hg19 was suspect and a known allele was arbitrarily chosen to be the reference for KIR2DL3 as well.

Likelihood function for allelic typing. For each reference gene, we align extracted reads using bwa (Li and Durbin (2010)) with default parameters. Again, we use a maximum likelihood computation; the length l of the reference gene g is chosen to be the length for all alleles, to normalize the likelihood computations.

Let $\mathcal{A}_g$ be the set of known alleles for gene g. Each allele is represented as a vector of strings $\vec{a} = [a_1, a_2, a_3, ..., a_l] \in \mathcal{A}_g$ where a_i is the nucleotide at position i. If the sample is predicted to have p_g copies of the gene, its allelic information will be represented by p_g alleles,

$$\vec{a}^{(1)}, \vec{a}^{(2)}, \ldots, \vec{a}^{(p_g)}.$$

However, we do not have the phasing information for alleles, and instead aim to identify the vector

$$\vec{\alpha}^{(p_g)} = [(a_{11}, a_{21}, \ldots, a_{p_g 1}), (a_{12}, a_{22}, \ldots, a_{p_g 2}), \ldots, (a_{1l}, a_{2l}, \ldots, a_{p_g l})]$$

Currently, we only handle cases where $1 \leq p_g \leq 2$; these represent the vast majority of cases and can be resolved using low coverage as well.

Each aligned read contributes a nucleotide at position i. Let d_i represent the *observed* data, or the collection of nucleotides at position i from the alignments of all mapped reads. Correspondingly, $\vec{d} = [d_1, d_2, d_3, ..., d_l]$ represents the total observed data. Assuming $p_g \leq 2$, and each position to be independent,

$$Pr(\vec{d}|\vec{\alpha}^{(p_g)}) = \begin{cases} \prod\limits_{1 \leq i \leq l} Pr(d_i|a_{1i}) & \text{if } p_g = 1 \\ \prod\limits_{1 \leq i \leq l} Pr(d_i|a_{1i}, a_{2i}) & \text{if } p_g = 2 \end{cases}$$

To calculate $Pr(d_i|a_{1i}, a_{2i})$ when $p_g = 2$, we represent the collection d_i by the triple (q_1, q_2, q_3) where q_1 is the count of reads in d_i consistent with a_{1i}, q_2 is the count of reads in d_i consistent with a_{2i}, and q_3 is the count of reads in d_i that don't match either a_{1i} or a_{2i}. Let ε be the sequencing error rate. Similarly, when $p_g = 1$, we represent d_i by the pair (q_1, q_3) where q_1 is the count of reads consistent with a_{1i}, and q_3 is count of all other reads. Then,

$$Pr(d_i|\alpha_i^{(p_g)}) = \begin{cases} \binom{q_1+q_3}{q_3}\varepsilon^{q_3} & \text{if } p_g = 1 \\ \frac{(q_1+q_2+q_3)!}{q1!\, q2!\, q3!} \frac{1}{2^{q_1+q_2}}\varepsilon^{q_3} & \text{if } p_g = 2 \end{cases}$$

Polymorphic Sites and Constructing $\mathcal{A}_g$. KIT-AT does not predict the alleles *de novo*, but instead selects the most likely pair from the extensive list of distinct alleles available in the IPD. The IPD-KIR Database lists a total of 678 alleles, but many of these have partial sequences. For each gene, we pre-extract a list of polymorphic sites from the IPD alleles. Next, an allele is kept as a candidate for a gene only if it includes at least 70% of the polymorphic sites. With this constraint, KIRtool-AT selects from 256 alleles, as shown in Table 5.2. For each gene and each pair of alleles (including the homozygous pair), a candidate $\vec{\alpha}^{(p_g)}$ is created for the likelihood computations. If for a candidate, a_{1i} or a_{2i} is unknown for some position i, KIRtool-AT scores all possibilities and selects the one that maximizes the likelihood.

Finally, to expedite the allele calling, the likelihood computations are limited to the extracted polymorphic sites, as all other sites will give the same value to the function.

5.4 Discussion and Conclusions

Given the proliferation of WGS data, we provide the first method for characterizing the KIR region directly from WGS reads. As the KIR region is marked by diversity in both gene content (including copy number variation) and allelic polymorphisms, KIT has two parts- one for inferring copy number and the other for allelic typing. Since KIR haplotypes are grouped by gene content, our method also has the ability to detect potentially novel haplotype groups in the population. The method, based on indicative strings and barcodes, is fast and accurate and does not require conventional assembly or mapping of the reads to a reference sequence, both techniques which can be challenging in regions which are repetitive and hyper-variable.

Finally, the computational techniques used in this manuscript are general enough to elucidate other complex regions of the genome such as the HLA region, which is also related to the immune system. In fact, the balancing selection in the KIR and HLA regions help maintain the advantageous diversity of the immune system in the human population (Norman et al. (2013)). As such, characterizing these regions are extremely relevant to personalized medicine and human population studies.

5.5 Appendix

Table 5.7: **Accession numbers for Trios from 1000 Genomes.**

Sample	Accession number	Platform	Coverage
NA12877	ERS179576	Illumina HiSeq 2000	189x
NA12878	ERS179577	Illumina HiSeq 2000	230x
NA12882	ERS179578	Illumina HiSeq 2000	462x
NA12889	ERX168847	Illumina HiSeq 2000	10x
NA12890	ERX168848	Illumina HiSeq 2000	16x
NA19238	ERX283213	Illumina HiSeq 2500	71x
NA19239	ERX283214	Illumina HiSeq 2500	71x
NA19240	ERX283215	Illumina HiSeq 2500	72x
HG00731	SRX028935	Illumina HiSeq 2000	6.7x
HG00732	SRX028921	Illumina HiSeq 2000	14x
HG00733	SRX254975	Illumina HiSeq 2000	5x
HG02024	SRX018743	Illumina HiSeq 2000	5.5x
HG02025	SRX015053	Illumina HiSeq 2000	4.4x
HG02026	SRX018741	Illumina HiSeq 2000	5x

5.6 Acknowledgements

Chapter 5, in part, is currently being prepared for submission for publication of the material. Lo, Christine; Bafna, Vineet. The dissertation author was the primary investigator and author of this paper.

Chapter 6

Conclusion

High throughput sequencing technology has advanced tremendously in recent year motivating the development of computational methods to leverage the technology's full potential. Current analysis methods based on mapping reads to a reference or assembling the genome allow us to capture the majority of the variation. However, they are unable to capture more complex variation including the combination of alleles on a single chromosome and variation in highly repetitive, hyper-variable regions of the genome such as the KIR region.

The onset of newer sequencing technologies allows for user specification of experiments. Unlike previous work in haplotype assembly, which focused on improving haplotype accuracy assuming specific technological parameters were dictated by sequencing technology, we address the question of what considerations (i.e. parameter choices) one should make in order to achieve long haplotypes in a cost-effective manner. We address this by parameterizing two variations of traditional sequencing technologies: strobe sequencing and clone-based haplotyping. In general, the cost of sequencing experiments can be split into two parts: cost of library preparation and cost of sequencing. The sequencing cost is proportional to the number of base pairs sequenced while the library preparation cost involves the process of extracting the DNA and preparing it for sequencing. It is known that long fragments are necessary to link distal heterozygous variants in order to reconstruct longer haplotypes. However, longer fragment lengths are usually associated with technologies that have higher library preparation cost. For example, on the one hand, next generation sequencing technologies are high throughput and relatively inexpensive. However, the read lengths are prohibitively short and the feasibility of assembling meaningful haplotypes with next generation sequencing has been questioned. On the other hand, the recent technology to isolate single cells (i.e. microdissection, customized microfluidic devices, florescent based cell sorting) have laid the foundation for haplotyping technologies that isolate single cells and then separate chromosome pairs in mitosis before sequencing. These technologies achieve chromosome length haplotypes but have high "library preparation" cost. Strobe sequencing is a technology that allows for variable advance lengths and multiple subreads. Our results are based on keeping the cost of sequencing constant. Keeping total read length fixed, we show that the most important parameter appears to be flexibility in choosing advance lengths. The optimal design for haplotyping favors a distribution of advance lengths that is heavily skewed towards the longer advances as these fragments help to connect distal heterozygous sites and improve haplotype length. Clone-based haplotyping is a method based on using clones to achieve longer fragments of DNA, pooling clones, and barcoding clones before sequencing. By absorbing a bit more laboratory preparation cost, these methods achieve longer fragments, and thus longer haplotypes. In our parameterization of clone-based haplotyping, we challenged the convention of having medium length clones and a large amount of pools and demonstrated, theoretically and empirically, the effect of having long clones with small number of pools on haplotype length.

We applied this concept using BAC clones on PGP1 to achieve haplotypes over a mega-base longer than other clone-based haplotyping methods to date. With technology improvements, it may be possible to separate long strands of DNA for improved haplotyping. Nevertheless, there will always be trade-offs between quality and effort; our techniques help to formalize and explore these trade-offs.

We also address the problem of characterizing highly repetitive, hypervariable regions directly from whole genome sequencing data. This dissertation includes two approaches to this problem. In our first approach, we developed a method, called SAGE that scores predicted reconstructions of the genomic region given a set of sequenced reads from the region. Our results on the KIR region showed that SAGE can be used to predict the haplotype group of the donor by scoring the reads against a set of known KIR haplotype sequence of different haplotype groups. In practice, it is quite difficult to extract the set of sequenced reads from a particular region of the genome given whole genome sequencing data. This fact motivated our second approach to the problem. In our second approach, called Kit, we developed a method that determines and applies indicative strings to directly determine copy number and allelic type of KIR genes from whole genome sequencing data. Our methods provide high throughput characterization of the KIR region and will open the door to larger population studies of this region. Gene-disease association studies in the KIR region should be of particular interest due to the role of the KIR genes in the immune system. Associations at different resolutions (i.e. haplotype group, allelic type, and variants) will provide insight into a more complete understanding of the impact of the KIR region. Our study of the KIR region is only the beginning of research in complex regions. The methods we describe in this dissertation, while they were applied to the KIR region, are general enough to be applied to other complex regions of the genome.

Chapter 7

Bibliography

G. R. Abecasis, D. Altshuler, A. Auton, L. D. Brooks, R. M. Durbin, R. A. Gibbs, M. E. Hurles, G. A. McVean, D. Altshuler, R. M. Durbin, G. R. Abecasis, D. R. Bentley, A. Chakravarti, A. G. Clark, F. S. Collins, F. M. De La Vega, P. Donnelly, M. Egholm, P. Flicek, S. B. Gabriel, R. A. Gibbs, B. M. Knoppers, E. S. Lander, H. Lehrach, E. R. Mardis, G. A. McVean, D. A. Nickerson, L. Peltonen, A. J. Schafer, S. T. Sherry, J. Wang, R. Wilson, R. A. Gibbs, D. Deiros, M. Metzker, D. Muzny, J. Reid, D. Wheeler, J. Wang, J. Li, M. Jian, G. Li, R. Li, H. Liang, G. Tian, B. Wang, J. Wang, W. Wang, H. Yang, X. Zhang, H. Zheng, E. S. Lander, D. L. Altshuler, L. Ambrogio, T. Bloom, K. Cibulskis, T. J. Fennell, S. B. Gabriel, D. B. Jaffe, E. Shefler, C. L. Sougnez, D. R. Bentley, N. Gormley, S. Humphray, Z. Kingsbury, P. Kokko-Gonzales, J. Stone, K. J. McKernan, G. L. Costa, J. K. Ichikawa, C. C. Lee, R. Sudbrak, H. Lehrach, T. A. Borodina, A. Dahl, A. N. Davydov, P. Marquardt, F. Mertes, W. Nietfeld, P. Rosenstiel, S. Schreiber, A. V. Soldatov, B. Timmermann, M. Tolzmann, M. Egholm, J. Affourtit, D. Ashworth, S. Attiya, M. Bachorski, E. Buglione, A. Burke, A. Caprio, C. Celone, S. Clark, D. Conners, B. Desany, L. Gu, L. Guccione, K. Kao, J. Kebbler, J. Knowlton, M. Labrecque, L. McDade, C. Mealmaker, M. Minderman, A. Nawrocki, F. Niazi, K. Pareja, R. Ramenani, D. Riches, W. Song, C. Turcotte, S. Wang, E. R. Mardis, R. K. Wilson, D. Dooling, L. Fulton, R. Fulton, G. Weinstock, R. M. Durbin, J. Burton, D. M. Carter, C. Churcher, A. Coffey, A. Cox, A. Palotie, M. Quail, T. Skelly, J. Stalker, H. P. Swerdlow, D. Turner, A. De Witte, S. Giles, R. A. Gibbs, D. Wheeler, M. Bainbridge, D. Challis, A. Sabo, F. Yu, J. Yu, J. Wang, X. Fang, X. Guo, R. Li, Y. Li, R. Luo, S. Tai, H. Wu, H. Zheng, X. Zheng, Y. Zhou, G. Li, J. Wang, H. Yang, G. T. Marth, E. P. Garrison, W. Huang, A. Indap, D. Kural, W. P. Lee, W. F. Leong, A. R. Quinlan, C. Stewart, M. P. Stromberg, A. N. Ward, J. Wu, C. Lee, R. E. Mills, X. Shi, M. J. Daly, M. A. DePristo, D. L. Altshuler, A. D. Ball, E. Banks, T. Bloom, B. L. Browning, K. Cibulskis, T. J. Fennell, K. V. Garimella, S. R. Grossman, R. E. Handsaker, M. Hanna, C. Hartl, D. B. Jaffe, A. M. Kernytsky, J. M. Korn, H. Li, J. R. Maguire, S. A. McCarroll, A. McKenna, J. C. Nemesh, A. A. Philippakis, R. E. Poplin, A. Price, M. A. Rivas, P. C. Sabeti,

S. F. Schaffner, E. Shefler, I. A. Shlyakhter, D. N. Cooper, E. V. Ball, M. Mort, A. D. Phillips, P. D. Stenson, J. Sebat, V. Makarov, K. Ye, S. C. Yoon, C. D. Bustamante, A. G. Clark, A. Boyko, J. Degenhardt, S. Gravel, R. N. Gutenkunst, M. Kaganovich, A. Keinan, P. Lacroute, X. Ma, A. Reynolds, L. Clarke, P. Flicek, F. Cunningham, J. Herrero, S. Keenen, E. Kulesha, R. Leinonen, W. M. McLaren, R. Radhakrishnan, R. E. Smith, V. Zalunin, X. Zheng-Bradley, J. O. Korbel, A. M. Stutz, S. Humphray, M. Bauer, R. K. Cheetham, T. Cox, M. Eberle, T. James, S. Kahn, L. Murray, A. Chakravarti, K. Ye, F. M. De La Vega, Y. Fu, F. C. Hyland, J. M. Manning, S. F. McLaughlin, H. E. Peckham, O. Sakarya, Y. A. Sun, E. F. Tsung, M. A. Batzer, M. K. Konkel, J. A. Walker, R. Sudbrak, M. W. Albrecht, V. S. Amstislavskiy, R. Herwig, D. V. Parkhomchuk, S. T. Sherry, R. Agarwala, H. M. Khouri, A. O. Morgulis, J. E. Paschall, L. D. Phan, K. E. Rotmistrovsky, R. D. Sanders, M. F. Shumway, C. Xiao, G. A. McVean, A. Auton, Z. Iqbal, G. Lunter, J. L. Marchini, L. Moutsianas, S. Myers, A. Tumian, B. Desany, J. Knight, R. Winer, D. W. Craig, S. M. Beckstrom-Sternberg, A. Christoforides, A. A. Kurdoglu, J. V. Pearson, S. A. Sinari, W. D. Tembe, D. Haussler, A. S. Hinrichs, S. J. Katzman, A. Kern, R. M. Kuhn, M. Przeworski, R. D. Hernandez, B. Howie, J. L. Kelley, S. C. Melton, G. R. Abecasis, Y. Li, P. Anderson, T. Blackwell, W. Chen, W. O. Cookson, J. Ding, H. M. Kang, M. Lathrop, L. Liang, M. F. Moffatt, P. Scheet, C. Sidore, M. Snyder, X. Zhan, S. Zollner, P. Awadalla, F. Casals, Y. Idaghdour, J. Keebler, E. A. Stone, M. Zilversmit, L. Jorde, J. Xing, E. E. Eichler, G. Aksay, C. Alkan, I. Hajirasouliha, F. Hormozdiari, J. M. Kidd, S. C. Sahinalp, P. H. Sudmant, E. R. Mardis, K. Chen, A. Chinwalla, L. Ding, D. C. Koboldt, M. D. McLellan, D. Dooling, G. Weinstock, J. W. Wallis, M. C. Wendl, Q. Zhang, R. M. Durbin, C. A. Albers, Q. Ayub, S. Balasubramaniam, J. C. Barrett, D. M. Carter, Y. Chen, D. F. Conrad, P. Danecek, E. T. Dermitzakis, M. Hu, N. Huang, M. E. Hurles, H. Jin, L. Jostins, T. M. Keane, S. Q. Le, S. Lindsay, Q. Long, D. G. MacArthur, S. B. Montgomery, L. Parts, J. Stalker, C. Tyler-Smith, K. Walter, Y. Zhang, M. B. Gerstein, M. Snyder, A. Abyzov, S. Balasubramanian, R. Bjornson, J. Du, F. Grubert, L. Habegger, R. Haraksingh, J. Jee, E. Khurana, H. Y. Lam, J. Leng, X. J. Mu, A. E. Urban, Z. Zhang, Y. Li, R. Luo, G. T. Marth, E. P. Garrison, D. Kural, A. R. Quinlan, C. Stewart, M. P. Stromberg, A. N. Ward, J. Wu, C. Lee, R. E. Mills, X. Shi, S. A. McCarroll, E. Banks, M. A. DePristo, R. E. Handsaker, C. Hartl, J. M. Korn, H. Li, J. C. Nemesh, J. Sebat, V. Makarov, K. Ye, S. C. Yoon, J. Degenhardt, M. Kaganovich, L. Clarke, R. E. Smith, X. Zheng-Bradley, J. O. Korbel, S. Humphray, R. K. Cheetham, M. Eberle, S. Kahn, L. Murray, K. Ye, F. M. De La Vega, Y. Fu, H. E. Peckham, Y. A. Sun, M. A. Batzer, M. K. Konkel, J. A. Walker, C. Xiao, Z. Iqbal, B. Desany, T. Blackwell, M. Snyder, J. Xing, E. E. Eichler, G. Aksay, C. Alkan, I. Hajirasouliha, F. Hormozdiari, J. M. Kidd, K. Chen, A. Chinwalla, L. Ding, M. D. McLellan, J. W. Wallis, M. E. Hurles, D. F. Conrad, K. Walter, Y. Zhang, M. B. Gerstein, M. Snyder, A. Abyzov, J. Du, F. Grubert, R. Haraksingh, J. Jee, E. Khurana, H. Y. Lam,

J. Leng, X. J. Mu, A. E. Urban, Z. Zhang, R. A. Gibbs, M. Bainbridge, D. Challis, C. Coafra, H. Dinh, C. Kovar, S. Lee, D. Muzny, L. Nazareth, J. Reid, A. Sabo, F. Yu, J. Yu, G. T. Marth, E. P. Garrison, A. Indap, W. F. Leong, A. R. Quinlan, C. Stewart, A. N. Ward, J. Wu, K. Cibulskis, T. J. Fennell, S. B. Gabriel, K. V. Garimella, C. Hartl, E. Shefler, C. L. Sougnez, J. Wilkinson, A. G. Clark, S. Gravel, F. Grubert, L. Clarke, P. Flicek, R. E. Smith, X. Zheng-Bradley, S. T. Sherry, H. M. Khouri, J. E. Paschall, M. F. Shumway, C. Xiao, G. A. McVean, S. J. Katzman, G. R. Abecasis, E. R. Mardis, D. Dooling, L. Fulton, R. Fulton, D. C. Koboldt, R. M. Durbin, S. Balasubramaniam, A. Coffey, T. M. Keane, D. G. MacArthur, A. Palotie, C. Scott, J. Stalker, C. Tyler-Smith, M. B. Gerstein, S. Balasubramanian, A. Chakravarti, B. M. Knoppers, G. R. Abecasis, C. D. Bustamante, N. Gharani, R. A. Gibbs, L. Jorde, J. S. Kaye, A. Kent, T. Li, A. L. McGuire, G. A. McVean, P. N. Ossorio, C. N. Rotimi, Y. Su, L. H. Toji, C. Tyler-Smith, L. D. Brooks, A. L. Felsenfeld, J. E. McEwen, A. Abdallah, C. R. Juenger, N. C. Clemm, F. S. Collins, A. Duncanson, E. D. Green, M. S. Guyer, J. L. Peterson, A. J. Schafer, Y. Xue, and R. A. Cartwright. A map of human genome variation from population-scale sequencing. *Nature*, 467(7319): 1061–1073, Oct 2010a.

G. R. Abecasis, D. Altshuler, A. Auton, L. D. Brooks, R. M. Durbin, R. A. Gibbs, M. E. Hurles, G. A. McVean, D. Altshuler, R. M. Durbin, G. R. Abecasis, D. R. Bentley, A. Chakravarti, A. G. Clark, F. S. Collins, F. M. De La Vega, P. Donnelly, M. Egholm, P. Flicek, S. B. Gabriel, R. A. Gibbs, B. M. Knoppers, E. S. Lander, H. Lehrach, E. R. Mardis, G. A. McVean, D. A. Nickerson, L. Peltonen, A. J. Schafer, S. T. Sherry, J. Wang, R. Wilson, R. A. Gibbs, D. Deiros, M. Metzker, D. Muzny, J. Reid, D. Wheeler, J. Wang, J. Li, M. Jian, G. Li, R. Li, H. Liang, G. Tian, B. Wang, J. Wang, W. Wang, H. Yang, X. Zhang, H. Zheng, E. S. Lander, D. L. Altshuler, L. Ambrogio, T. Bloom, K. Cibulskis, T. J. Fennell, S. B. Gabriel, D. B. Jaffe, E. Shefler, C. L. Sougnez, D. R. Bentley, N. Gormley, S. Humphray, Z. Kingsbury, P. Kokko-Gonzales, J. Stone, K. J. McKernan, G. L. Costa, J. K. Ichikawa, C. C. Lee, R. Sudbrak, H. Lehrach, T. A. Borodina, A. Dahl, A. N. Davydov, P. Marquardt, F. Mertes, W. Nietfeld, P. Rosenstiel, S. Schreiber, A. V. Soldatov, B. Timmermann, M. Tolzmann, M. Egholm, J. Affourtit, D. Ashworth, S. Attiya, M. Bachorski, E. Buglione, A. Burke, A. Caprio, C. Celone, S. Clark, D. Conners, B. Desany, L. Gu, L. Guccione, K. Kao, J. Kebbler, J. Knowlton, M. Labrecque, L. McDade, C. Mealmaker, M. Minderman, A. Nawrocki, F. Niazi, K. Pareja, R. Ramenani, D. Riches, W. Song, C. Turcotte, S. Wang, E. R. Mardis, R. K. Wilson, D. Dooling, L. Fulton, R. Fulton, G. Weinstock, R. M. Durbin, J. Burton, D. M. Carter, C. Churcher, A. Coffey, A. Cox, A. Palotie, M. Quail, T. Skelly, J. Stalker, H. P. Swerdlow, D. Turner, A. De Witte, S. Giles, R. A. Gibbs, D. Wheeler, M. Bainbridge, D. Challis, A. Sabo, F. Yu, J. Yu, J. Wang, X. Fang, X. Guo, R. Li, Y. Li, R. Luo, S. Tai, H. Wu, H. Zheng, X. Zheng, Y. Zhou, G. Li, J. Wang, H. Yang, G. T. Marth, E. P. Garrison, W. Huang, A. Indap, D. Kural,

W. P. Lee, W. F. Leong, A. R. Quinlan, C. Stewart, M. P. Stromberg, A. N. Ward, J. Wu, C. Lee, R. E. Mills, X. Shi, M. J. Daly, M. A. DePristo, D. L. Altshuler, A. D. Ball, E. Banks, T. Bloom, B. L. Browning, K. Cibulskis, T. J. Fennell, K. V. Garimella, S. R. Grossman, R. E. Handsaker, M. Hanna, C. Hartl, D. B. Jaffe, A. M. Kernytsky, J. M. Korn, H. Li, J. R. Maguire, S. A. McCarroll, A. McKenna, J. C. Nemesh, A. A. Philippakis, R. E. Poplin, A. Price, M. A. Rivas, P. C. Sabeti, S. F. Schaffner, E. Shefler, I. A. Shlyakhter, D. N. Cooper, E. V. Ball, M. Mort, A. D. Phillips, P. D. Stenson, J. Sebat, V. Makarov, K. Ye, S. C. Yoon, C. D. Bustamante, A. G. Clark, A. Boyko, J. Degenhardt, S. Gravel, R. N. Gutenkunst, M. Kaganovich, A. Keinan, P. Lacroute, X. Ma, A. Reynolds, L. Clarke, P. Flicek, F. Cunningham, J. Herrero, S. Keenen, E. Kulesha, R. Leinonen, W. M. McLaren, R. Radhakrishnan, R. E. Smith, V. Zalunin, X. Zheng-Bradley, J. O. Korbel, A. M. Stutz, S. Humphray, M. Bauer, R. K. Cheetham, T. Cox, M. Eberle, T. James, S. Kahn, L. Murray, A. Chakravarti, K. Ye, F. M. De La Vega, Y. Fu, F. C. Hyland, J. M. Manning, S. F. McLaughlin, H. E. Peckham, O. Sakarya, Y. A. Sun, E. F. Tsung, M. A. Batzer, M. K. Konkel, J. A. Walker, R. Sudbrak, M. W. Albrecht, V. S. Amstislavskiy, R. Herwig, D. V. Parkhomchuk, S. T. Sherry, R. Agarwala, H. M. Khouri, A. O. Morgulis, J. E. Paschall, L. D. Phan, K. E. Rotmistrovsky, R. D. Sanders, M. F. Shumway, C. Xiao, G. A. McVean, A. Auton, Z. Iqbal, G. Lunter, J. L. Marchini, L. Moutsianas, S. Myers, A. Tumian, B. Desany, J. Knight, R. Winer, D. W. Craig, S. M. Beckstrom-Sternberg, A. Christoforides, A. A. Kurdoglu, J. V. Pearson, S. A. Sinari, W. D. Tembe, D. Haussler, A. S. Hinrichs, S. J. Katzman, A. Kern, R. M. Kuhn, M. Przeworski, R. D. Hernandez, B. Howie, J. L. Kelley, S. C. Melton, G. R. Abecasis, Y. Li, P. Anderson, T. Blackwell, W. Chen, W. O. Cookson, J. Ding, H. M. Kang, M. Lathrop, L. Liang, M. F. Moffatt, P. Scheet, C. Sidore, M. Snyder, X. Zhan, S. Zollner, P. Awadalla, F. Casals, Y. Idaghdour, J. Keebler, E. A. Stone, M. Zilversmit, L. Jorde, J. Xing, E. E. Eichler, G. Aksay, C. Alkan, I. Hajirasouliha, F. Hormozdiari, J. M. Kidd, S. C. Sahinalp, P. H. Sudmant, E. R. Mardis, K. Chen, A. Chinwalla, L. Ding, D. C. Koboldt, M. D. McLellan, D. Dooling, G. Weinstock, J. W. Wallis, M. C. Wendl, Q. Zhang, R. M. Durbin, C. A. Albers, Q. Ayub, S. Balasubramaniam, J. C. Barrett, D. M. Carter, Y. Chen, D. F. Conrad, P. Danecek, E. T. Dermitzakis, M. Hu, N. Huang, M. E. Hurles, H. Jin, L. Jostins, T. M. Keane, S. Q. Le, S. Lindsay, Q. Long, D. G. MacArthur, S. B. Montgomery, L. Parts, J. Stalker, C. Tyler-Smith, K. Walter, Y. Zhang, M. B. Gerstein, M. Snyder, A. Abyzov, S. Balasubramanian, R. Bjornson, J. Du, F. Grubert, L. Habegger, R. Haraksingh, J. Jee, E. Khurana, H. Y. Lam, J. Leng, X. J. Mu, A. E. Urban, Z. Zhang, Y. Li, R. Luo, G. T. Marth, E. P. Garrison, D. Kural, A. R. Quinlan, C. Stewart, M. P. Stromberg, A. N. Ward, J. Wu, C. Lee, R. E. Mills, X. Shi, S. A. McCarroll, E. Banks, M. A. DePristo, R. E. Handsaker, C. Hartl, J. M. Korn, H. Li, J. C. Nemesh, J. Sebat, V. Makarov, K. Ye, S. C. Yoon, J. Degenhardt, M. Kaganovich, L. Clarke, R. E. Smith, X. Zheng-Bradley, J. O. Korbel, S. Humphray, R. K. Cheetham, M. Eberle, S. Kahn, L. Murray,

K. Ye, F. M. De La Vega, Y. Fu, H. E. Peckham, Y. A. Sun, M. A. Batzer, M. K. Konkel, J. A. Walker, C. Xiao, Z. Iqbal, B. Desany, T. Blackwell, M. Snyder, J. Xing, E. E. Eichler, G. Aksay, C. Alkan, I. Hajirasouliha, F. Hormozdiari, J. M. Kidd, K. Chen, A. Chinwalla, L. Ding, M. D. McLellan, J. W. Wallis, M. E. Hurles, D. F. Conrad, K. Walter, Y. Zhang, M. B. Gerstein, M. Snyder, A. Abyzov, J. Du, F. Grubert, R. Haraksingh, J. Jee, E. Khurana, H. Y. Lam, J. Leng, X. J. Mu, A. E. Urban, Z. Zhang, R. A. Gibbs, M. Bainbridge, D. Challis, C. Coafra, H. Dinh, C. Kovar, S. Lee, D. Muzny, L. Nazareth, J. Reid, A. Sabo, F. Yu, J. Yu, G. T. Marth, E. P. Garrison, A. Indap, W. F. Leong, A. R. Quinlan, C. Stewart, A. N. Ward, J. Wu, K. Cibulskis, T. J. Fennell, S. B. Gabriel, K. V. Garimella, C. Hartl, E. Shefler, C. L. Sougnez, J. Wilkinson, A. G. Clark, S. Gravel, F. Grubert, L. Clarke, P. Flicek, R. E. Smith, X. Zheng-Bradley, S. T. Sherry, H. M. Khouri, J. E. Paschall, M. F. Shumway, C. Xiao, G. A. McVean, S. J. Katzman, G. R. Abecasis, E. R. Mardis, D. Dooling, L. Fulton, R. Fulton, D. C. Koboldt, R. M. Durbin, S. Balasubramaniam, A. Coffey, T. M. Keane, D. G. MacArthur, A. Palotie, C. Scott, J. Stalker, C. Tyler-Smith, M. B. Gerstein, S. Balasubramanian, A. Chakravarti, B. M. Knoppers, G. R. Abecasis, C. D. Bustamante, N. Gharani, R. A. Gibbs, L. Jorde, J. S. Kaye, A. Kent, T. Li, A. L. McGuire, G. A. McVean, P. N. Ossorio, C. N. Rotimi, Y. Su, L. H. Toji, C. Tyler-Smith, L. D. Brooks, A. L. Felsenfeld, J. E. McEwen, A. Abdallah, C. R. Juenger, N. C. Clemm, F. S. Collins, A. Duncanson, E. D. Green, M. S. Guyer, J. L. Peterson, A. J. Schafer, Y. Xue, and R. A. Cartwright. A map of human genome variation from population-scale sequencing. *Nature*, 467(7319): 1061–1073, Oct 2010b.

Alfred V. Aho and Margaret J. Corasick. Efficient string matching: an aid to bibliographic search. *Commun. ACM*, 18(6):333–340, June 1975. ISSN 0001-0782. doi: 10.1145/360825.360855. URL http://doi.acm.org/10.1145/360825.360855.

D. Altshuler, L. D. Brooks, A. Chakravarti, F. S. Collins, MJ Daly, P Donnelly, and International HapMap Consortium. A haplotype map of the human genome. *Nature*, 437(7063):1299–1320, Oct 2005.

F. Aversa, A. Tabilio, A. Velardi, I. Cunningham, A. Terenzi, F. Falzetti, L. Ruggeri, G. Barbabietola, C. Aristei, P. Latini, Y. Reisner, and M. F. Martelli. Treatment of high-risk acute leukemia with T-cell-depleted stem cells from related donors with one fully mismatched HLA haplotype. *N. Engl. J. Med.*, 339:1186–1193, Oct 1998.

V. Bafna, Istrail S., G. Lancia, and R. Rizzi. Polynomial and APX-hard cases of Individual Haplotyping Problems. *Theoretical Computer Science*, 335(1):109–125, 2005.

V. Bansal and V. Bafna. HapCUT: an efficient and accurate algorithm for the haplotype assembly problem. *Bioinformatics*, 24:i153–159, Aug 2008.

V. Bansal, A. L. Halpern, N. Axelrod, and V. Bafna. An MCMC algorithm for haplotype assembly from whole-genome sequence data. *Genome Res*, 18(8): 1336–1346, Aug 2008. doi: 10.1101/gr.077065.108.

beta. Beta distribution. http://en.wikipedia.org/wiki/Beta_distribution, Aug 2010. Wikipedia.

B. L. Browning and S. R. Browning. A unified approach to genotype imputation and haplotype-phase inference for large data sets of trios and unrelated individuals. *Am. J. Hum. Genet.*, 84(2):210–223, Feb 2009.

C. Burgtorf, P. Kepper, M. Hoehe, C. Schmitt, R. Reinhardt, H. Lehrach, and S. Sauer. Clone-based systematic haplotyping (CSH): a procedure for physical haplotyping of whole genomes. *Genome Res.*, 13(12):2717–2724, Dec 2003.

M. J. Chaisson, D. Brinza, and P. A. Pevzner. De novo fragment assembly with short mate-paired reads: Does the read length matter? *Genome Res.*, 19(2): 336–346, Feb 2009.

G. M. Church. The personal genome project. *Mol. Syst. Biol.*, 1:2005.0030, 2005.

S.A. Cook. The complexity of theorem-proving procedures. In *Proceedings of the third annual ACM symposium on Theory of computing*, pages 151–158. ACM, 1971.

K. Davies. Splash down: Pacific biosciences unveils third generation sequencing machine. *Bio-IT World*, 2010.

V. De Re, L. Caggiari, M. De Zorzi, and G. Toffoli. KIR molecules: recent patents of interest for the diagnosis and treatment of several autoimmune diseases, chronic inflammation, and B-cell malignancies. *Recent Pat DNA Gene Seq*, 5(3):169–174, Dec 2011.

R. Drmanac, A. B. Sparks, M. J. Callow, A. L. Halpern, N. L. Burns, B. G. Kermani, P. Carnevali, I. Nazarenko, G. B. Nilsen, G. Yeung, F. Dahl, A. Fernandez, B. Staker, K. P. Pant, J. Baccash, A. P. Borcherding, A. Brownley, R. Cedeno, L. Chen, D. Chernikoff, A. Cheung, R. Chirita, B. Curson, J. C. Ebert, C. R. Hacker, R. Hartlage, B. Hauser, S. Huang, Y. Jiang, V. Karpinchyk, M. Koenig, C. Kong, T. Landers, C. Le, J. Liu, C. E. McBride, M. Morenzoni, R. E. Morey, K. Mutch, H. Perazich, K. Perry, B. A. Peters, J. Peterson, C. L. Pethiyagoda, K. Pothuraju, C. Richter, A. M. Rosenbaum, S. Roy, J. Shafto, U. Sharanhovich, K. W. Shannon, C. G. Sheppy, M. Sun, J. V. Thakuria, A. Tran, D. Vu, A. W. Zaranek, X. Wu, S. Drmanac, A. R. Oliphant, W. C. Banyai, B. Martin, D. G.

Ballinger, G. M. Church, and C. A. Reid. Human genome sequencing using unchained base reads on self-assembling DNA nanoarrays. *Science*, 327(5961): 78–81, Jan 2010.

J. Edmonds and R.M. Karp. Theoretical improvements in algorithmic efficiency for network flow problems. *Journal of the ACM (JACM)*, 19(2):248–264, 1972.

J. Eid, A. Fehr, J. Gray, K. Luong, J. Lyle, G. Otto, P. Peluso, D. Rank, P. Baybayan, B. Bettman, A. Bibillo, K. Bjornson, B. Chaudhuri, F. Christians, R. Cicero, S. Clark, R. Dalal, A. Dewinter, J. Dixon, M. Foquet, A. Gaertner, P. Hardenbol, C. Heiner, K. Hester, D. Holden, G. Kearns, X. Kong, R. Kuse, Y. Lacroix, S. Lin, P. Lundquist, C. Ma, P. Marks, M. Maxham, D. Murphy, I. Park, T. Pham, M. Phillips, J. Roy, R. Sebra, G. Shen, J. Sorenson, A. Tomaney, K. Travers, M. Trulson, J. Vieceli, J. Wegener, D. Wu, A. Yang, D. Zaccarin, P. Zhao, F. Zhong, J. Korlach, and S. Turner. Real-time dna sequencing from single polymerase molecules. *Science*, 323(5910):133–138, Jan 2009. doi: 10.1126/science.1162986. URL http://www.hubmed.org/display.cgi?uids=19023044.

P. Erdos and A. Renyi. On random graphs. *Publ. Math. Debrecen*, 6(290-297):156, 1959.

H. C. Fan, J. Wang, A. Potanina, and S. R. Quake. Whole-genome molecular haplotyping of single cells. *Nat. Biotechnol.*, 29(1):51–57, Jan 2011.

K. A. Frazer, D. G. Ballinger, D. R. Cox, D. A. Hinds, L. L. Stuve, R. A. Gibbs, J. W. Belmont, A. Boudreau, P. Hardenbol, S. M. Leal, S. Pasternak, D. A. Wheeler, T. D. Willis, F. Yu, H. Yang, C. Zeng, Y. Gao, H. Hu, W. Hu, C. Li, W. Lin, S. Liu, H. Pan, X. Tang, J. Wang, W. Wang, J. Yu, B. Zhang, Q. Zhang, H. Zhao, H. Zhao, J. Zhou, S. B. Gabriel, R. Barry, B. Blumenstiel, A. Camargo, M. Defelice, M. Faggart, M. Goyette, S. Gupta, J. Moore, H. Nguyen, R. C. Onofrio, M. Parkin, J. Roy, E. Stahl, E. Winchester, L. Ziaugra, D. Altshuler, Y. Shen, Z. Yao, W. Huang, X. Chu, Y. He, L. Jin, Y. Liu, Y. Shen, W. Sun, H. Wang, Y. Wang, Y. Wang, X. Xiong, L. Xu, M. M. Waye, S. K. Tsui, H. Xue, J. T. Wong, L. M. Galver, J. B. Fan, K. Gunderson, S. S. Murray, A. R. Oliphant, M. S. Chee, A. Montpetit, F. Chagnon, V. Ferretti, M. Leboeuf, J. F. Olivier, M. S. Phillips, S. Roumy, C. Sallee, A. Verner, T. J. Hudson, P. Y. Kwok, D. Cai, D. C. Koboldt, R. D. Miller, L. Pawlikowska, P. Taillon-Miller, M. Xiao, L. C. Tsui, W. Mak, Y. Q. Song, P. K. Tam, Y. Nakamura, T. Kawaguchi, T. Kitamoto, T. Morizono, A. Nagashima, Y. Ohnishi, A. Sekine, T. Tanaka, T. Tsunoda, P. Deloukas, C. P. Bird, M. Delgado, E. T. Dermitzakis, R. Gwilliam, S. Hunt, J. Morrison, D. Powell, B. E. Stranger, P. Whittaker, D. R. Bentley, M. J. Daly, P. I. de Bakker, J. Barrett, Y. R. Chretien, J. Maller, S. McCarroll, N. Patterson, I. Pe'er, A. Price, S. Purcell, D. J. Richter, P. Sabeti, R. Saxena, S. F. Schaffner, P. C. Sham, P. Varilly, D. Altshuler, L. D. Stein, L. Krishnan, A. V. Smith,

M. K. Tello-Ruiz, G. A. Thorisson, A. Chakravarti, P. E. Chen, D. J. Cutler, C. S. Kashuk, S. Lin, G. R. Abecasis, W. Guan, Y. Li, H. M. Munro, Z. S. Qin, D. J. Thomas, G. McVean, A. Auton, L. Bottolo, N. Cardin, S. Eyheramendy, C. Freeman, J. Marchini, S. Myers, C. Spencer, M. Stephens, P. Donnelly, L. R. Cardon, G. Clarke, D. M. Evans, A. P. Morris, B. S. Weir, T. Tsunoda, J. C. Mullikin, S. T. Sherry, M. Feolo, A. Skol, H. Zhang, C. Zeng, H. Zhao, I. Matsuda, Y. Fukushima, D. R. Macer, E. Suda, C. N. Rotimi, C. A. Adebamowo, I. Ajayi, T. Aniagwu, P. A. Marshall, C. Nkwodimmah, C. D. Royal, M. F. Leppert, M. Dixon, A. Peiffer, R. Qiu, A. Kent, K. Kato, N. Niikawa, I. F. Adewole, B. M. Knoppers, M. W. Foster, E. W. Clayton, J. Watkin, R. A. Gibbs, J. W. Belmont, D. Muzny, L. Nazareth, E. Sodergren, G. M. Weinstock, D. A. Wheeler, I. Yakub, S. B. Gabriel, R. C. Onofrio, D. J. Richter, L. Ziaugra, B. W. Birren, M. J. Daly, D. Altshuler, R. K. Wilson, L. L. Fulton, J. Rogers, J. Burton, N. P. Carter, C. M. Clee, M. Griffiths, M. C. Jones, K. McLay, R. W. Plumb, M. T. Ross, S. K. Sims, D. L. Willey, Z. Chen, H. Han, L. Kang, M. Godbout, J. C. Wallenburg, P. L'Archeveque, G. Bellemare, K. Saeki, H. Wang, D. An, H. Fu, Q. Li, Z. Wang, R. Wang, A. L. Holden, L. D. Brooks, J. E. McEwen, M. S. Guyer, V. O. Wang, J. L. Peterson, M. Shi, J. Spiegel, L. M. Sung, L. F. Zacharia, F. S. Collins, K. Kennedy, R. Jamieson, and J. Stewart. A second generation human haplotype map of over 3.1 million SNPs. *Nature*, 449(7164):851–861, Oct 2007.

M.R. Garey and D.S. Johnson. *Computers and intractability*, volume 174. Freeman San Francisco, CA, 1979.

K. Gendzekhadze, P. J. Norman, L. Abi-Rached, Z. Layrisse, and P. Parham. High KIR diversity in Amerindians is maintained using few gene-content haplotypes. *Immunogenetics*, 58(5-6):474–480, Jun 2006.

K. Gendzekhadze, P. J. Norman, L. Abi-Rached, T. Graef, A. K. Moesta, Z. Layrisse, and P. Parham. Co-evolution of KIR2DL3 with HLA-C in a human population retaining minimal essential diversity of KIR and HLA class I ligands. *Proc. Natl. Acad. Sci. U.S.A.*, 106(44):18692–18697, Nov 2009.

Z. Guo, L. Hood, M. Malkki, and E. W. Petersdorf. Long-range multilocus haplotype phasing of the MHC. *Proc. Natl. Acad. Sci. U.S.A.*, 103:6964–6969, May 2006.

I. Hajirasouliha, F. Hormozdiari, C. Alkan, J.M. Kidd, I. Birol, E.E. Eichler, and S.C. Sahinalp. Detection and characterization of novel sequence insertions using paired-end next-generation sequencing. *Bioinformatics*, 26(10):1277–1283, 2010.

Y. Hall. Coming Soon: Your Personal DNA Map? National Geographic News. http://news.nationalgeographic.com/news/2006/03/0307_060307_dna.html, 2011.

B. V. Halldórsson, V. Bafna, N. Edwards, R. Lippert, Yooseph. S., and S. Istrail. Combinatorial Problems arising in SNP and Haplotype Analysis. In *DMTCS*, pages 26–47, 2003.

P. Havlak, R. Chen, K. J. Durbin, A. Egan, Y. Ren, X. Z. Song, G. M. Weinstock, and R. A. Gibbs. The Atlas genome assembly system. *Genome Res.*, 14(4): 721–732, Apr 2004.

D. He, A. Choi, K. Pipatsrisawat, A. Darwiche, and E. Eskin. Optimal algorithms for haplotype assembly from whole-genome sequence data. *Bioinformatics*, 26: i183–190, Jun 2010.

A. Helgason, G. Nicholson, K. Stefansson, and P. Donnelly. A reassessment of genetic diversity in Icelanders: strong evidence from multiple loci for relative homogeneity caused by genetic drift. *Ann. Hum. Genet.*, 67(Pt 4):281–297, Jul 2003.

F. Hormozdiari, I. Hajirasouliha, A. McPherson, E.E. Eichler, and S.C. Sahinalp. Simultaneous structural variation discovery among multiple paired-end sequenced genomes. *Genome research*, 21(12):2203–2212, 2011.

K. C. Hsu, S. Chida, D. E. Geraghty, and B. Dupont. The killer cell immunoglobulin-like receptor (KIR) genomic region: gene-order, haplotypes and allelic polymorphism. *Immunol. Rev.*, 190:40–52, Dec 2002.

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

F. Kaper, S. Swamy, B. Klotzle, S. Munchel, J. Cottrell, M. Bibikova, H. Y. Chuang, S. Kruglyak, M. Ronaghi, M. A. Eberle, and J. B. Fan. Whole-genome haplotyping by dilution, amplification, and sequencing. *Proc. Natl. Acad. Sci. U.S.A.*, 110(14):5552–5557, Apr 2013.

J. M. Kidd et al. Mapping and sequencing of structural variation from eight human genomes. *Nature*, 453:56–64, May 2008.

S. Kim, K. Jeong, K. Bhutani, J. H. Lee, A. Patel, E. Scott, H. Nam, H. Lee, J. G. Gleeson, and V. Bafna. Virmid: accurate detection of somatic mutations with sample impurity inference. *Genome Biol.*, 14(8):R90, Aug 2013.

E. F. Kirkness, R. V. Grindberg, J. Yee-Greenbaum, C. R. Marshall, S. W. Scherer, R. S. Lasken, and J. C. Venter. Sequencing of isolated sperm cells for direct haplotyping of a human genome. *Genome Res.*, 23(5):826–832, May 2013.

J. O. Kitzman, A. P. Mackenzie, A. Adey, J. B. Hiatt, R. P. Patwardhan, P. H. Sudmant, S. B. Ng, C. Alkan, R. Qiu, E. E. Eichler, and J. Shendure. Haplotype-resolved genome sequencing of a Gujarati Indian individual. *Nat. Biotechnol.*, 29 (1):59–63, Jan 2011.

Y. Kodama, M. Shumway, and R. Leinonen. The Sequence Read Archive: explosive growth of sequencing data. *Nucleic Acids Res.*, 40(Database issue):D54–56, Jan 2012.

J. Krumsiek, R. Arnold, and T. Rattei. Gepard: a rapid and sensitive tool for creating dotplots on genome scale. *Bioinformatics*, 23(8):1026–1028, Apr 2007.

E. S. Lander and M. S. Waterman. Genomic mapping by fingerprinting random clones: a mathematical analysis. *Genomics*, 2(3):231–239, Apr 1988.

B. Langmead and S.L. Salzberg. Fast gapped-read alignment with Bowtie 2. *Nature methods*, 9(4):357–359, 2012.

J. Lee, B. Kim, J. Yoon, and U. Lee. Detection of copy number variation using scale space filtering. *Conf Proc IEEE Eng Med Biol Soc*, 2011:5555–5558, 2011.

S. Levy, G. Sutton, P. C. Ng, L. Feuk, A. L. Halpern, B. P. Walenz, N. Axelrod, J. Huang, E. F. Kirkness, G. Denisov, Y. Lin, J. R. MacDonald, A. W. Pang, M. Shago, T. B. Stockwell, A. Tsiamouri, V. Bafna, V. Bansal, S. A. Kravitz, D. A. Busam, K. Y. Beeson, T. C. McIntosh, K. A. Remington, J. F. Abril, J. Gill, J. Borman, Y. H. Rogers, M. E. Frazier, S. W. Scherer, R. L. Strausberg, and J. C. " Venter. The diploid genome sequence of an individual human. *PLoS biology*, 5(10):e254, 2007.

H. Li and R. Durbin. Fast and accurate long-read alignment with Burrows-Wheeler transform. *Bioinformatics*, 26(5):589–595, Mar 2010.

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

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

R. Li, H. Zhu, J. Ruan, W. Qian, X. Fang, Z. Shi, Y. Li, S. Li, G. Shan, K. Kristiansen, S. Li, H. Yang, J. Wang, and J. Wang. De novo assembly of human genomes with massively parallel short read sequencing. *Genome Research*, 20(2): 265–272, February 2010.

C. Lo, R. Liu, J. Lee, K. Robasky, S. Byrne, C. Lucchesi, J. Aach, G. Church, V. Bafna, and K. Zhang. On the design of clone-based haplotyping. *Genome Biol.*, 14(9):R100, 2013a.

Christine Lo, Sangwoo Kim, Shay Zakov, and Vineet Bafna. Evaluating genome architecture of a complex region via generalized bipartite matching. *BMC Bioinformatics*, 14(Suppl 5):S13, 2013b. ISSN 1471-2105. doi: 10.1186/1471-2105-14-S5-S13. URL http://www.biomedcentral.com/1471-2105/14/S5/S13.

L. Lovász and M.D. Plummer. *Matching Theory*, volume 29 of *Annals of Discrete Mathematics*. North-Holland, Amsterdam, 1986.

M. Lynch. Rate, molecular spectrum, and consequences of human mutation. *Proc. Natl. Acad. Sci. U.S.A.*, 107(3):961–968, Jan 2010.

L. Ma, Y. Xiao, H. Huang, Q. Wang, W. Rao, Y. Feng, K. Zhang, and Q. Song. Direct determination of molecular haplotypes by chromosome microdissection. *Nat. Methods*, 7:299–301, Apr 2010.

G. M. Maniatis. Erythropoiesis: a model for differentiation. *Prog. Clin. Biol. Res.*, 102 pt A:13–24, 1982.

J. Marchini, D. Cutler, N. Patterson, M. Stephens, E. Eskin, E. Halperin, S. Lin, Z. S. Qin, H. M. Munro, G. R. Abecasis, and P. Donnelly. A comparison of phasing algorithms for trios and unrelated individuals. *Am. J. Hum. Genet.*, 78: 437–450, Mar 2006.

E. R. Mardis. Next-generation DNA sequencing methods. *Annu Rev Genomics Hum Genet*, 9:387–402, 2008.

S. G. Marsh, P. Parham, B. Dupont, D. E. Geraghty, J. Trowsdale, D. Middleton, C. Vilches, M. Carrington, C. Witt, L. A. Guethlein, H. Shilling, C. A. Garcia, K. C. Hsu, and H. Wain. Killer-cell immunoglobulin-like receptor (KIR) nomenclature report, 2002. *Tissue Antigens*, 62(1):79–86, Jul 2003.

K. E. McElroy, F. Luciani, and T. Thomas. GemSIM: general, error-model based simulator of next-generation sequencing data. *BMC Genomics*, 13:74, 2012a.

K. E. McElroy, F. Luciani, and T. Thomas. GemSIM: general, error-model based simulator of next-generation sequencing data. *BMC Genomics*, 13:74, 2012b.

A. McKenna, M. Hanna, E. Banks, A. Sivachenko, K. Cibulskis, A. Kernytsky, K. Garimella, D. Altshuler, S. Gabriel, M. Daly, and M. A. DePristo. The Genome Analysis Toolkit: a MapReduce framework for analyzing next-generation DNA sequencing data. *Genome Res.*, 20(9):1297–1303, Sep 2010a.

A. McKenna, M. Hanna, E. Banks, A. Sivachenko, K. Cibulskis, A. Kernytsky, K. Garimella, D. Altshuler, S. Gabriel, M. Daly, and M. A. DePristo. The Genome Analysis Toolkit: a MapReduce framework for analyzing next-generation DNA sequencing data. *Genome Res.*, 20(9):1297–1303, Sep 2010b.

P. Medvedev, M. Fiume, M. Dzamba, T. Smith, and M. Brudno. Detecting copy number variation with mated short reads. *Genome research*, 20(11):1613–1622, 2010.

D. Middleton and F. Gonzelez. The extensive polymorphism of KIR genes. *Immunology*, 129(1):8–19, Jan 2010.

R. E. Mills et al. Mapping copy number variation by population-scale genome sequencing. *Nature*, 470:59–65, Feb 2011.

P. J. Norman, J. A. Hollenbach, N. Nemat-Gorgani, L. A. Guethlein, H. G. Hilton, M. J. Pando, K. A. Koram, E. M. Riley, L. Abi-Rached, and P. Parham. Coevolution of human leukocyte antigen (HLA) class I ligands with killer-cell immunoglobulin-like receptors (KIR) in a genetically diverse population of sub-Saharan Africans. *PLoS Genet.*, 9(10):e1003938, Oct 2013.

P. Parham. Immunogenetics of killer-cell immunoglobulin-like receptors. *Tissue Antigens*, 62(3):194–200, Sep 2003.

K. Pelak, K. V. Shianna, D. Ge, J. M. Maia, M. Zhu, J. P. Smith, E. T. Cirulli, J. Fellay, S. P. Dickson, C. E. Gumbs, E. L. Heinzen, A. C. Need, E. K. Ruzzo, A. Singh, C. R. Campbell, L. K. Hong, K. A. Lornsen, A. M. McKenzie, N. L. Sobreira, J. E. Hoover-Fong, J. D. Milner, R. Ottman, B. F. Haynes, J. J. Goedert, and D. B. Goldstein. The characterization of twenty sequenced human genomes. *PLoS Genet.*, 6(9):e1001111, Sep 2010.

B. A. Peters, B. G. Kermani, A. B. Sparks, O. Alferov, P. Hong, A. Alexeev, Y. Jiang, F. Dahl, Y. T. Tang, J. Haas, K. Robasky, A. W. Zaranek, J. H. Lee, M. P. Ball, J. E. Peterson, H. Perazich, G. Yeung, J. Liu, L. Chen, M. I. Kennemer, K. Pothuraju, K. Konvicka, M. Tsoupko-Sitnikov, K. P. Pant, J. C. Ebert, G. B. Nilsen, J. Baccash, A. L. Halpern, G. M. Church, and R. Drmanac. Accurate whole-genome sequencing and haplotyping from 10 to 20 human cells. *Nature*, 487(7406):190–195, Jul 2012.

P. A. Pevzner, H. Tang, and M. S. Waterman. An eulerian path approach to dna fragment assembly. *Proc Natl Acad Sci U S A*, 98(17):9748–9753, August 2001.

N. Pontikos, D. J. Smyth, H. Schuilenburg, J. M. Howson, N. M. Walker, O. S.

Burren, H. Guo, S. Onengut-Gumuscu, W. M. Chen, P. Concannon, S. S. Rich, J. Jayaraman, W. Jiang, J. A. Traherne, J. Trowsdale, J. A. Todd, and C. Wallace. A hybrid qPCR/SNP array approach allows cost efficient assessment of KIR gene copy numbers in large samples. *BMC Genomics*, 15:274, 2014.

C. W. Pyo, L. A. Guethlein, Q. Vu, R. Wang, L. Abi-Rached, P. J. Norman, S. G. Marsh, J. S. Miller, P. Parham, and D. E. Geraghty. Different patterns of evolution in the centromeric and telomeric regions of group A and B haplotypes of the human killer cell Ig-like receptor locus. *PLoS ONE*, 5(12):e15115, 2010.

S. Rajagopalan and E. O. Long. Understanding how combinations of HLA and KIR genes influence disease. *J. Exp. Med.*, 201(7):1025–1029, Apr 2005.

D. E. Reich, M. Cargill, S. Bolk, J. Ireland, P. C. Sabeti, D. J. Richter, T. Lavery, R. Kouyoumjian, S. F. Farhadian, R. Ward, et al. Linkage disequilibrium in the human genome. *Nature*, 411(6834):199–204, 2001.

A. Ritz, A. Bashir, and B. J. Raphael. Structural variation analysis with strobe reads. *Bioinformatics*, 26:1291–1298, May 2010.

J. C. Roach, G. Glusman, A. F. Smit, C. D. Huff, R. Hubley, P. T. Shannon, L. Rowen, K. P. Pant, N. Goodman, M. Bamshad, J. Shendure, R. Drmanac, L. B. Jorde, L. Hood, and D. J. Galas. Analysis of genetic inheritance in a family quartet by whole-genome sequencing. *Science*, 328:636–639, Apr 2010.

J. Robinson, M. J. Waller, P. Stoehr, and S. G. Marsh. IPD–the Immuno Polymorphism Database. *Nucleic Acids Res.*, 33(Database issue):D523–526, Jan 2005.

J. Robinson, J. A. Halliwell, H. McWilliam, R. Lopez, and S. G. Marsh. IPD–the Immuno Polymorphism Database. *Nucleic Acids Res.*, 41(Database issue): D1234–1240, Jan 2013.

S. L. Salzberg, A. M. Phillippy, A. Zimin, D. Puiu, T. Magoc, S. Koren, T. J. Treangen, M. C. Schatz, A. L. Delcher, M. Roberts, G. Marcais, M. Pop, and J. A. Yorke. GAGE: A critical evaluation of genome assemblies and assembly algorithms. *Genome Res.*, 22(3):557–567, Mar 2012.

E. E. Schadt, S. Turner, and A. Kasarskis. A window into third-generation sequencing. *Hum. Mol. Genet.*, 19(R2):R227–240, Oct 2010.

J. Shendure and H. Ji. Next-generation DNA sequencing. *Nat. Biotechnol.*, 26: 1135–1145, Oct 2008.

S. T. Sherry, M. H. Ward, M. Kholodov, J. Baker, L. Phan, E. M. Smigielski, and

K. Sirotkin. dbSNP: the NCBI database of genetic variation. *Nucleic Acids Res.*, 29(1):308–311, Jan 2001.

T. Shiina, H. Inoko, and J. K. Kulski. An update of the HLA genomic region, locus information and disease associations: 2004. *Tissue Antigens*, 64:631–649, Dec 2004.

H. Shizuya, B. Birren, U. J. Kim, V. Mancino, T. Slepak, Y. Tachiiri, and M. Simon. Cloning and stable maintenance of 300-kilobase-pair fragments of human DNA in Escherichia coli using an F-factor-based vector. *Proc. Natl. Acad. Sci. U.S.A.*, 89(18):8794–8797, Sep 1992.

E. K. Suk, G. K. McEwen, J. Duitama, K. Nowick, S. Schulz, S. Palczewski, S. Schreiber, D. T. Holloway, S. McLaughlin, H. Peckham, C. Lee, T. Huebsch, and M. R. Hoehe. A comprehensively molecular haplotype-resolved genome of a European individual. *Genome Res.*, 21(10):1672–1685, Oct 2011.

J. van Oeveren, M. de Ruiter, T. Jesse, H. van der Poel, J. Tang, F. Yalcin, A. Janssen, H. Volpin, K. E. Stormo, R. Bogden, M. J. van Eijk, and M. Prins. Sequence-based physical mapping of complex genomes by whole genome profiling. *Genome Res.*, 21(4):618–625, Apr 2011.

A. Voskoboynik, N. F. Neff, D. Sahoo, A. M. Newman, D. Pushkarev, W. Koh, B. Passarelli, H. C. Fan, G. L. Mantalas, K. J. Palmeri, K. J. Ishizuka, C. Gissi, F. Griggio, R. Ben-Shlomo, D. M. Corey, L. Penland, R. A. White, I. L. Weissman, and S. R. Quake. The genome sequence of the colonial chordate, Botryllus schlosseri. *Elife*, 2:e00569, 2013.

D. A. Wheeler, M. Srinivasan, M. Egholm, Y. Shen, L. Chen, A. McGuire, W. He, Y. J. Chen, V. Makhijani, G. T. Roth, X. Gomes, K. Tartaro, F. Niazi, C. L. Turcotte, G. P. Irzyk, J. R. Lupski, C. Chinault, X. Z. Song, Y. Liu, Y. Yuan, L. Nazareth, X. Qin, D. M. Muzny, M. Margulies, G. M. Weinstock, R. A. Gibbs, and J. M. Rothberg. The complete genome of an individual by massively parallel DNA sequencing. *Nature*, 452(7189):872–876, Apr 2008.

H. Yang, X. Chen, and W. H. Wong. Completely phased genome sequencing through chromosome sorting. *Proc. Natl. Acad. Sci. U.S.A.*, 108(1):12–17, Jan 2011.