UNIVERSITY OF CALIFORNIA SAN DIEGO

Haplotype Assembly and Small Variant Calling using Emerging Sequencing Technologies

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

in

Computer Science

by

Peter Joseph Edge

Committee in charge:

Professor Vikas Bansal, Chair
Professor Vineet Bafna
Professor Melissa Gymrek
Professor Pavel Pevzner
Professor Kun Zhang

2019

The dissertation of Peter Joseph Edge is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

_____

_____

_____

_____

_____

Chair

University of California San Diego

2019

DEDICATION

To my parents, Chris and Karen, who always encouraged me to follow my passions. And to my grandpa, Ron, the original Dr. Edge.

EPIGRAPH

One never notices what has been done;

one can only see what remains to be done.

—Marie Curie

However difficult life may seem,

there is always something you can do, and succeed at.

It matters that you don't just give up.

—Stephen Hawking

I see this as an absolute win!

—Bruce Banner

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

Thanks to Vikas Bansal for being an excellent mentor.

Thanks to Vineet Bafna and the rest of my thesis committee.

Thanks to Eric Chu and the others I've had the opportunity to collaborate with.

Thanks to my parents and my family for always being there for me.

Thanks to Anish Tondwalkar for helping me in a hard time.

Thanks to the doctors and physical therapists who have helped me over the last 6 months: Jonathon Walters, Amy Trautman, Lisa Harris, Lawrence Donovan, Jeffrey Payne, Aaron Hellem, and Sarah Anderson.

Thanks to all of my amazing friends for their love and support!

Chapter 2 and Appendix A, in full, are a reprint of the material as it appears in HapCUT2: robust and accurate haplotype assembly for diverse sequencing technologies. Edge, Peter; Bafna, Vineet; Bansal, Vikas. Genome Research, 27(5), pp.801-812. Cold Spring Harbor Laboratory Press, 2017. The dissertation author was the primary author of this paper.

Chapter 3, in part, is a reprint of the material as it appears in Ultraaccurate genome sequencing and haplotyping of single human cells. Chu, Wai Keung; Edge, Peter; Lee, Ho Suk; Bansal, Vikas; Bafna, Vineet; Huang, Xiaohua; Zhang, Kun. Proceedings of the National Academy of Sciences, 114(47), pp.12512-12517. PNAS, 2017. The dissertation author was the secondary author of this paper.

Chapter 4 and Appendix B, in full, are a reprint of the material as it appears in Longshot enables accurate variant calling in diploid genomes from single-molecule long read sequencing, 2019. Edge, Peter; Bansal, Vikas. Nature Communications, 10(1), pp.1-10. Nature Publishing Group, 2019. The dissertation author was the primary author of this paper.

VITA

2014	B. S. in Computer Science, University of Minnesota - Twin Cities
2014	B. S, in Genetics, Cell Biology and Development, University of Minnesota - Twin Cities
2017	M. S. in Computer Science, University of California, San Diego
2019	Ph. D. in Computer Science, University of California, San Diego

PUBLICATIONS

Sundaram, V., Cheng, Y., Ma, Z., Li, D., Xing, X., Edge, P., Snyder, M.P. and Wang, T., 2014. Widespread contribution of transposable elements to the innovation of gene regulatory networks. *Genome research*, 24(12), pp.1963-1976.

Patel, A., Edge, P., Selvaraj, S., Bansal, V. and Bafna, V., 2016. InPhaDel: integrative shotgun and proximity-ligation sequencing to phase deletions with single nucleotide polymorphisms. *Nucleic acids research*, 44(12), pp.e111-e111.

Edge, P., Bafna, V. and Bansal, V., 2017. HapCUT2: robust and accurate haplotype assembly for diverse sequencing technologies. *Genome research*, 27(5), pp.801-812.

Chu, W.K., Edge, P., Lee, H.S., Bansal, V., Bafna, V., Huang, X. and Zhang, K., 2017. Ultraaccurate genome sequencing and haplotyping of single human cells. *Proceedings of the National Academy of Sciences*, 114(47), pp.12512-12517.

Edge, P. and Bansal, V., 2019. Longshot enables accurate variant calling in diploid genomes from single-molecule long read sequencing. *Nature communications*, 10(1), pp.1-10.

ABSTRACT OF THE DISSERTATION

Haplotype Assembly and Small Variant Calling using Emerging Sequencing Technologies

by

Peter Joseph Edge

Doctor of Philosophy in Computer Science

University of California San Diego, 2019

Professor Vikas Bansal, Chair

Short read DNA sequencing technologies from Illumina have made sequencing a human genome significantly more affordable, greatly accelerating studies of biological function and the association of genetic variants to disease. These technologies are frequently used to detect small genetic variants such as single nucleotide variants (SNVs) using a reference genome. However, short read sequencing technologies have several limitations. First, the human genome is diploid and short reads contain limited information for assembling haplotypes, or the sequences of alleles on homologous chromosomes. Moreover, there is significant input DNA required, which poses challenges for analyzing single cells. Further, there is limited ability to detect genetic variants inside long duplicated sequences that occur in the genome. As a result, there has been widespread

development of novel methods to overcome these deficiencies using short reads. These include clone based sequencing, linked read sequencing, and proximity ligation sequencing, as well as various single cell sequencing methods. There are also entirely new sequencing technologies from Pacific Biosciences and Oxford Nanopore Technologies that produce significantly longer reads. While these emerging methods and technologies demonstrate improvements compared to short reads, they also have properties and error modalities that pose unique computational challenges. Moreover, there is a shortage of bioinformatics methods for accurate small variant detection and haplotype assembly using these approaches compared to short reads. This dissertation aims to address this problem with the introduction of several new algorithms for highly accurate haplotype assembly and SNV calling. First, it introduces HapCUT2, an algorithm that can rapidly assemble haplotypes using a broad range of sequencing technologies. Second, it introduces an algorithm for variant calling and haplotyping using SISSOR, a recently introduced microfluidics based technology for sequencing single cells. Finally, it introduces Longshot, an algorithm for detecting and phasing SNVs using error-prone long read technologies. In each case, the algorithms are benchmarked using multiple real whole-genome sequencing datasets and are found to be highly accurate. The methods introduced in this dissertation contribute to the goal of sequencing diploid genomes accurately and completely for a broad range of scientific and clinical purposes.

Chapter 1

Introduction

1.1 The diploid human genome

The human genome describes the heritable material that controls cellular function and organism development. The human genome is comprised of 22 autosomal chromosomes as well as the sex chromosomes, which are DNA molecules comprised of four nucleotides. For this reason, each chromosome can be represented as a string of the four letters A, C, G, T. Since humans are diploid, there are two "homologous" copies of each autosomal chromosome (one copy from each parent) that are highly similar to one another ($\sim$99.9% identical), but have different alleles (alternate forms) at the differing sites. The sequences of alleles on homologous chromosome copies are known as haplotypes. The human genome is approximately 3.2 billion bases long in total, or approximately 6.4 billion bases considering its diploid nature. The genome contains genes, which encode the sequences of proteins that are responsible for a large amount of cellular function. Therefore, variations between genomes can be tied to differences in phenotype (observable traits), including genetic diseases.

1.2 Advances in DNA sequencing technology

The process of reading the DNA bases in a genome is called "DNA sequencing". The first complete draft of the human genome was completed in 2003 after 13 years of effort with a total cost of 2.7 billion dollars [1]. This draft was constructed by the collaboration of many labs, using an expensive and labor-intensive method of DNA sequencing. In 2006, the first "second generation sequencing" method was introduced that provided significantly greater throughput and cheaper operation [2]. Since then, the price of second generation sequencing has continued to drop, and it is now possible to sequence a human genome for less than $1,000 [3]. This trend has resulted in increased interest in sequencing the genomes of many individuals in order to better understand how genetic variations relate to observable traits and human disease [4]. It is believed that routine and affordable DNA sequencing will bring in an age of personalized medicine, when medical treatments will be custom tailored to an individual's genome to yield optimal results [5, 6].

1.3 Single nucleotide variant calling

All available DNA sequencing methods work by sampling subsequences, or reads, from the full DNA sequence. The reads contain errors, but by sampling with redundancy the reads can be used to determine the overall sequence. Determining the entire sequence without prior knowledge, also known as de novo assembly, requires long read lengths and significant computation in order to determine the entire sequence unambiguously. It is common instead to align the reads to a "reference genome" and detect the variants, or differences, between the sample and the reference genome. 90% of variants in the genome are alterations of a single base, known as single nucleotide variants (SNVs)[1] [7, 8]. Using the common language of DNA variants also enables

[1]The term single nucleotide polymorphism (SNP) is sometimes used interchangeably with the term SNV, but is now usually understood to refer to SNVs that have a frequency of at least 1% in the population

the creation of databases that associate those variants to phenotypes and diseases. There are many known associations of SNVs to diseases, including lung cancer, type II diabetes, and a variety of mendelian disorders [9, 10]. Oftentimes, observed traits are the result of the combined effect of many variants or complex interactions between different DNA variants [11, 12].

1.4 Limitations of second generation sequencing

While second generation sequencing brought on an era of cheap DNA sequencing and large-scale sequencing studies, it has notable limitations. Firstly, the method produces significantly shorter read lengths then previous methods, and for this reason it is commonly referred to as "short read sequencing" [13]. While short read sequencing enables the detection of SNVs and short insertion deletion (indel) variants, it reveals little to no information about haplotypes (also known as the variant phase). Most short reads cover at most a single variant site, so it is not possible to associate alleles on adjacent sites to the same haplotype. Haplotype information is functionally relevant and necessary for complete prediction of phenotype. For example, haplotype information can predict diseases caused by a phenomenon known as compound heterozygosity [14].

Another limitation of short-read sequencing, which is common to most sequencing methods, is that there is appreciable input DNA required to accurately sequence a sample. While it is common to perform short read sequencing using a bulk tissue sample, there is significant clinical and scientific interest in sequencing the genomes of single cells[15, 16, 17]. Because of the input DNA required for short read sequencing, this is commonly performed by whole genome amplification (repeated DNA replication) followed by short read sequencing. This results in significant false positive variants resulting from amplification error [18].

Another limitation of short read sequencing has to do with the fact that reference-based analysis requires reads to be "mapped" unambiguously to the appropriate site in the reference genome. Approximately 3.6% of the genome consists of long duplicated sequences where the

short reads cannot be accurately mapped, since it is not known which copy of the duplication a read belongs to [19]. In order to detect variants in these regions, it is usually necessary to use longer reads that can span repetitive sequence.

1.5 New technologies and new challenges

There has been widespread development of new technologies and protocols aiming to address the deficiencies of short read sequencing. These include methods to encode information about a larger DNA molecule into short reads using a pooling or partitioning based strategy. These include clone based sequencing as well as linked-read sequencing [20, 21]. These strategies allow the assembly of long haplotypes using short reads, but the read information may be sparsely distributed over the original molecule (in the case of linked reads) or have errors when short reads are mistakenly attributed to the wrong molecule (more common in clone based sequencing). There are also techniques such as proximity ligation sequencing which encode spatial information about the genome into short reads, and this method can be used to assemble haplotypes [22, 23]. This technique can result in haplotype errors caused by spatial interactions with homologous chromosomes. New single-cell sequencing methods have been developed, and for these it is necessary to overcome errors from genome amplification [24]. There are also entirely new "3rd generation" sequencing technologies such as those from Pacific Biosciences (PacBio) and Oxford Nanopore Technologies. These technologies produce significantly longer reads than second generation sequencing, at the cost of a significantly higher error rate [25, 26]. Each of these new technologies introduces benefits over traditional short reads, but also introduces new challenges that must be overcome with computational techniques.

1.6 Scope of the thesis

Despite encouraging progress for these new emerging sequencing technologies, there is nonetheless a shortage of bioinformatics methods for accurately detecting SNVs and assembling haplotypes using data from these methods compared to those for short reads. The scope of this thesis is the discovery of novel computational techniques for detecting SNVs and assembling haplotypes using emerging sequencing technologies.

First, we consider the problem of haplotype assembly. Chapter 2 introduces an algorithm, HapCUT2, that is designed to assemble haplotypes quickly and accurately using a wide variety of sequencing technologies. We show that HapCUT2 rapidly assembles haplotypes with best-in-class accuracy using multiple different data types, including clone-based sequencing, linked-read sequencing, single molecule real-time (SMRT) sequencing, and proximity ligation (Hi-C) sequencing.

Secondly, we consider the problem of single-cell sequencing. As mentioned earlier, single-cell sequencing usually requires the error-prone process of whole genome amplification. Chapter 3 considers a new method, SISSOR, that uses a microfluidic device to enable highly accurate variant calling and haplotyping for single cells by amplifying the single strands of the double stranded DNA molecule separately. This high accuracy is achieved with the help of a novel algorithm that models the SISSOR protocol and the unique error modalities it presents, as well as haplotype-assembly-based analyses.

Finally, we consider the problem of detecting SNVs using error-prone reads. Third-generation sequencing technologies such as those from Pacific Biosciences and Oxford Nanopore Technologies offer significantly greater read lengths than short read sequencing, at a higher per base error rate. These longer reads enable the assembly of long haplotypes as well as genotyping variants that occur in duplicated regions with low short read mappability. However, there are limited methods for detecting SNVs in diploid organisms using these read technologies. Chapter

4 introduces an algorithm, Longshot, that performs highly accurate SNV calling and haplotype assembly for error prone long read technologies.

Chapter 2

HapCUT2: robust and accurate haplotype assembly for diverse sequencing technologies

2.1 Abstract

Many tools have been developed for haplotype assembly - the reconstruction of individual haplotypes using reads mapped to a reference genome sequence. Due to increasing interest in obtaining haplotype-resolved human genomes, a range of new sequencing protocols and technologies have been developed to enable the reconstruction of whole-genome haplotypes. However, existing computational methods designed to handle specific technologies do not scale well on data from different protocols. We describe a new algorithm, HapCUT2, that extends our previous method (HapCUT) to handle multiple sequencing technologies. Using simulations and whole-genome sequencing (WGS) data from multiple different data types – dilution pool sequencing, linked-read sequencing, single molecule real-time (SMRT) sequencing, and proximity ligation (Hi-C) sequencing – we show that HapCUT2 rapidly assembles haplotypes with best-

in-class accuracy for all data types. In particular, HapCUT2 scales well for high sequencing coverage and rapidly assembled haplotypes for two long-read WGS datasets on which other methods struggled. Further, HapCUT2 directly models Hi-C specific error modalities resulting in significant improvements in error rates compared to HapCUT, the only other method that could assemble haplotypes from Hi-C data. Using HapCUT2, haplotype assembly from a $90\times$ coverage whole-genome Hi-C dataset yielded high-resolution haplotypes (78.6% of variants phased in a single block) with high pairwise phasing accuracy ($\sim 98\%$ across chromosomes). Our results demonstrate that HapCUT2 is a robust tool for haplotype assembly applicable to data from diverse sequencing technologies.

2.2 Introduction

Humans are diploid organisms with two copies of each chromosome (except the sex chromosomes). The two *haplotypes* (described by the combination of alleles at variant sites on a single chromosome) represent the complete information on DNA variation in an individual. Reconstructing individual haplotypes has important implications for understanding human genetic variation, interpretation of variants in disease, and reconstructing human population history [27, 28, 29, 30]. A number of methods, computational and experimental, have been developed for haplotyping human genomes. Statistical methods for haplotype phasing using population genotype data have proven successful for phasing common variants and for genotype imputation but are limited in their ability to phase rare variants and phase long stretches of the genome that cross recombination hot-spots [27, 31].

Haplotypes for an individual genome at known heterozygous variants can be directly reconstructed from reference-aligned sequence reads derived from whole-genome sequencing. Sequence reads that are long enough to cover multiple heterozygous variants provide partial haplotype information. Using overlaps between such haplotype-informative reads, long haplotypes

can be assembled. This *haplotype assembly* approach does not rely on information from other individuals (such as parents) and can phase even individual-specific variants. Levy et al. [32] demonstrated the feasibility of this approach using sequence data derived from paired Sanger sequencing of long insert DNA fragment libraries to computationally assemble long haplotype blocks (N50 of 350 kb) for the first individual human genome.

Since then, advancements in massively parallel sequencing technologies have reduced the cost of human WGS drastically, leading to the sequencing of thousands of human genomes. However, the short read lengths generated by technologies such as Illumina (100-250 bases) and the use of short fragment lengths in WGS protocols makes it infeasible to link distant variants into haplotypes. To overcome this limitation, a number of innovative methods that attempt to preserve haplotype information from long DNA fragments (tens to hundreds of kilobases) in short sequence reads have been developed.

The underlying principle for these methods involves generating multiple pools of high-molecular-weight DNA fragments such that each pool contains only a small fraction of the DNA from a single genome. As a result, there are very few overlapping DNA fragments in each pool and high-throughput sequencing of the DNA in each pool can be used to reconstruct the fragments by alignment to a reference genome [33, 34]. Therefore, each pool provides haplotype information from long DNA fragments and long haplotypes can be assembled using information from a sufficiently large number of independent pools [30]. A number of methods based on this approach have been developed to phase human genomes [33, 34, 35, 36, 37]. Recently, 10X Genomics described a novel microfluidics based library preparation approach that generates long linked-reads that can be assembled into long haplotypes [21]. Third-generation sequencing technologies such as Pacific Biosciences (PacBio) generate long sequence reads (2-20 kilobases in length) that can directly enable genome-wide haplotyping. Pendleton and colleagues demonstrated the feasibility of assembling haplotypes from SMRT reads using variants identified from short read Illumina sequencing [38].

Haplotype assembly is also feasible with paired-end sequencing, i.e. pairs of short reads derived from the ends of long DNA fragments, but requires long and variable insert lengths to assemble long haplotypes [27]. Selvaraj et al. [23] used sequence data from a proximity ligation method (Hi-C) to assemble accurate haplotypes for mouse and human genomes. Using mouse data, they demonstrated that the vast majority of intra-chromosomal Hi-C read pairs correspond to 'cis' interactions (between fragments on the same chromosome) and therefore contain haplotype information equivalent to paired-end reads with long and variable insert lengths. Subsequently, $17\times$ whole-genome Hi-C data was used to assemble chromosome-spanning haplotypes for a human genome, albeit with low resolution (less than 22% of variants phased).

In summary, multiple different sequencing technologies and protocols have the capability to generate sequence reads with haplotype information, but require computational tools to assemble the reads into long haplotypes. A number of combinatorial algorithms have been developed for haplotype assembly [39, 40, 41, 42]. Among these, HapCUT [39] was developed for phasing Sanger WGS data for the first individual genome [32]. HapCUT utilizes max-cuts in read-haplotype graphs, an approach that is equally adept at handling data with local haplotype information and data with long-range haplotype information such as that from long insert paired-end reads. As a result, it has been successfully utilized to assemble haplotypes from different types of high-throughput sequence datasets including fosmid pool sequencing [33], Hi-C data [23], and single molecule long reads [38] with appropriate modifications. However, HapCUT only models simple sequencing errors and does not scale well for long read data. More recently, several algorithms have been designed specifically to enable accurate haplotype assembly from long reads [41, 43].

The diverse characteristics and specific error modalities of data generated by different haplotype-enabling protocols and technologies continue to pose challenges for haplotype assembly algorithms. Some protocols, such as clone-based sequencing, can generate very long fragments (BAC clones of length 140 kb have been used to assemble haplotypes [44]) but may have low

fragment coverage. Other protocols, such as PacBio SMRT, generate fragments with shorter mean lengths than clone-based approaches but can be scaled to higher read coverage more easily. 10X Genomics linked-reads are long (longest moleculars $>$ 100 kilobases) but have gaps resulting in high clone coverage for each variant. Proximity ligation approaches, such as Hi-C, generate paired-end read data with very short read lengths, but with larger genomic span. Hi-C reads can span from a few kilobases to tens of megabases in physical distance. While an algorithm that leverages characteristics of a specific type of data is likely to perform well on that particular type of data, it may not perform well or not work at all on other types of data. For example, dynamic programming algorithms such as ProbHap [43] that were developed for low-depth long read sequence data are unlikely to scale well for datasets with high sequence coverage or for other types of data such as Hi-C. Even if a haplotype assembly algorithm has broad support for data qualities, there remains the challenge that different sequencing protocols each have systematic error modalities. For instance, fragment data from the sequencing of multiple haploid subsets of a human genome [33, 34] generates long haplotype fragments, but some of these fragments are chimeric due to overlapping DNA molecules that originate from different chromosomes. Similarly, noise in Hi-C data due to ligated fragments from opposite homologous chromosomes increases with increasing distance between the variants. The accuracy of haplotypes assembled from each sequencing protocol depends on both the haplotype assembly algorithm's ability to effectively utilize the sequence data and its ability to model protocol-specific errors.

2.3 Results

To address the challenge of haplotype assembly for diverse types of sequence datasets, we developed HapCUT2, an algorithm that generalizes the HapCUT approach in several ways. Compared to a discrete score optimized by HapCUT, HapCUT2 uses a likelihood-based model, which allows for the modeling and estimation of technology-specific errors such as 'h-trans

errors' in Hi-C data. To improve memory performance for long read data, HapCUT2 does not explicitly construct the complete read-haplotype graph. Further, it implements a number of optimizations to enable fast runtimes on diverse types of sequence datasets. To demonstrate the accuracy and robustness of HapCUT2, we compared its performance with existing methods for haplotype assembly using simulated and real WGS datasets. Previous publications [20, 43] have compared different methods for haplotype assembly and concluded that RefHap [41], ProbHap [43], FastHare [45] and HapCUT [39] are among the best performing methods. Other methods such as DGS [32], MixSIH [46] and HapTree [47] did not perform as well on the datasets evaluated in this study. Therefore, we compared the performance of HapCUT2 with four other methods: RefHap, ProbHap, FastHare and HapCUT (Table 2.1).

2.3.1 Overview of HapCUT2 algorithm

The input to HapCUT2 consists of haplotype fragments (sequence of alleles at heterozygous variant sites identified from aligned sequence reads) and a list of heterozygous variants (identified from WGS data). HapCUT2 aims to assemble a pair of haplotypes that are maximally consistent with the input set of haplotype fragments. This consistency is measured using a likelihood function that captures sequencing errors and technology specific errors such as h-trans errors in proximity ligation data. HapCUT2 is an iterative procedure that starts with a candidate haplotype pair. Given the current pair of haplotypes, HapCUT2 searches for a subset of variants (using max-cut computations in the read-haplotype graph) such that changing the phase of these variants relative to the remaining set of variants results in a new pair of haplotypes with greater likelihood. This procedure is repeated iteratively until no further improvements can be made to the likelihood (see Methods for details).

2.3.2 Comparison of runtimes on simulated data

We used simulations to compare the runtime of HapCUT2 with existing methods for haplotype assembly across different types of sequence datasets. A fair comparison of the performance of different methods is not completely straightforward. Different methods chose to optimize different technology parameters and highlighted performance using those parameters. We considered the following parameters: number of variants per read (V), coverage per variant (d), and the number of paired-end reads spanning a variant (d'). The parameter V is a natural outcome of read length; for example, PacBio provides higher values of V compared to Illumina sequencing. The parameter d is similar to read coverage, but only considers haplotype informative reads – higher values result in better accuracy, but also increased running time. Finally, many sequencing technologies (such as Hi-C) generate paired-end sequencing with long inserts and d' can potentially be much greater than d. Some haplotype assembly methods implicitly analyze all paired-end reads spanning a specific position and their runtime depends upon d' rather than d.

In order to make a fair comparison of runtimes and allow users to determine the most efficient method for any technology, we summarized the computational complexity of each method as a function of these parameters (Table 2.1) and used simulations to verify the dependence of runtime and accuracy on each parameter (Figure 2.1). We simulated reads using a single chromosome of length $\sim$250 Mb (approximately equal to the length of human chromosome 1) with a heterozygous variant density of 0.08 % and a uniform rate of sequencing errors (2%), performing 10 replicates for each simulation. Standard deviations of runtimes and error rates between replicates were small (Supplemental Fig A1). A method was cut off if it exceeded 10 CPU-hours of runtime or 8 GB of memory on a single CPU, since most methods required significantly less resources than these limits. We note that the runtimes in Table 2.1 refer to complexity as implemented, with parameters referring to maximum values (e.g. maximum coverage per variant), while in simulations the parameters refer to mean values (e.g. mean coverage per variant).

To assess the dependence of runtime on d, we generated reads with a mean of 4 variants per read (V) and varied the mean read coverage per variant (d) from 5 to 100. The error rates of HapCUT2, HapCUT, ProbHap, and RefHap were similar and decreased with increasing coverage before reaching saturation. FastHare was significantly faster than other methods, but had error rates that were several times greater. As predicted by the computational complexity of the different methods (Table 2.1), HapCUT2 is significantly faster than HapCUT, RefHap, and ProbHap, once the coverage exceeds $10\times$ (Figure 2.1A). For example, RefHap required 10 CPU-hours to phase reads at a coverage of $38\times$, while HapCUT2 took only 34 CPU-minutes to phase reads with $100\times$ coverage per variant. ProbHap reached the 10 CPU-hour limit at a coverage of only $8\times$. HapCUT shows a similar trend to HapCUT2, but is significantly slower and requires more than 8 GB of memory at coverages of $40\times$ or greater. RefHap constructs a graph with the sequence reads as nodes and performs a max-cut operation that scales quadratically with number of reads. Therefore, its runtime is expected to increase as the square of read-coverage. ProbHap's runtime is exponential in the maximum read-depth [43] and exceeds the maximum allotted time for modest values of d. FastHare greedily builds a maximally consistent haplotype from left to right in a single pass, resulting in a low run-time but also lower accuracy. While HapCUT2 has the same asymptotic behavior as HapCUT, it improves upon the memory usage and runtime significantly in practice. It does this by only adding edges that link adjacent variants on each read to the read-haplotype graph, as well as using convergence heuristics that reduce the number of iterations performed (see Methods for details).

Next, we varied the number of variants per read (V) and kept the coverage per variant (d) fixed at $5\times$. The error rates for each method decrease monotonically (Figure 2.1B). HapCUT2, RefHap, and ProbHap have similarly low error rates, while FastHare and HapCUT have error rates higher than the other methods. The runtimes of RefHap and FastHare are consistently very low, although the runtime of RefHap peaks very slightly around $V = 15$. The runtime of ProbHap decreases monotonically as V increases. This is consistent with the fact that the runtime of these

methods has a linear dependence on the read length because for a fixed sequence coverage, the number of reads decreases as the read length increases. In comparison, HapCUT2's runtime is observed to increase linearly with V. This is consistent with the complexity of HapCUT2 being proportional to the square of the number of variants per read (see Table 2.1). Although HapCUT2's runtime increases, it remains practical across all tested values and is less than 50 CPU-minutes for mean read lengths consistent with very long sequences (160 variants per read or 200 kilobases). The space requirements for HapCUT have a quadratic dependence on the number of variants per read, and therefore, exceeded the memory limit after only 8 variants per read.

Finally, we compared runtimes as a function of the average number of paired-end reads crossing a variant (d'). For single-end reads, this parameter is identical to d. Proximity ligation data, on the other hand, consists of pairs of short reads each with a single large gap (insert) between them. The large and highly variable insert sizes result in a large number of reads crossing each variant position. This property is important for linking distant variants, because the extremely long insert size spans of proximity ligation methods are capable of spanning long variant-free regions. For this reason, we simulated paired-end short read data with random insert sizes up to a parametrized maximum value, to represent a generalized proximity ligation experiment. We varied d' by increasing the maximum insert size value from 6.25 kb ($\sim$5 SNVs) to 125 kb ($\sim$100 SNVs) while keeping d and V constant at 5$\times$ and 150 base pairs (0.1195 SNVs), respectively. ProbHap and RefHap exceeded the time limit at $d' = 10$ and $d' = 17$, respectively. FastHare exceeded the time limit at $d' = 36$, but had extremely high error rates (10 to 18 times higher than HapCUT2). ProbHap's dynamic programming algorithm needs to consider the haplotype of origin for each read crossing a variant, therefore the complexity scales exponentially in d'. In the case of RefHap and FastHare, the failure to scale with increasing d' appears to be a result of representing fragments as continuous arrays with length equal to the number of variants spanned by each read. Thus, as implemented, the runtimes for RefHap and FastHare scale with d' rather than d. In contrast, both HapCUT and HapCUT2 were able to phase data with arbitrarily long

insert lengths, reaching $d' = 100$ (Figure 2.1C). The runtime of HapCUT2 was independent of d' and 8 to 10$\times$ times faster than that for HapCUT.

Overall, the results on simulated data demonstrate that the complexity of HapCUT2 is linear in the number of reads and quadratic in the number of variants per read. HapCUT2 is fast in practice and effective for both long reads and paired-end reads with long insert lengths, with scalability unmatched by the four other tools we evaluated. Additionally, HapCUT2 and HapCUT were the only tools tested that can reasonably phase paired-end data with long insert lengths that result from proximity ligation (Hi-C) sequencing.

2.3.3 Comparison of methods on diverse WGS datasets for a single individual

We next assessed the accuracy of HapCUT2 using data from four different sequencing data types for a single individual (NA12878): fosmid-based dilution pool sequencing, 10X Genomics linked-read sequencing, SMRT sequencing, and proximity ligation sequencing. Haplotype assembly methods require a set of heterozygous variants as input. Therefore, a set of heterozygous variants for NA12878 identified from WGS Illumina data were used as input to assemble haplotypes for each data type (see Methods for description). The accuracy of the haplotypes was assessed by comparing the assembled haplotypes to gold-standard trio phased haplotypes and using the switch error rate and mismatch error rate metrics (see Methods).

Fosmid-based dilution pool data: To assess HapCUT2 on long read sequencing data, we used whole-genome fosmid-based dilution pool sequence data for a human individual, NA12878 [20]. This data was generated from 1.44 million fosmids (33-38 kb and 38-45 kb in length) that were partitioned into 32 pools such that each pool contains DNA from a small fraction of the genome ($\sim$5%). Subsequently, each pool was sequenced using the ABI SOLiD sequencer and haplotype fragments identified using read depth analysis [20]. Although this dataset has low sequence coverage ($d \approx 3\times$), the processed fragment data (needed as input for haplotype

assembly) is publicly available and has been used to assess the performance of haplotype assembly methods in several papers [20, 43]. On this data, the switch error and the mismatch error rates for HapCUT2 were virtually identical or slightly better than ProbHap, the second best performing method, across all chromosomes (Supplemental Fig A2). However, ProbHap pruned approximately 1.2% of the variants from the assembled haplotypes in comparison to HapCUT2 which only pruned 0.6% of the variants. The switch error rates for RefHap and FastHare were also similar to HapCUT2 and ProbHap (Supplemental Fig A2). To enable a head-to-head comparison of the switch error rate across different methods, we also calculated the switch and mismatch error rates on a subset of variants that were phased by all tools (not pruned). On this subset of variants, the switch and mismatch error rates for HapCUT2 were similar to but slightly lower than ProbHap (Figure 2.2A). In terms of running time, RefHap and FastHare were the fastest methods on this dataset while HapCUT2 took a total of 1:09 CPU-hour to phase all chromosomes (Table 2.2). In summary, HapCUT2 had similar (but slightly better) accuracy to ProbHap, RefHap and FastHare on this dataset and was more accurate than HapCUT.

10X Genomics linked-read data: We also used HapCUT2 to assemble haplotypes from 10X Genomics linked-read data [21], which is based on a similar ideas as the fosmid-based dilution pool approach. 10X Genomics technology labels short DNA fragments originating from a single long DNA fragment with barcodes inside hundreds of thousands of separate nano-scale droplets [21]. The linked-reads produced can be extremely long ($>$100 kb). This dataset has a short read coverage of 34$\times$, with a linked-read coverage per variant of 12$\times$ [48]. For haplotype assembly, we used the same set of variant calls as for the fosmid dataset and extracted haplotype fragments from the 10X aligned reads (see Methods, "Long read datasets"). On this dataset, neither RefHap nor ProbHap finished haplotype assembly within the time limit. HapCUT2 was the fastest method and analyzed all chromosomes in 1:55 CPU-hours (Table 2.2). When compared on the subset of variants that were phased by all tools, HapCUT2 had an accuracy slightly better than the next best approach (HapCUT), which took 16:50 CPU-hours (Figure 2.2C).

PacBio SMRT data: SMRT sequencing on the Pacific Biosciences platform generates long (2-20 kilobases) but error prone ($>$ 10 % indel error rate) reads. We used HapCUT2 to assemble haplotypes from 44$\times$ coverage PacBio reads [38]. We extracted haplotype fragments from the PacBio reads that were aligned to the human reference genome (hg19), using the same set of variant calls as for the previous two datasets. On the full dataset, HapCUT2 was not only the most accurate, but was also significantly faster than RefHap and HapCUT (see Supplemental Fig A3 for detailed comparisons of error rates and runtimes). We calculated the switch error and mismatch error rates on the subset of variants that were phased by all methods. HapCUT2 had a 12.4% lower switch error and a 2% lower mismatch rate than RefHap. RefHap took 215:53 CPU-hours to phase the dataset. By comparison, HapCUT2 took only 4:05 CPU-hours in total. Because ProbHap was unable to complete within the time limit on the full dataset, we also compared the performance of the haplotype assembly tools on a lower, 11$\times$ coverage subsample of this dataset. On the subsample, HapCUT2 had the lowest switch error and mismatch error rates of the five methods (Figure 2.2B). FastHare was the fastest method on this dataset and ProbHap was the slowest method taking 52:32 CPU-hours (Table 2.2).

HapCUT2 implements likelihood-based strategies for pruning low-confidence variants to reduce mismatch errors and splitting blocks at poor linkages to reduce switch errors (see Methods). These post-processing steps allow a user to improve accuracy of the haplotypes at the cost of reducing completeness and contiguity. ProbHap's "transition, posterior, and emission" confidence scores are designed for the same purpose [43]. Post-processing strategies are of particular interest for haplotype assembly with PacBio SMRT reads because the individual reads have a high error rate. Therefore, we compared HapCUT2's pruning strategies to ProbHap's confidence scores on chromosome 1 of the 11$\times$ coverage PacBio data. For single variant pruning, we found that HapCUT2's confidence scores provided a better trade-off between reducing the mismatch error rate and pruning variants compared to ProbHap's emission scores (Supplemental Fig A4 (A)). By pruning 3.1% of covered variants, HapCUT2 achieved a 55.1% reduction in mismatch error

rate. In comparison, ProbHap mismatch error rate was reduced by less than 15% when 3.1% of variants were pruned. Similarly, HapCUT2's block splitting strategy resulted in a lower switch error rate compared to ProbHap at a fixed value of the AN50 for the haplotype assembly except at very small AN50's (Supplemental Fig A4 (B)).

Hi-C data: The Hi-C method was developed to comprehensively detect chromatin interactions in the cell nucleus using proximity ligation and shotgun sequencing of the ligation products [22]. Selvaraj et al. [23] demonstrated that the long-range information contained in Hi-C reads can be used to determine the phase between distant variants and assemble chromosome-spanning haplotypes from $\sim$17$\times$ coverage. They collaborated with some of the authors of the current study in customizing HapCUT to assemble haplotypes from Hi-C data. Hi-C reads suffer from a source of error that was referred to as "h-trans interactions". An h-trans interaction (or h-trans error) occurs when a piece of DNA interacts with a DNA fragment from the homologous chromosome rather than the same chromosome (a "cis" interaction). The probability of h-trans error depends on the insert size and can be estimated if the true haplotypes are known. We use the function $\tau(I)$ to refer to the probability of an h-trans error for a read with insert size I. Selvaraj et al. [23] estimated τ using Hi-C data from a mouse genome and used these estimates to lower the base quality values of reads before running HapCUT. In developing HapCUT2, we were motivated in part by the need to develop a method that could estimate τ directly from the data and use these estimates to improve the accuracy of the haplotypes.

To assess different haplotype assembly tools, we used a high coverage Hi-C dataset with $\sim$395$\times$ coverage on NA12878 generated using the MboI enzyme [49] and sub-sampled reads from this dataset to generate 40$\times$ and 90$\times$ coverage. As expected from the simulations using paired-end reads with variable span, only HapCUT and HapCUT2 were able to generate haplotypes from Hi-C data within the 20 CPU-hour per chromosome time limit. The error rates were significantly lower for the 90$\times$ sample, and HapCUT2 had lower error rates compared to HapCUT at both coverage levels (Figure 2.2D). In terms of runtime, HapCUT2 was 4 to 5 times

slower than HapCUT on Hi-C data since it performs the haplotype assembly procedure multiple times in order to estimate τ. We note that if HapCUT2 does not account for h-trans errors, it is several times faster than HapCUT (Supplemental Table A1).

At 40$\times$ coverage, HapCUT2 achieved a 16.3% lower switch error rate and 13.2% lower mismatch rate compared to HapCUT on variants phased by both methods. Similarly, at 90$\times$ coverage, HapCUT2 achieved a 16.4% lower switch error rate and 7.2% lower mismatch rate compared to HapCUT. The lower error rates for HapCUT2 are primarily due to the modeling and estimation of h-trans errors in Hi-C data. HapCUT2 directly models h-trans errors as probabilities in the likelihood formulation and estimates τ directly from the data using an iterative approach (see Methods, "Estimation of h-trans error probabilities in Hi-C data"), eliminating the need for a model dataset with known haplotypes. The h-trans function was estimated separately for each chromosome since we observed significant variation in the h-trans error rates across chromosomes (estimated using known haplotypes for NA12878). The per-chromosome h-trans error rates estimated by HapCUT2 were very similar to those obtained using known trio-phased haplotypes for NA12878 (see Supplemental Fig A5) demonstrating the accuracy of the estimation procedure.

Overall, results on a variety of sequence datasets reaffirm what we observed on simulated reads, i.e. HapCUT2 is the only tool that works across all sequencing paradigms. In particular, haplotype assembly tools that were developed to phase low-coverage long read data, such as ProbHap and RefHap, do not work on Hi-C data. Even on long read data (PacBio SMRT sequencing and 10X Genomics linked-reads), HapCUT2 scales better in running time with increasing coverage (Table 2.2). Moreover, it assembles haplotypes that are more accurate than all other methods that we evaluated in this paper. This was somewhat surprising because ProbHap implements an exact likelihood optimization approach. However, to reduce runtime, ProbHap also uses an initial heuristic to merge reads that convey similar information and this could reduce the accuracy.

2.3.4 Comparison of haplotypes assembled using Hi-C and SMRT sequencing

Sequencing technologies such as SMRT generate long reads that contain multiple *variants per read*. Although most of the reads contain haplotype information, the read length limits the ability to phase heterozygous variants that are separated by long runs of homozygosity. In contrast, paired-end reads derived from Hi-C contain very few variants per read pair (most read pairs do not cover any variant or cover only a single variant). However, read pairs that cover a variant at each end have the potential to link distant variants because the insert size of Hi-C reads varies from a few hundred bases to tens of millions of bases. Therefore, haplotypes assembled using these two approaches differ significantly in both contiguity and accuracy. We utilized the PacBio SMRT sequencing and MboI enzyme Hi-C datasets to compare the haplotypes assembled using HapCUT2 for these two technologies.

The haplotypes assembled from the 44× coverage SMRT sequence data had an AN50 length of 218 kb with the largest block being 1.66 Mb in length. Also, 99% of the heterozygous variants could be phased as part of some block. In contrast, the haplotypes assembled from the 90× coverage Hi-C data (for each autosomal chromosome) comprised of a large chromosome-spanning block that contains 72-87% of the variants in addition to numerous small blocks with a median block size of 2 variants. This effect can be observed in Figure 2.3A, which shows the cumulative number of variants covered by the haplotype blocks (sorted in descending order) for chromosome 1. One limitation of Hi-C data is that some of the variants that are far away from restriction enzyme cut-sites cannot be phased due to lack of reads covering such variants. Chromosome X, which has a lower variant density than autosomes, was more difficult to phase than autosomal chromosomes, with only 55% of SNVs in the largest block and 32% of variants unphased (Supplemental Fig A6).

In terms of accuracy measured using switch error rates, both technologies achieve com-

parable error rates (0.002-0.003) at sufficiently high coverage (Figure 2.2). Further, the switch error rates for haplotypes assembled using these two technologies decrease rapidly as coverage is increased initially and saturate quickly after that (Supplemental Fig A7). Although the switch and mismatch error rates are similar between high-coverage Hi-C data and high-coverage PacBio data, we found that these statistics do not adequately distinguish between short stretches of erroneous phased variants (masquerading as two switch errors) and "point" switches that effectively divide the resulting haplotype into two pieces. Long read data, because of its linear structure, has a tendency for this type of error. In comparison, Hi-C data is more web-like in structure and therefore has essentially no incidence of point switches once there is sufficient coverage. To observe the effect of point switches on accuracy for long reads compared to Hi-C, we plotted the fraction of correctly phased variant pairs separated by a given genomic distance (Figure 2.3B). Point switch errors accumulate linearly to diminish the probability of correct phasing with distance for PacBio, but not for Hi-C. For example, two variants separated by 200 kb and phased with the $11\times$ PacBio data have a 62% chance of being phased correctly. On the other hand, MboI Hi-C data maintains a high and constant rate of pairwise phasing accuracy across the entire chromosome: ~ 96 at $40\times$ and ~ 98 at $90\times$. This implies that not only do Hi-C haplotypes span the entire chromosome, but the overall haplotype structure is also highly accurate.

2.3.5 Considerations when haplotyping with Hi-C

For Hi-C based haplotyping, the choice of restriction enzyme (RE) and depth of sequence coverage can impact the completeness and accuracy of the haplotypes. Selvaraj et al. [23] were able to assemble chromosome-spanning haplotypes for NA12878 using Hi-C data generated using the HindIII RE. Despite assembling blocks that spanned the genomic distance of each chromosome, the "resolution" of the largest block was rather low (only 18-22% of the variants on each chromosome could be linked into haplotypes). The low resolution could potentially be due to the modest sequence depth (approximately $17\times$). However, we observe that even at

200× coverage, Hi-C data obtained using the HindIII RE from the Rao et al. [49] study has less than 40% of variants phased in the largest block, with 71% of variants phased in total. In contrast, 80% of the heterozygous variants on chromosome 1 can be successfully phased in a single block using only 90× Hi-C data obtained using the MboI RE. The trend is similar across autosomal chromosomes, with the largest block of each chromosome containing 72% to 87% of the heterozygous variants (Supplemental Fig A6). This indicates that the choice of RE has an important effect on the number of variants that can be phased.

A key step in the Hi-C protocol is the digestion of cross-linked DNA by an RE that cleaves DNA at specific recognition sites. In comparison to the HindIII RE which recognizes a 6 base pair DNA sequence (Aˆ AGCTT), the MboI RE has a 4 base pair recognition site which occurs with much greater frequency in the genome (ˆGATC). The significantly greater completeness of the haplotypes assembled using Hi-C data generated using the MboI RE is primarily due to this reason. Even with an RE with a 4 base pair recognition sequence, some fraction of variants are expected to be far away from a cut-site and therefore, cannot be captured in Hi-C ligated fragments. Indeed, the fraction of SNVs phased using the MboI Hi-C data saturates with increasing coverage and 7.3% of SNVs on chromosome 1 cannot be phased into the largest block even at 395× coverage. The fraction of variants phased can potentially be increased by integrating Hi-C data from different REs or by using imputation based approaches.

At low sequence coverage ($< 25\times$), the largest haplotype block assembled using MboI derived Hi-C data contains less than 40% of the variants (Figure 2.4A) and has a high error rate (Figure 2.4B). With increasing sequence coverage, the fraction of the variants in the largest component increases rapidly and the error rate of the largest haplotype block (measured as the sum of the switch and mismatch error rates) decreases rapidly. The improvements in both these aspects of Hi-C based haplotype assembly saturate around 80-100× coverage. These results demonstrate that highly accurate, high-resolution, chromosome-spanning haplotypes can be assembled from 80-100× whole-genome Hi-C data for a human genome generated using a single

restriction enzyme with a 4 base pair recognition sequence.

Some applications of haplotyping may benefit from combining sequence data derived from different library preparation methods. Fortunately, HapCUT2's flexibility enables haplotype assembly using different sources of data. To demonstrate this, we combined 40× coverage Hi-C data with 10X Genomics linked-read data (34× short read coverage) to assemble haplotypes with 98.9% of variants contained in the largest block for each chromosome (Supplemental Fig A8). The haplotypes were highly accurate, with a switch error rate of 0.0008 and a mismatch rate of 0.003.

2.4 Discussion

We introduced HapCUT2, a maximum likelihood based algorithm for the haplotype assembly problem that extends the original HapCUT method to model technology specific errors and can handle sequence data from diverse technologies efficiently. Using simulated and real WGS data, we demonstrated that HapCUT2 can assemble haplotypes for a diverse array of data modalities while other tools are specialized for certain subsets of data modalities. One of the new features of HapCUT2 is its support for long reads such as those generated by dilution-pool sequencing based methods and long read sequencing technologies such as Pacific Biosciences. Using multiple long read WGS datasets we demonstrate that HapCUT2 obtains higher accuracy than all leading methods, while offering significantly higher speed and scalability. Apart from PacBio, Oxford Nanopore sequencers are also capable of producing long reads, albeit with lower throughput than PacBio sequencers [50]. As current technologies improve and new long read data types continue to emerge, having a fast and flexible tool like HapCUT2 that can efficiently and accurately assemble haplotypes from any type of data is important.

Using simulated data as well as whole-genome Hi-C data, we observed that HapCUT2 and HapCUT were the only computational methods for haplotype assembly that are reasonably

capable of phasing paired-end reads with large insert sizes. We demonstrated that high coverage Hi-C data (e.g., $\sim$80-100$\times$ with a 4-cutter restriction enzyme) can be used to phase $>$ 75-80% of the variants per chromosome with high accuracy. While it was known that Hi-C can be used to link distant variants, our results demonstrate that high resolution whole-chromosome haplotypes can be assembled directly from the sequence reads. In addition, the low rate of pairwise variant error at long genomic distances is a unique feature of the assembled haplotypes and could be useful for applications that require accurate long-range phasing. Although generating Hi-C data requires intact cells, Putnam et al. [51] recently described a proximity ligation method using in-vitro reconstituted chromatin from high-molecular-weight DNA.

HapCUT2 implements an iterative approach for modeling and estimating h-trans error probabilities *de novo* that reduces errors in assembled Hi-C haplotypes compared to HapCUT. We expect that a similar approach could be utilized to improve the accuracy of haplotypes assembled using data from other technologies that exhibit systemic patterns of error, e.g. chimeric fragments present in dilution pool sequencing and reference allele bias in PacBio reads due to alignment ambiguity. In general, the flexibility of the HapCUT2 likelihood model lends itself well to modeling sources of error that result from experimental protocol and design, but are not adequately represented by read quality scores. Another advantage of the HapCUT2 likelihood model and its implementation is the ability to integrate sequence data from diverse methods to generate highly accurate and comprehensive haplotypes for reference human genomes, e.g. NA12878 and other genomes that have been sequenced by the GIAB consortium [48]. We demonstrated this by assembling accurate and complete chromosome-spanning haplotypes for NA12878 by combining Hi-C data with linked-read data.

Similar to the original HapCUT method, HapCUT2 is a heuristic algorithm that iteratively searches for better haplotypes with increasing likelihood using graph-cuts in a greedy manner. Although it provides no performance guarantees on the optimality of the final haplotype assembly, its performance on multiple sequence datasets demonstrates its high accuracy and suggests that

it is able to find haplotypes that are close to the optimum. Further, previous work on exact algorithms for haplotype assembly [40] has shown that the haplotypes assembled using HapCUT are very close to the optimal solution.

Even at high sequencing depth, not all variants on a chromosome can be assembled into a single haplotype block. Using Hi-C data, $\sim 20\%$ of the variants that are at a large distance from cut-sites for a 4 bp restriction enzyme remain unphased. In comparison, long read technologies can phase the vast majority ($> 95\%$) of variants into multiple haplotype blocks for each chromosome (N50 lengths ranging from few hundred kilobases to several megabases). In the absence of additional sequence data, information from population haplotype data can be used to link unphased variants to the chromosome spanning haplotype block in the case of Hi-C [23] and to determine the phase between disjoint haplotype blocks assembled from long read datasets [52]. Recently, a population phasing method, SHAPEIT2, has been extended to incorporate information from haplotype-informative sequence reads in the statistical model for phasing using population haplotypes [53]. Analogous to this, it should be feasible to incorporate information from population haplotypes while assembling haplotypes from sequence reads for an individual in the likelihood based framework of HapCUT2.

Another important consideration for sequencing based haplotype assembly is the source of the variant calls. Most haplotype assembly methods, including HapCUT2, require a set of reliably called variants as input. Illumina short read sequencing at $\sim$30-40$\times$ is considered the de facto standard approach to obtain reliable heterozygous calls [54]. Therefore, the simplest approach would be to perform short read WGS in addition to the sequencing protocol for phasing variants. However, in some cases (e.g. 10X linked-read data [21]), variants can be called directly from the sequence data used for haplotyping, eliminating the need to generate additional sequence data. In principle, it should also be possible to call variants directly from high-coverage Hi-C data before haplotyping. For PacBio sequence data, variant calling is more challenging due to high error rates, but an integrated variant calling and haplotyping approach could potentially work

because haplotype information can be used to distinguish true heterozygous variants from errors.

Finally, all analysis and comparisons of different methods in this paper were performed using SNVs only. Short insertions and deletions (indels) represent the second most frequent form of variation in the human genome and are frequently associated with diseases. Therefore, reconstructing haplotypes that include not only SNVs but also small and large indels is important for obtaining a complete picture of genetic variation in an individual genome. However, the detection and analysis of indels is more challenging compared to SNVs. In principle, HapCUT2 can phase indels along with SNVs. However, it may not be feasible to phase short indels using PacBio reads that have high rate of indel errors. Recently, Patel et al. [55] developed a machine learning method to phase large deletions using Hi-C data. Assessing the capability of different sequencing technologies and protocols for haplotyping all forms of genetic variation is an important topic of future work.

2.5 Methods

The input to the haplotype assembly problem consists of fragments or "reads" from an individual genome that have been aligned to a reference genome with information about alleles (encoded as 0 and 1 for bi-allelic variants) at each heterozygous variant. The heterozygous variants are assumed to have been identified separately from WGS data for the same individual. Haplotype assembly algorithms for diploid genomes aim to either (i) partition the fragments into two disjoint sets such that fragments in each set originate from the same homologous chromosome, or (ii) reconstruct a pair of haplotypes such that the fragments are maximally consistent with the assembled haplotypes. HapCUT belongs to the second type and aims to optimize the Minimum Error Correction (MEC) objective function: the number of allele calls in the fragment matrix that need to be changed for each fragment to be perfectly consistent with one of the two haplotypes [56].

Several algorithms use a probabilistic model for haplotype assembly and attempt to maximize a likelihood function that relates the observed reads to potential haplotypes [57, 58]. ProbHap [43] aims to optimize a likelihood function that generalizes the MEC criteria. Rather than the MEC criterion, HapCUT2 uses a haplotype likelihood model for sequence reads [58].

2.5.1 Haplotype likelihood for sequence reads

Let $H = (H_1, H_2)$ represent the unordered pair of haplotypes where H_1 is a binary string of length n. H_2 is also a binary string of length n. H_2 is the bitwise complement of H_1 if all sites are heterozygous. Consider a collection of reads R, where each read (fragment) R_i is denoted by a string of length n over the alphabet $\{0, 1, -\}$ where $-$ corresponds to heterozygous loci not covered by the read. Given a haplotype h and a fragment R_i, define the delta function $\delta(R_i[j], h[j]) = 1$ if $R_i[j] = h[j]$ and 0 otherwise. Given $q_i[j]$, the probability that the allele call at variant j in read R_i is incorrect, the likelihood of observing read R_i is:

$$p(R_i|q,h) = \prod_{j,R_i[j]\neq -} \delta(R_i[j],h[j])(1-q_i[j]) + (1-\delta(R_i[j],h[j]))q_i[j] \; . \tag{2.1}$$

Extending this to a haplotype pair $H = (H_1, H_2)$, we can define

$$p(R_i|q,H) = \frac{p(R_i|q,H_1) + p(R_i|q,H_2)}{2}, \tag{2.2}$$

assuming equal probability of sampling the read from either haplotype. Then, $P(R|q,H)$, the data likelihood given a pair of haplotypes H, can be computed as a product over fragments (assuming independence of fragments) as:

$$p(R|q,H) = \prod_i p(R_i|q,H)$$

The read likelihood function assumes a simple copying model where the read R_i is copied

from either H_1 or H_2 with zero or more sequencing errors. It can be modified to account for additional types of errors in reads. For example, Hi-C reads can be 'cis' or 'trans' (switch error) and the probability of a read being trans depends on the distance between the two interacting loci captured in a Hi-C fragment. Given a set S of variants, $H(S)$ is defined as the haplotype pair formed by flipping the alleles between the haplotype pair at the variants in the set S. If $\tau(I)$ is the probability that a read is trans, the likelihood of a Hi-C fragment with insert length I is the sum of two terms:

$$p(R_i|q,H,\tau(I)) = (1-\tau(I))p(R_i|q,H) + (\tau(I))p(R_i|q,H(S)) \tag{2.3}$$

where S is the set of variants covered by one end of the Hi-C fragment.

2.5.2 Likelihood-based HapCUT2 algorithm

The original HapCUT algorithm is an iterative method that attempts to find better haplotypes using a max-cut heuristic that operates on the read-haplotype graph. This graph is constructed using the fragments and the current haplotype. The nodes of this graph correspond to variants and edges correspond to pairs of variants that are connected by a fragment. Similar to HapCUT, HapCUT2 also uses a greedy method to find a max-cut in the read-haplotype graph such that the variants on one side of the cut can be flipped to improve the current haplotype. However, it utilizes the likelihood function instead of the MEC score allowing it to account for read quality scores as well as model technology specific errors such as trans errors in Hi-C data.

To describe the new likelihood-based max-cut procedure used in HapCUT2, we define a partial likelihood function that represents the likelihood of the fragments restricted to a subset of variants S as follows:

$$p_S(R|q,H) = \prod_i p_S(R_i|q,H) \tag{2.4}$$

where

$$p_S(R_i|q,h) = \prod_{j \in S} \delta(R_i[j], h[j])(1 - q_i[j]) + (1 - \delta(R_i[j], h[j]))q_i[j] \tag{2.5}$$

The objective of the greedy algorithm for finding the maximum likelihood cut is to find a subset of variants or vertices S such that the haplotype $H(S)$ has better likelihood than the current haplotype H. It starts by initializing the two shores (S_1 and S_2) of the cut using a pair of vertices in the graph and at each step adds a vertex or node to one of the two shores of the cut. Adding a vertex v to S_1 results in a new haplotype $H(S_1 \cup v)$ while adding the vertex to S_2 does not change the current haplotype $H(S_1)$. This vertex v is chosen such that it maximizes the absolute difference between two log likelihoods:

$$L(v) = \log\left[p_S(R|q, H(S_1 \cup \{v\}))\right] - \log\left[p_S(R|q, H(S_1))\right] \tag{2.6}$$

where $S = \{S_1 \cup S_2 \cup v\}$.

In other words, we select the vertex v for which adding it to one side of the cut is significantly better than adding it to the other side of the cut. This process is repeated until all variants have been added to one side of the cut. The Maximum-Likelihood-Cut routine (full description available in Supplemental Methods) considers many possible cuts by initializing each cut using a different edge in the graph and selects the cut that gives the maximum improvement in the likelihood of the haplotypes defined by the cut. This maximum-likelihood-cut heuristic is the core of the HapCUT2-Assemble algorithm outlined below:

Initialization: $H = H^0$

Iteration: until $p(R|q,H)$ stops changing:

1. $S^* = \text{Maximum-Likelihood-Cut}(H, R)$

2. if $p(R|q,H(S^*)) > p(R|q,H)$: $H = H(S^*)$

Return: H

2.5.3 Complexity of HapCUT2

The HapCUT2-Assemble routine is run for T iterations (default value = 10000) on each connected component of the read-haplotype graph. To speed up the convergence, we utilize a convergence criterion wherein components for which there has been no improvement in the likelihood for C iterations (default value = 5) are not analyzed further. A similar convergence criterion is also used for the maximum-likelihood-cut heuristic. In practice, this simple convergence criterion results in considerable improvement in running time compared to HapCUT.

HapCUT2 does not store an edge for each pair of nodes covered by a read because this leads to prohibitive storage requirements for long reads (proportional to V^2 where V is the maximum number of variants per read). Instead, it only stores an edge for adjacent vertices covered by each read. However, it still has to consider all pairs of edges per fragment in order to calculate and update the partial likelihoods. Therefore, the computational complexity of HapCUT2 scales as V^2. The runtime of one iteration of the maximum-likelihood-cut routine is $O(N\log(N) + N \cdot d \cdot V^2)$ where N is the number of variants, d is the average coverage per variant, and V is the maximum number of variants per read. Therefore, the overall runtime of HapCUT2 is $O(T \cdot M \cdot (N \log N + N \cdot d \cdot V^2))$ where T and M are the maximum number of iterations for the HapCUT2-assemble and maximum-likelihood-cut routines respectively.

2.5.4 Estimation of h-trans error probabilities in Hi-C data

In order to properly model h-trans error we must know the probability that a read pair with insert size I is h-trans, i.e. the two ends of the paired-end read originate from different homologous chromosomes. We assume that the h-trans error probability, represented as $\tau(I)$, is

the same for all reads with the same insert length I. If the true haplotypes are known, $\tau(I)$ can be estimated by comparison of all reads with insert length I to the haplotypes and calculating the fraction of reads that are inconsistent with the haplotypes. Because the true haplotypes are unknown, we use an iterative Expectation-Maximization-like approach to directly estimate τ from the data. Initially, the HapCUT2-Assemble routine is used to phase all reads with $\tau(I) = 0$ for all I. The assembled haplotypes H are used to calculate a maximum-likelihood estimate of $\tau(I)$ for each insert size I. Subsequently, the HapCUT2 routine is used to assemble a new set of haplotypes using τ and the Hi-C version of the read likelihood function. This is repeated until the likelihood of the haplotypes does not improve (see Supplemental Methods for more details).

2.5.5 Post-processing of haplotypes

HapCUT2 assumes that the heterozygous sites are known in advance. However, some of the heterozygous sites in the input may actually be homozygous, e.g. due to errors during variant calling or read alignment. In addition, some variants cannot be phased reliably due to low read coverage or errors. Therefore, the accuracy of the final assembled haplotypes can be improved by removing variants with low confidence phasing. HapCUT2 implements a likelihood-based pruning scheme that considers the possible phasings for each variant individually and calculates a bayesian posterior probability for each of the 4 possible configurations (00,11,10,01). If the maximum posterior probability is less than a user-defined threshold (0.8 by default) then the variant is pruned from the output haplotypes (see Supplemental Methods for details). For long read datasets, we also consider the possibility of a switch error between each pair of adjacent variants in a haplotype block and if the posterior probability of the final haplotype configuration is less than a threshold, the block is split at that position. This can reduce switch errors at the cost of reducing the length of haplotype blocks.

2.5.6 Accuracy and completeness of haplotype assemblies

The AN50 metric summarizes the contiguity of assembled haplotypes [59]. It represents the span (in base pairs) of a block such that half of all phased variants are in a block of that span or larger. To adjust for unphased variants, the base pair span of a block is multiplied by the fraction of variants spanned by the block that are phased. For Hi-C data, we assessed the completeness of the haplotypes on a chromosome-wide scale by using the fraction of variants in the largest (also called most-variants-phased or MVP) block [23]. The accuracy of a haplotype assembly is typically assessed by comparing the assembled haplotypes to 'truth' haplotypes and calculating the switch error rate [41, 43]. A "switch error" (also known as long switch) occurs when the phase between two adjacent variants in the assembled haplotypes is discordant relative to the truth haplotypes. Two consecutive switch errors correspond to the flipping of the phase of a single variant and were counted as "mismatch" (also known as short switch) errors instead of two switch errors.

For many applications of haplotyping, the ability to determine the phase between a pair of heterozygous variants is important. To assess the pairwise accuracy of the haplotypes, we utilized a pairwise phasing accuracy metric where all pairs of phased variants in a block were classified as concordant (1) or discordant (0) (by comparison to the gold-standard haplotypes) and the accuracy was defined as the fraction of discordant pairs among all pairs with the same genomic distance [30].

2.5.7 Long read datasets and haplotype assembly tools

Haplotype fragment files corresponding to the whole genome fosmid sequence data (32 pools) for NA12878 [20] were downloaded from `http://www.molgen.mpg.de/~genetic-variation/SIH/data/` Aligned PacBio SMRT whole genome read data for NA12878 [48] was obtained from the GIAB ftp site: `ftp://ftp-trace.ncbi.nlm.nih.`

gov/giab/ftp/data/NA12878/NA12878_PacBio_MtSinai Haplotype fragments for phasing were extracted from the sorted BAM files using the extractHAIRS tool (see Supplemental Methods). Aligned 10X data for NA12878 [48] was also obtained from the GIAB ftp site: ftp://ftp-trace.ncbi.nlm.nih.gov/giab/ftp/data/NA12878/10XGenomics. Molecule boundaries were called when the distance between two consecutive reads with the same barcode exceeded 20 kb, and haplotype fragments were generated for each molecule using the extractHAIRS tool (see Supplemental Methods). The software tools RefHap and ProbHap were downloaded from the authors' websites. For FastHare, we used the implementation of Duitama et al [20]. Default parameters were used for each tool except for HapCUT which was run with memory reduction heuristics to enable it to generate results for comparison.

2.5.8 Variant calls and haplotypes for NA12878

The NA12878 trio haplotypes from the 1000 Genomes project [20] were used as truth haplotypes for assessing the accuracy of all haplotype assemblies. Only single nucleotide variants (SNVs) were considered for phasing. The variants from this dataset (aligned to hg18) were used for phasing the fosmid dataset. For usage with the PacBio, 10X Genomics, and Rao Hi-C data, the hg18 NA12878 VCF file was carried over to hg19 with CrossMap [60].

2.5.9 Alignment and processing of Hi-C data

Two sets of Hi-C read datasets for NA12878 from Rao et al. [49] were used: one containing all primary and replicate experiments using the restriction enzyme MboI (total of $\sim 395\times$ coverage) and another containing all experiments performed with the restriction enzyme HindIII (total of $\sim 366\times$ coverage). The paired-end reads were mapped as single reads to the reference human genome (hg19) using BWA-MEM[61]. To handle reads that contain the ligation junction for the Hi-C fragments, we developed a post-processing pipeline (see Supplemental Methods)

to generate sorted BAM files that were used for haplotyping. Read pair information from intra-chromosomal read pairs with an insert size greater than 40 Mb were not used to avoid linkages with excessively high h-trans error rates. To subsample datasets to lower coverage, fragments were randomly sampled from the aggregate dataset with the appropriate frequency.

2.5.10 Read simulations

Haplotypes were simulated by randomly introducing heterozygous SNVs at a uniform rate of 0.0008 in a genome of length 250 megabases. Each heterozygous SNV is assigned a random allele $\in \{0,1\}$ for haplotype H_1, with H_2 assigned to the complement. Reads of a given length were generated by selecting the start position randomly and the corresponding haplotype fragment was obtained by appending all overlapping alleles from one of the two haplotypes. Base miscalls were introduced in the reads with probability 0.02, resulting in an allele flip with probability $\frac{1}{3}$ or an uncalled SNV otherwise (to represent miscalls to non-reference, non-alternate calls). Hi-C-like reads of length 150 base pairs were simulated in pairs. The insert length of each read pair was sampled from the uniform distribution (minimum value of 0 and maximum value equal to the maximum insert length).

2.6 Software availability

HapCUT2 is available for download at https://github.com/vibansal/HapCUT2 and also from Supplementary Materials.

2.7 Disclosure declaration

V. Bafna is a co-founder, has an equity interest, and receives income from Digital Proteomics, LLC. The terms of this arrangement have been reviewed and approved by the University

of California, San Diego in accordance with its conflict of interest policies. DP was not involved in the research presented here.

2.8 Acknowledgments

V. Bansal was supported in part by a grant from the NIH (HG007430). V. Bafna and P. Edge were supported in part by grants from the NIH (HG007836 and GM114362) and NSF (DBI-1458557, IIS-1318386).

Chapter 2, in full, is a reprint of the material as it appears in HapCUT2: robust and accurate haplotype assembly for diverse sequencing technologies. Edge, Peter; Bafna, Vineet; Bansal, Vikas. Genome Research, 27(5), pp.801-812. Cold Spring Harbor Laboratory Press, 2017. The dissertation author was the primary author of this paper.

2.9 Tables

Table 2.1: Comparison of the approach, time complexity, and applicability of five algorithms for haplotype assembly: HapCUT2, HapCUT, RefHap, ProbHap and FastHare. R denotes the number of reads (all algorithms process reads for each haplotype block separately), N denotes the total number of variants, V denotes the maximum number of variants in a read, and d is the maximum read depth per site. d' is the maximum number of reads crossing a site (equivalent to d except with paired-end inserts being included as part of the read). c_1, c_2, and c_3 represent method-specific variables that are either fixed in advance or selected by the user. Reads are assumed to be sorted by starting position.

Method	**Approach**	**Complexity**	**Long reads**	**Hi-C support**	**Variant pruning**
HapCUT2	likelihood optimization using graph-cuts	$O(c_1c_2(N\log(N)+NdV^2))$	scalable	yes	likelihood based
HapCUT	MEC optimization using graph-cuts	$O(c_1c_2(N\log(N)+NdV^2))$	high memory requirements	yes	no
RefHap	Max-cut on read graph	$O(c_3(R^2Vd'+RV^2d'^2))$	low-to-medium coverage	no	discrete
ProbHap	exact likelihood using dynamic prog. + merging heuristic	$O(Nd'2^{d'})$	low-coverage	no	confidence scores
FastHare	read partitioning optimization	$O(RVd')$	yes	no	discrete

Table 2.2: Comparison of total runtime (hours:minutes, summed across all chromosomes) for different haplotype assembly methods on various sequence datasets for NA12878. For each dataset, only methods that produced results within 20 CPU-hours per chromosome are shown.

	HapCUT2	**HapCUT**	**RefHap**	**ProbHap**	**FastHare**
Fosmid	1:09	1:49	0:01	0:31	0:01
PacBio (11×)	0:52	1:45	0:25	52:32	0:02
PacBio (44×)	4:05	6:56	215:53	-	0:20
10X Genomics	1:55	16:50	-	-	12:07
Hi-C (40×)	4:38	0:46	-	-	-
Hi-C (90×)	9:11	1:49	-	-	-

2.10 Figures

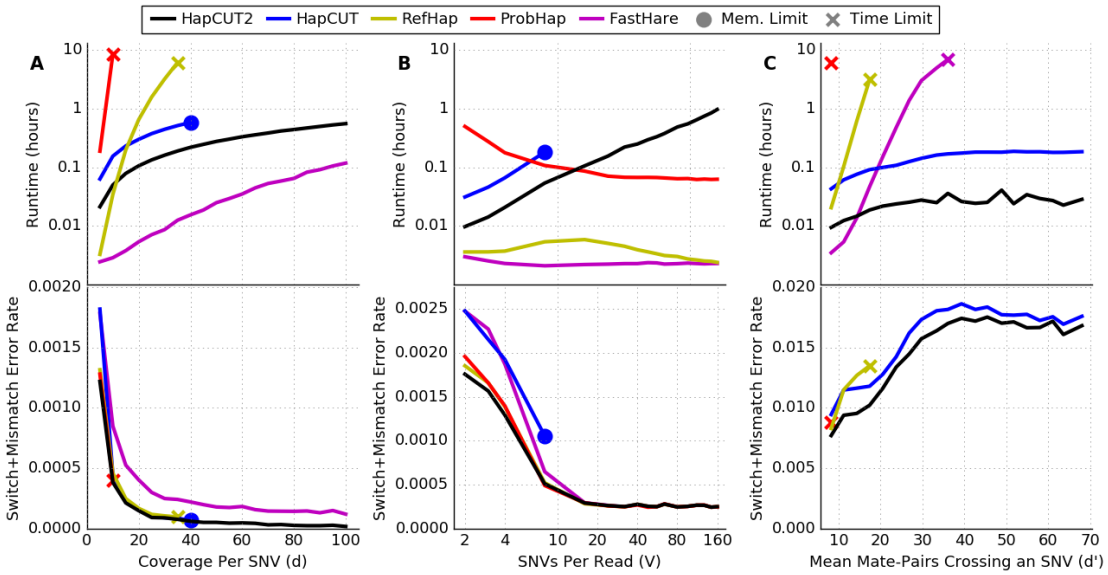

Figure 2.1: Comparison of runtime (top panel) and switch+mismatch error rate (bottom panel) for HapCUT2 with four methods for haplotype assembly (HapCUT, RefHap, ProbHap, and FastHare) on simulated read data as a function of (A): mean coverage per variant (variants per read fixed at 4), (B): mean variants per read (mean coverage per variant fixed at 5), (C): mean number of paired-end reads crossing a variant (mean coverage per variant fixed at 5, read length 150 base pairs, random insert size up to a variable maximum value). Lines represent the mean of 10 replicate simulations. FastHare is not visible on panel C (bottom) due to significantly higher error rates.

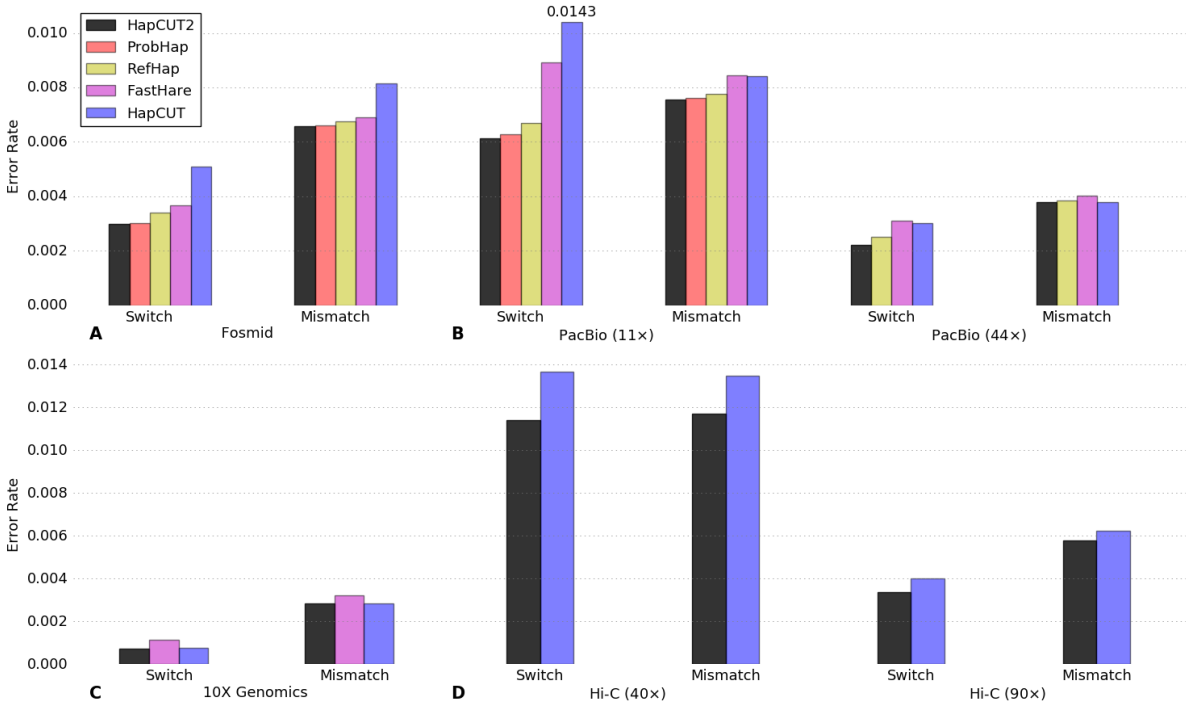

Figure 2.2: Accuracy of HapCUT2 compared to four other methods for haplotype assembly on diverse whole-genome sequence datasets for NA12878: (A) fosmid dilution pool data [20], (B) PacBio SMRT data(11× and 44× coverage), (C) 10X Genomics linked-reads, (D) Whole-genome Hi-C data (40× and 90× coverage, created with MboI enzyme). Switch and mismatch error rates were calculated across all chromosomes using the subset of variants that were phased by all methods. For each dataset, only methods that produced results within 20 CPU-hours per chromosome are shown.

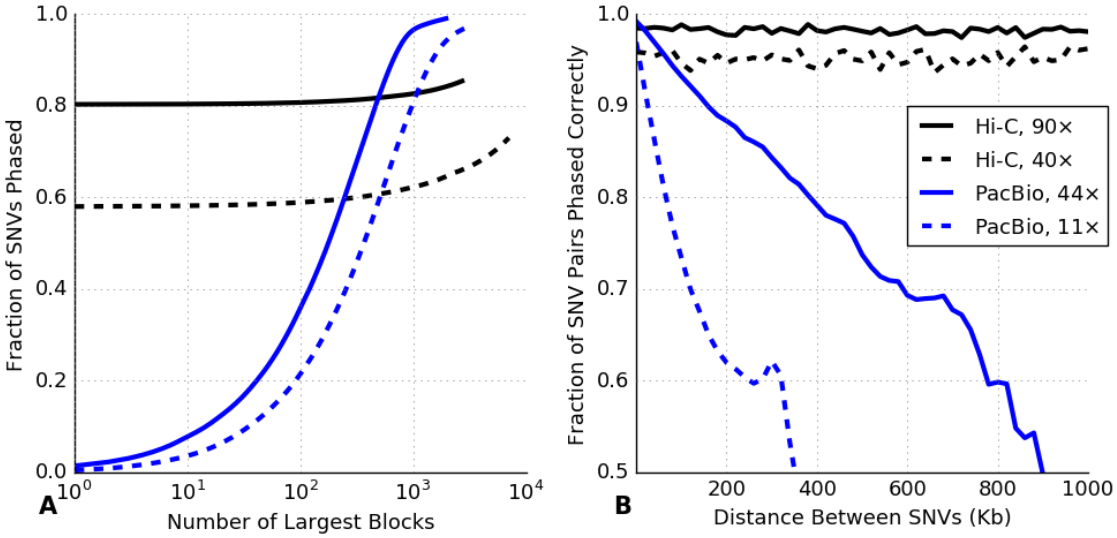

Figure 2.3: Haplotype completeness and accuracy compared between Hi-C (MboI enzyme, 90× and 40× coverage) and PacBio SMRT (44× and 11× coverage). (A) Cumulative measure of the fraction of variants phased within a given number of the largest haplotype blocks. (B) Fraction of correctly phased variant pairs as a function of distance.

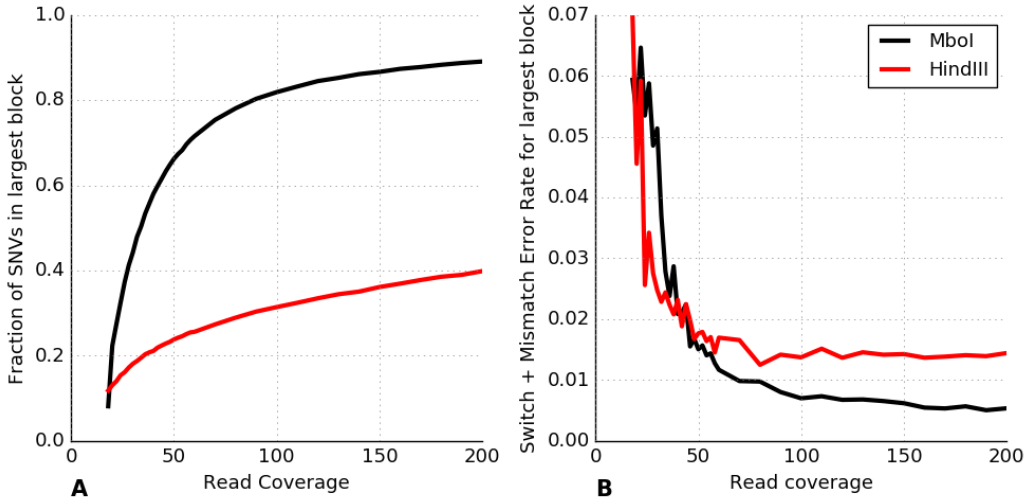

Figure 2.4: Improvements in the (A) completeness and (B) accuracy (switch + mismatch error rates) of the largest haplotype block with increasing Hi-C sequencing coverage for two different restriction enzymes: MboI and HindIII. Results are presented using data for chromosome 1 with coverage ranging from 18× to 200×.

Chapter 3

Computational techniques for highly accurate variant calling and haplotyping of single human cells

3.1 Abstract

There is widespread interest in sequencing the genomes of individual human cells for clinical and scientific purposes. However, single-cell sequencing usually requires amplification of the source DNA with a DNA polymerase, and errors from replication result in a significant amount of false positive variant calls. Recently, a new method called SISSOR (single-stranded sequencing using microfluidic reactors) for single-cell sequencing was described that uses a microfluidic processor to separate the positive and negative strands of megabase scale DNA fragments for amplification in separate reaction chambers before barcoding and short read sequencing. The redundancy that is afforded by sequencing the strands separately enables the removal of amplification errors, observed as differences between the strands. SISSOR also enables the construction of megabase-scale haplotypes using the single stranded fragments. The

data produced using the SISSOR method introduces unique computational challenges that are not met by existing variant calling tools, including unique error modalities. We introduce a single nucleotide variant calling algorithm uniquely designed for the SISSOR method. Given SISSOR data for multiple single cells, the algorithm calculates the probability that a variant is present by considering all possible ways that single DNA strands may have been distributed amongst reaction chambers while being subject to multiple sources of error. The algorithm enables SISSOR to achieve error rates orders of magnitude lower then the amplification error rate. Further, haplotype assembly of the single-stranded fragments enables direct error correction between the positive and negative strands for error rates as low as $1 * 10^{-8}$.

3.2 Introduction

Single nucleotide variants (SNVs) are the most common form of human genetic variation, and detection of SNVs is crucial for scientific and clinical purposes [7]. It is now routine to detect SNVs genomewide via whole-genome resequencing of a sample of cells. Usually this is performed with a bulk tissue sample and as a result information about cell-to-cell variations is lost [62]. Also, this approach requires tens of thousands of cells per sample. There are many applications of DNA sequencing where it is not feasible to obtain a sample of this size, such as profiling circulating tumor cells or in preimplantation genetic diagnosis [15, 16]. Furthermore, single-cell sequencing enables the sequencing of individual cells in samples such as tumors that have significant genomic heterogeneity [17].

Because short read sequencing technologies require a large sample of source DNA, single-cell sequencing is typically performed by whole genome amplification with a technique such as multiple displacement amplification (MDA). The error rate of conventional MDA followed by sequencing is approximately $1.2 * 10^{-5}$, resulting in tens of thousands of false positive variant calls per genome and obscuring the true cell-specific variations [18]. Further, conventional short

read sequencing provides limited information about haplotypes, or the ordering of alleles on homologous chromosomes. Recently, a new method for single-cell sequencing was introduced called SISSOR that aims to address these problems [24]. SISSOR uses a microfluidic device to separate the single strands of DNA and partition them in separate reaction chambers for amplification, after which they are barcoded and sequenced. The sequenced strands of DNA are haploid and can be used to assemble long haplotypes similar to other partitioning and synthetic long read approaches [21]. A schematic of the SISSOR device is in figure 3.1. A microfluidic device is used to capture a single cell, which is lysed and the DNA is separated into single strands. The single strands are mixed in a ring mixer and randomly distributed to 24 reaction chambers, where they are amplified with MDA. The products in the individual chambers are bar coded and sequenced. The independent amplification of the two single strands provides redundancy so that errors occurring on one strand or the other can be removed.

Traditional SNV calling algorithms such as samtools, GATK, or FreeBayes are designed to call variants using sequencing reads from a single organism or a population of multiple organisms [63, 64, 65]. These algorithms do not account for error modalities that occur in single-cell sequencing. There have been algorithms designed specifically for single-cell sequencing, including Monovar and SCcaller [66, 67]. However, neither of these methods are ideal for calling variants using sequencing libraries derived from randomly partitioned single strands from each haplotype. We introduce a novel algorithm for calling variants designed and optimized for the SISSOR method to enable highly accurate variant calls. The algorithm models all of the possible ways that single stranded DNA from each haplotype might have been distributed amongst all of the reaction chambers, accounting for sources of error such as amplification error and overlap of different haplotypes in the same chamber. We also describe strategies for processing the sequenced single strands for optimal haplotype assembly, and show that even greater accuracy can be achieved by directly correcting errors between haplotype-assigned strands.

3.3 Results

3.3.1 SNV calling algorithm

The goal of the SISSOR SNV calling algorithm is to determine the SNV alleles present for each site in the genome, given SISSOR aligned read data (24 reaction chambers) for one or multiple individual cells. It is assumed that the input data for each variant site is $24n$ read base pileups, where n is the number of sequenced cells. In other words, the input data is the observed DNA bases for the variant site across all the reaction chambers, as well as quality information.

In the absence of DNA strand loss or errors from sources such as sequencing or amplification, it is trivial to determine all of the alleles present (the full genotype) for each site. For each cell, exactly 4 of the 24 chambers should have read base observations. If the individual is homozygous for an allele, then the read bases in all four chambers should be the same. If the individual is heterozygous for an allele, the bases in two chambers should match while the bases observed in the other two chambers should match. However, there is significant DNA strand loss in the current implementation of the SISSOR method which means that most sites will have anywhere from 0 to 4 chambers with observations, and it is not known which haplotypes the observations belong to. Further, it is possible for strands to be placed in the same chamber, which results in a mixture of DNA bases observed in that chamber at heterozygous sites. There are also errors from amplification, which can cause a subset of bases observed in a single chamber to be changed to a random DNA base.

To address these problems, the algorithm uses a probabilistic model for the SISSOR experimental protocol that considers all possible ways that single DNA strands could be arranged to reaction chambers and subjected to errors (Figure 3.2). It is assumed that the locus has a diploid source genotype, which results in four single strands (two strands from each haplotype) after strand separation. After strand separation, the strands could be distributed to any of the 24 chambers, or be unsampled. The model accounts for the fact that multiple strands could overlap

in the same chamber, and that there could be errors during amplification. Given this model, the algorithm accounts for all possible ways that the strands might have been distributed amongst the reaction chambers and subjected to the multiple types of errors. A bayesian calculation is used to calculate the posterior probability of a genotype or the presence of an allele given the data and model. The algorithm also supports calling variants using data for multiple cells, in which case all possible ways of distributing strands and applying error sources in all of the available cells is calculated.

The algorithm was applied to data from 3 single cells sequenced with the SISSOR method from the PGP1f fibroblast cell line, and compared against a highly confident reference for PGP1 (see methods). The results are in table 3.1. At the most lenient threshold, 1.7 million SNVs were called with a false-positive rate of $5 * 10^{-5}$. At a moderate threshold, 613,669 SNVs were called with a false-positive rate of $1 * 10^{-6}$. At the strictest threshold, 177,096 SNVs were called with a false-positive rate of $1 * 10^{-7}$. For comparison, the base error rate of amplification and sequencing for these samples was estimated to be $1.7 * 10^{-5}$, so using the variant calling algorithm to leverage multiple-chamber information can result in orders of magnitude better accuracy [24].

3.3.2 Haplotype Assembly

Accurate assembly of haplotypes requires a set of haploid sequence reads or fragments, and an accurate set of known heterozygous SNV sites. A set of heterozygous SNVs obtained from 60× WGS sequencing of the PGP1f cell line were used for haplotype assembly (ENCODE project "ENCSR674PQI"). SISSOR fragments are observed as sets of aligned reads clustered in the same region of the reference genome. Normally, a cluster of reads like this are sampled from a single haploid fragment originating from a single strand of DNA. For this reason, the boundaries of SISSOR fragments were drawn using a read coverage based Hidden Markov Model [24]. The variant calling algorithm used for calling consensus variants was extended to determine the most likely sequence of bases for each haploid fragment in each chamber. Then, the HapCUT2

method was used to assemble haplotypes for the $\sim$1.2 million SNV sites overlapping SISSOR fragments [68]. The haplotype N50 was $\sim$15 Mb and the total error rate (switch+mismatch errors) was $\sim$1.5%.

However, when single-stranded fragments aligning to the same region of the reference genome but from different haplotypes occur in the same amplification chamber, it can result in a hybrid fragment of mixed ploidy. If there is extensive overlap between the two fragments, the resulting hybrid fragment may contain numerous heterozygous variant sites with observations of both alleles. If there is limited overlap between the fragments, then the haplotype phase may "switch" from one haplotype to the other at a single point in the fragment with no such signal. Two general strategies were used to minimize errors in haplotype assembly from this problem (Figure 3.3). First, the SISSOR haplotype fragments were split at the sites of detectable switch errors, where a single haplotype fragment has a switch error inconsistency with multiple other haplotype fragments in the same location. Secondly, the haplotype fragments were split at clusters of mixed allele calls (more than 25% of allele calls having a mixed allele in a span of ≥ 3 heterozygous SNV locations), which are likely to indicate an overlapping region between two haploid fragments. Fragments were completely removed if they were found to be low-quality by these two metrics, i.e. having excessively high switch error rate compared to other fragments or having excessively many mixed alleles.

This strategy greatly improved the haplotype accuracy, for a trade-off in haplotype contiguity (Figure 3.4). The haplotype N50 reduced from $\sim$15 Mb using unprocessed fragments to $\sim$7 Mb using processed fragments, but the haplotype switch error rate dropped from 0.009 to 0.004 and the mismatch error rate dropped from 0.006 to 0.003. Note that these error rates are upper bounds; the haplotypes were compared to haplotypes assembled using BAC sequencing that have non-negligible errors [44].

3.3.3 Strand-to-strand matching for improved SNV accuracy

The variant calling algorithm enables more accurate single-cell sequencing variant calling by comparing read base observations between chambers and between cells. However, the SISSOR method can enable even greater accuracy when it is possible to compare directly between the two single strands in a sample. Amplification errors occur separately in the independent reaction chambers, so most amplification errors will be observed as differences between the two independently amplified and sequenced strands and can be removed. However, this type of error removal can only be performed between two single strands from the same DNA molecule (haplotype). This was addressed by assigning SISSOR fragments back to the assembled haplotypes. Fragments matching the source haplotype with 80% accuracy were considered to be assigned and the remaining fragments were discarded.

The strand-to-strand consensus strategy was used to call variants in each of the 3 sequenced cells separately (table 3.3). The total number of positions called ranged from 30 to 70 million, and the number of SNVs called per cell ranging from 23477 to 54832. The calls were compared to a reference for PGP1 obtained by intersecting high-confidence complete genomics (CGI) and high-coverage WGS calls, plus calls from BAC libraries for PGP1 (see methods). Variants confirmed by a third independent chamber outside of the strand-matched chambers were also considered unlikely to be false positives. By these metrics, the upper bound of the error rate for SISSOR was found to be $2.63 * 10^{-7}$, $6.50 * 10^{-7}$ and $2.97 * 10^{-7}$ for the three cells. This error rate includes actual errors as well as cell-specific variations.

Real cell-specific variations may contribute significantly to the measured error rates. Therefore, a proxy argument was used to measure a more accurate upper bound of the SISSOR error rate. The same strand-to-strand consensus strategy was used between haplotype-matched strands from different cells (cross-cell), so that the strand-to-strand matching would remove real cell-specific variants as well as the errors in the sequenced single strands. When this analysis was performed, 355 million cross-cell matched positions were called, and there were 9 differences to

the high-quality PGP1 reference. 5 of those differences were confirmed by a 3rd chamber, leaving 4 positions that are possible errors. This upper bounds the SISSOR error rate, after accounting for cell-specific variations, to $1.14 * 10^{-8}$.

3.4 Discussion

We have introduced a novel variant calling method designed specifically for the SISSOR method that probabilistically models the SISSOR experimental workflow. The approach demonstrates significantly improved accuracy over the baseline error rate of MDA based sequencing, and using haplotype-based strand-to-stand consensus analysis improves the accuracy even further. In the future, this approach could be improved in several ways.

Firstly, the existing algorithm considers variant sites independently rather than considering sites together in haplotypes. While the early SNV calling algorithms for short reads considered variant sites independently, the state-of-the-art algorithms for short read variant calling now utilize haplotype information as much as possible [64, 65]. If the algorithm were extended to consider and model the random distribution of haplotype chunks to the different chambers rather than SNVs only, it could greatly improve variant calling accuracy. This would essentially bring the consideration of haplotype information directly into the main algorithm, rather than calling variants first and performing secondary analysis of haplotype assembly and haplotype-based error removal.

Secondly, the existing algorithm considers the entire space of possible strand configuration events, with minimal pruning or optimizations in place. As a result, the existing implementation requires thousands of CPU hours to process whole genome data across three single cells. The algorithm could be redesigned to prune or approximate the probabilities for the strand-to-chamber configurations that are highly unlikely. Optimizations similar this would be crucial if the algorithm is extended to integrate haplotype information. A naive implementation that directly replaces

single-base alleles with haplotype chunks would result in an exponential increase in runtime with the length of haplotypes.

Finally, haplotype assembly with SISSOR could be improved with more sophisticated methods for delineating separate fragments, such as the FragmentCut algorithm [69]. For this work, a simple Hidden Markov Model based on read coverage was used to call fragment boundaries. However, read coverage after amplification with MDA is highly nonuniform [70]. A method that models the read coverage and combines it with signals such as mixed-allele observations could result in fewer hybrid fragments and better haplotype accuracy.

3.5 Methods

3.5.1 SNV calling algorithm

SNV calling overview

The goal of SNV calling with SISSOR data is twofold: first, to determine the best consensus call (SNV or reference) for every genomic position, given read data for every chamber, second, to determine the best call for each individual SISSOR chamber in light of information from the other chambers. For instance, if the same SNV is observed in multiple chambers, then the confidence of the SNV call in each chamber is higher than if the SNV were only observed in one chamber. Similarly, the confidence for that SNV in the consensus genotype over all chambers is higher if it is observed multiple times. In general, a group of reads observed in a chamber at a genomic position is generated from one of four strands: the forward or reverse strand from one haplotype, or the forward or reverse strand from the other haplotype (hereafter referred to as strands 1, 2, 3 and 4, respectively). The variant caller accounts for multiple sources of errors besides sequencer error, including error introduced during MDA from the Phi29 enzyme, and the occurrence of multiple source DNA strands being amplified in the same chamber. Given sets

of read observations from different chambers, the variant caller considers every possible way in which the four single strands of DNA for each genotype could be distributed across chambers. Given multiple single-cell libraries, it considers all combinations of events in each cell that could result in the combined set of data. These events are modeled in a likelihood framework to make a bayesian calculation for the posterior probability that a SNV is present. The variant caller is implemented in python and takes as input a multi-sample pileup of all the chamber bam files, generated with samtools mpileup [63]. The caller makes the following primary assumptions: reads are correctly mapped, variant calls at different genomic positions are independent, and the genotypes of each cell in the multiple-cell case are the same. To make use of reads amplified from strands smaller than the 60 kb HMM window sizes, all chambers with read coverage ≥ 3 were considered in the model (and genomic positions with more than 4 such chambers in a cell were left uncalled, in keeping with the diploid model). The following sections describe the consensus SNV calling model.

SNV calling parameters

The SNV caller models the experimental workflow of the SISSOR method. As such, it requires knowledge of various library-specific parameters. We estimated parameters for the model either empirically from the data or based on prior studies. The prior probability of a genotype, $P(G)$, was estimated using the method described by Li et al [8]. We refer to the set of genotype priors as P_G.

We denote the probability of sampling a fragment from a given chamber i as $P_s[i]$. $P_s[i]$ was assumed to be proportional to the relative genomic coverage of a chamber:

$$P_s[i] = (1 - P_s[\varnothing]) * \frac{cov(i)}{\sum_{j \in 1..24} cov(j)} \tag{3.1}$$

$P_s[\varnothing]$ represents the probability that a strand is not sampled, and was estimated to be consistent with the distribution of strand depth (the number of chambers at a given position

containing fragments. Let $S[i]$ be the number of genomic positions with exactly i chambers with reads. Let the 4-tuple $C = (c_1, c_2, c_3, c_4)$ represent a *chamber configuration* of 4 distinct DNA strands to chambers, with $c_1, c_2, c_3, c_4 \in \{1, 2, ..., 24, \varnothing\}$. Let $\breve{\mathrm{C}}$ refer to the set of all possible configurations, and let $\breve{C}_i \in \breve{C}$ be the set of chamber configurations such that exactly i of (c_1, c_2, c_3, c_4) are not equal to $\varnothing$. If we assume that strand coverage per position results from independent trials depending only on the overall probability of drawing exactly that many strands, we can describe a likelihood for the observed strand coverages:

$$P(S|P_s) = \prod_{i=0}^{4} \left(\sum_{c \in \breve{C}_i} P_s[c_1] * P_s[c_2] * P_s[c_3] * P_s[c_4] \right)^{S[i]} \tag{3.2}$$

We selected $P_s[\varnothing]$ 0.81 as the approximate value that maximizes this likelihood, and this result was consistent with estimates based on the difference of the total coverage from theoretical perfect 4-strand coverage.

We use ε to refer to the probability of error in base-calling. It is described as a constant variable for simplicity but in general represents the per-base quality score of a read position. However, errors can also be introduced as a result of MDA. We use $P_m[x]$ to denote the probability that x fraction of reads in a chamber are noise of a minority base resulting from MDA amplification. Assuming the X chromosome should be monoallelic except for MDA error (the PGP1f cell line is male), we estimated $P_m[x]$ as the distribution of the fraction of the second-most-common allele for each position on the X chromosome. Although this accounts for noise from secondary bases due to MDA, it is known that the consensus error rate from MDA is on the order of $1 * 10^{-5}$ [71]. We let $\omega = 1 * 10^{-5}$ represent the probability that the majority base in a chamber (the consensus) was changed as a result of MDA amplification.

We use $P_p[x]$ to denote the probability that x fraction of reads in a chamber originate from a given parent. $P_p[x]$ accounts for the possibility of strands from different haplotypes occuring at the same position in the same chamber. P_p was estimated using the logic that the fragments

of the hemizygous X chromosome can be shuffled to random positions to simulate a separate homologous chromosome. By overlapping the original fragments with the shuffled fragments we can simulate strand overlaps in a diploid case. With this in mind, we sampled coverages x_1 and x_2 of independent random positions from the X chromosome $1 * 10^8$ times, and used the distribution of the value $\frac{x_1}{x_1+x_2}$ as an estimate of P_p. In the following formulation, we use $\pi = \{P_G, P_s, \varepsilon, P_m, \omega, P_p\}$ to refer to the entire collection of parameters.

SNV calling framework

We begin by considering a single genomic position, with some number of observed reads aligned to the position in each of 24 chambers. Let a_i represent the pileup of observed bases in chamber i, and $a_{i,j}$ denote the base $\in \{A,C,G,T\}$ observed in chamber i at the j-th read (in chamber is base pileup). The set of observed data is $A = [a_1, a_2, ..., a_{24}]$. Let G denote the true genotype of the individual at the site, so G can be homozygous in the reference or alternate allele, or heterozygous. Using Bayes rule, we can compute the posterior probability of a specific genotype in terms of the probability of the data given each genotype:

$$P(G|A,\pi) = \frac{P(A|G,\pi)P(G))}{\sum_G P(A|G,\pi)P(G)} \tag{3.3}$$

Because of DNA strand loss and uneven amplification, the data for many positions may be insufficient to assign a diploid genotype even though it is highly likely that a specific allele is present. For this reason, we computed the probability that each allele $\alpha \in \{A,C,G,T\}$ is present:

$$P(\alpha|A,\pi) = \sum_{G,\alpha\in G} P(G|A,\pi) \tag{3.4}$$

Likelihood of data in all chambers

In order to calculate $P(A|G,\pi)$, it is necessary to account for every configuration in which the 4 strands can be distributed amongst 25 chambers (treating $\varnothing$, or unsampled, as a 25th

chamber). Let $C = (c_1, c_2, c_3, c_4)$ be the chambers corresponding to the four strands where c_i can take values from $1 - 25$. We can compute the probability of this configuration as the product of the probabilities of sampling strands in those chambers:

$$P(C) = \prod_{i \in C} P_s[i] \tag{3.5}$$

Given SISSOR read data A for a single position, Let $N(A) \subseteq \breve{C}$ denote the set of configurations that could have generated A with non-zero probability. Then,

$$\begin{aligned} P(A|G,\pi) &= \sum_{C \in N(A)} P(A|C,G,\pi) P(C|G) && (3.6) \\ &= \sum_{C \in N(A)} \left(\prod_{i=1}^{24} P(a_i|C,G,\pi) \right) P(C|G) && (3.7) \end{aligned}$$

In the case of multiple cells, the data from different cells is independent conditional on the genotype G. To generalize to multiple (in our case) cell samples, we change A to be of length $24n$ and refer to the observed data across all $24n = 72$ chambers. We redefine $\breve{C}$ for multiple cells as the nth Cartesian power of $\breve{C}$ in the single-cell case, or the set of unique n-tuples of 4-tuples that combines one single-cell strand configuration of each cell. The probability of an n-cell configuration is equal to the product of the constituent single-cell strand configurations probabilities.

Likelihood of data in one chamber

Now we consider how to calculate $P(a_i|C,G)$, or the likelihood of the observed chamber data (read bases) given the genotype and knowledge of which strands are present (strand configuration). Let $g_1, g_2 \in \{A,C,G,T\}$ denote the allelic values of G currently being considered. First,

we define the probability of seeing a read base $a_{i,j}$ given that it originated from genotype allele g:

$$P(a_{i,j}|g,\pi) = \begin{cases} (1-\varepsilon) & if a_{i,j} = g \\ \varepsilon & \text{otherwise} \end{cases} \tag{3.8}$$

We address each possible case for chamber-strand configurations separately. We use K_1 to represent the set of configurations in which 1 strand falls into chamber i. We use K_2 to represent the set of configurations in which 2 or more strands fall into chamber i, and more than 1 distinct allele is present. In the case where there is only one strand allele present, we take the product of the probabilities of observing each base $a_{i,j}$ given that the true allele is g. g refers to the allele of G that is present in chamber i as a result of configuration C_i. We assume that MDA error changes the majority allele from g to a different base b (consensus allele) with probability ω. We assume that MDA also introduces noise bases of a base $\sim b$ with probability $\frac{j}{n}$, and otherwise the base is b with probability $\frac{n-j}{n}$. $P_m[\frac{j}{n}]$ is the probability that j of the n bases are noise.

$$\Omega[b] = \begin{cases} 1-\omega & if b = g \\ \frac{\omega}{3} & otherwise \end{cases} \tag{3.9}$$

$$P(a_i|C \in K_1, G, \pi) = \sum_{b \in \{A,C,G,T\}} \Omega[b] \sum_{j=1}^{n} P(a_i|b,\pi,\text{noise} = j) P_m[\frac{j}{n}] \tag{3.10}$$

To compute the probability of chamber data given that the consensus allele is b, we sum over all possible proportions of allele mixture from MDA:

$$P(a_i|b,\pi,\text{noise} = j) = \prod_{k=1}^{n} \left(\left(\frac{j}{n} \right) P(a_{i,k}|\sim b,\pi) + \left(\frac{n-j}{n} \right) P(a_{i,k}|b,\pi) \right) \tag{3.11}$$

We now consider the case where there are multiple strands with different alleles occurring in the same chamber. This is modeled similarly to MDA noise. To compute the total likelihood of

an allele call c_i given a genotype, we sum over all possible proportions of strand mixture, with j representing reads originating from parental allele 1 and $n-j$ representing reads originating from parental allele 2:

$$P(a_i|C \in K_2, G, \pi) = \sum_{j=1}^{n} \left[P_p \left[\frac{j}{n} \right] \prod_{k=1}^{n} \left(\left(\frac{j}{n} \right) P(a_{i,k}|g_1, \pi) + \left(\frac{n-j}{n} \right) P(a_{i,k}|g_2, \pi) \right) \right] \quad (3.12)$$

The $P_p[\frac{j}{n}]$ term accounts for the probability of occurrence of a parental allele in the given fraction. k represents the current index in the set of base calls for chamber i. The term $\left(\frac{j}{n}\right) P(a_{i,k}|g_1, \pi)$ represents the case that $a_{i,k}$ was generated by g from strand 1 (probability $\frac{j}{n}$ that $a_{i,k}$ came from strand 1, times $P(a_{i,k}|g_1)$ the probability of allele $a_{i,k}$ given that it came from g_1). The next two terms represent analogous information for the case that $a_{i,k}$ came from strand 2. To reduce computation, we assume that MDA noise and strand overlap do not occur in the same chamber. Computation was further minimized by constraining the domains of P_m and P_p to ≤ 20 evenly spaced bins.

Likelihood of an allele in a chamber

Along with a consensus call across many chambers, we also called the most likely allele occurring in a specific chamber, in light of information from other chambers. This is done in a similar fashion to computing the most likely genotype G. Consider a single chamber i for which we want to determine the allele present (if any) on the original strand. We denote the assignment of an allele $\in \{A, C, G, T\}$ to chamber i as α_i. We want to choose the most likely α_i:

$$\max_{\alpha} P(\alpha_i|A, \pi) \quad (3.13)$$

As before, we use Bayes rule:

$$P(\alpha_i|A,\pi) = \frac{P(A|\alpha_i,\pi)P(\alpha_i)}{\sum_{\hat{\alpha}_i} P(A|\hat{\alpha}_i,\pi)P(\hat{\alpha}_i)} \tag{3.14}$$

Computing $P(A|\alpha_i,\pi)$ follows similarly to computing $P(A|G,\pi)$, except that we sum the likelihood of chamber data over all genotypes and configurations in which chamber i contains allele α.

3.5.2 Haplotype assembly

Haplotype assembly requires a large set of high-confidence heterozygous SNVs. For the purpose of haplotype assembly, we used a set of heterozygous SNVs from 60$\times$ Illumina WGS of PGP1f cells (Encode phase 3, ENCSR674PQI) [72]. The original VCF containing SNV calls was lifted over to hg19 using CrossMap and sorted with vcftools [60, 73]. After this, the heterozygous SNV calls were filtered for coverage ≥ 10 and quality score ≥ 30. Reference and variant calls in each SISSOR chamber were grouped into haplotype fragments if they fell inside the boundaries of the same called fragment. Chamber calls that differed from the majority base in the chambers base pileup were filtered out (e.g. in unusual cases where data for individual chambers does not fit cleanly into the diploid base-calling model). Only base calls with coverage ≥ 5 that overlapped the set of heterozygous SNVs were retained, and quality scores of allele calls in haplotype fragments were fixed to 20. Four post-processing steps were applied to increase haplotype accuracy: first, fragments were filtered out if more than 5% of base calls were mixed alleles, which indicate strand overlap from different haplotypes or similar error. Then, fragments were split at spans of multiple mixed-allele base calls (more than 25% of calls having a mixed allele in a span of ≥ 3 heterozygous SNV locations). Then, fragments that were highly discordant to other fragments were filtered out ($\geq 30\%$ rate of switch errors of any length across all overlaps to other fragments). Finally, a haplotype fragment was split if it had a switch error of length 2 SNVs or greater with

respect to multiple overlapping fragments. If it was ambiguous which fragment was the source of the error (for instance, in the case of only two overlapping fragments), multiple fragments were split. Following these fragment processing steps, the processed fragments were assembled into haplotype blocks with HapCUT2 [68]. SNVs were pruned at a HapCUT2 SNV confidence level of 0.95, blocks were split at a switch confidence level of 0.95, and a standard discrete pruning heuristic was applied [20].

3.5.3 Accuracy of haplotypes

To assess haplotype accuracy, it is important to compare against a confident reference haplotype. We compared against haplotypes assembled from BAC clones [44]. To maximize the accuracy of the BAC clone haplotypes, the original BAC fragments were filtered for heterozygous SNVs present in the PGP1f Illumina WGS dataset used to generate SISSOR haplotype fragments [72]. In the same fashion as raw SISSOR fragments were processed, BAC clones were split at positions where 2 or more heterozygous SNVs were switched with respect to other clones. After this, the processed BAC clones were assembled into haplotype blocks using HapCUT2 [68] and pruned at high stringency: SNVs were removed at HapCUT2 SNV confidence level of 0.9999, blocks were split at switch confidence level of 0.9999, and a standard discrete pruning heuristic was applied [20]. Accuracy of SISSOR haplotypes was assessed by allpairs comparison of SISSOR haplotype blocks to these high-stringency BAC haplotype blocks. Accuracy was measured using the concept of switch and mismatch errors (also called long and short switches, respectively) [43]. Looking at positions shared by a single SISSOR haplotype block and a single BAC haplotype block, a switch error is defined as a heterozygous SNV position where the phase in the SISSOR haplotype block is different than the BAC reference with respect to the previous shared position (called in both haplotypes). Two switch errors occurring in a row are instead called a single mismatch error, which results in a difference in phase of only one SNV with respect to the BAC reference. The mismatch discordancy rate is defined as the fraction of compared

positions that had a mismatch error. The switch discordancy rate is similarly defined, but the denominator is slightly smaller as it does not count first and last compared SNVs in a block (these are always called mismatch errors if the phase differs). The term discordancy rate is used instead of error rate because it is assumed that the BAC haplotypes, while accurate, may have non-negligible error.

3.5.4 Same haplotype strand pairing

Fragments were assigned to haplotypes by matching them back to the assembled haplotype blocks. A fragment was required to match the assembled block with 80% accuracy or greater and contain at least 2 haplotype-informative calls at heterozygous SNV positions. After assignment, all base calls (with calls different from the pileup majority base filtered out) inside overlapping fragments were analyzed and a position was classified as a strand-match if both fragments had the same call and as a strand-mismatch if the fragments had different calls. Strand-mismatched positions were quantified for the purpose of estimating the effects of errors from MDA, DNA damage, and other sources. Strand-matched positions in adjacent chambers (chambers 1 and 2, chambers 2 and 3, ... chambers 23 and 24) were discarded, because cases of DNA leakage were observed where DNA from a single strand leaked to physically adjacent chambers and generated a false haplotype-paired strand. The remaining strand-matched calls are of higher confidence than other calls because of their haplotype support, so these calls were tested for concordance against a curated set of SNV and reference calls for PGP1 (described below). Strand-matched calls between strands in different cells that differed from the PGP1 reference were used to estimate the maximum error rate for strand-matched calls in SISSOR technology since these calls are shared by the cell line. Strand-matched calls between strands in the same cell that differed from the PGP1 reference were analyzed as potential de novo variants specific to the cell.

3.5.5 Accuracy of SNV calling

SNV (and reference) calls from SISSOR were compared to a dataset obtained by combining multiple sources for PGP1. First, raw BAMs from a 60× Illumina WGS sequencing of PGP1f cells (Encode phase 3, ENCSR674PQI) were used to generate calls at every genomic position using Freebayes with the standard_filters and report_monomorphic options [65]. These calls were lifted over to hg19 with CrossMap and sorted with vcftools [60, 73]. The single-nucleotide calls in this dataset were filtered for those matching a CGI WGS dataset for PGP1, to filter for only high-quality calls shared by both samples [74]. The resulting intersected dataset had 2.7 billion reference calls and 3.0 million SNV calls. This dataset served as the basis for comparison (SISSOR calls were compared against positions called in the intersected dataset). Variants observed in BAC sequencing libraries [44] served as an extra source for validation; calls that differed from the intersected data but were seen in BAC were considered to be correct.

3.5.6 Workflow management

The complete workflow for variant calling, haplotype assembly, haplotype strand pairing, and accuracy calculations was managed with Snakemake [75].

3.6 Acknowledgements

Chapter 3, in part, is a reprint of the material as it appears in Ultraaccurate genome sequencing and haplotyping of single human cells. Chu, Wai Keung; Edge, Peter; Lee, Ho Suk; Bansal, Vikas; Bafna, Vineet; Huang, Xiaohua; Zhang, Kun. Proceedings of the National Academy of Sciences, 114(47), pp.12512-12517. PNAS, 2017. The dissertation author was the secondary author of this paper.

Permission to reuse the article is granted by PNAS as described at `https://www.pnas.org/page/authors/licenses`.

3.7 Figures and Tables

Table 3.1: Tabulated data in cross chamber base calling algorithm

Phred-scaled Quality	10	30	50	70	90	110	130	150
Calls above cutoff[1]	2.10E+09	2.10E+09	2.09E+09	2.05E+09	1.33E+09	1.33E+09	1.30E+09	6.98E+08
Calls seen in References[2]	2.05E+09	2.05E+09	2.05E+09	2.01E+09	1.31E+09	1.31E+09	1.28E+09	6.89E+08
Mismatch References[3]	122852	45563	26663	12152	4614	2027	1396	510
SNV Matches	1704182	909054	889605	613669	379925	357653	177096	144378
False Positive Rate[4]	5.47E-05	1.35E-05	5.07E-06	1.42E-06	1.03E-06	5.40E-07	1.31E-07	2.13E-07
False Discovery Rate[4]	6.72E-02	4.77E-02	2.91E-02	1.94E-02	1.20E-02	5.64E-03	7.82E-03	0.000992
Error rate[4]	5.98E-05	2.22E-05	1.30E-05	6.05E-06	3.52E-06	1.55E-06	1.18E-06	8.44E-07

[1] Unique base called in SISSOR.
[2] Regions covered by both SISSOR and the combined coverage of CGI, WGS and BAC references (7, 12, 14).
[3] Base call (SNV or reference) disagreed with CGI, WGS and BAC references.
[4] These are upper bounds for each statistic; Calculated against CGI/WGS+BAC data set as ground truth:

- False Positive Rate = FP / (FP + TN)
- False Discovery Rate = FP / (TP + FP)
- Error Rate = (FP + FN) / (FP + TP + FN + TN)

where:

- FP = called SNV allele, CGI+WGS called 0/0
- TP = called SNV allele, CGI+WGS called 0/1 or 1/1 (same SNV)
- FN = called hg19 reference allele, CGI+WGS called 1/1
- TN = called hg19 reference allele, CGI+WGS called 0/0 or 0/1

Table 3.2: Summary of error rate analysis from strand-strand consensus

	Same Cell[1]	All Cell[2]	Cross Cell[3]
Total Unique Positions (DP>=5)[4]	153,115,359	461,395,530	355,226,678
Positions included in CGI/WGS reference[5]	150,976,835	455,719,630	351,156,792
SNP counts	118308	357722	273683
Difference to CGI/WGS reference[5]	98	115	19
Difference to CGI/WGS/BAC reference[6]	94	102	9
Difference to CGI/WGS/BAC/3rd chamber[7] and unconfirmed variants	68	72	4
Error rate (upper bound)[8]	4.50E-07	1.58E-07	1.14E-08
False Discovery rate	5.07E-04	1.79E-04	1.46E-05

[1] Two strands of identical haplotype only in the same cell. (Unique haploid positions)
[2] Two strands of identical haplotype matching in any cell. (Unique haploid positions)
[3] Two strands of identical haplotype only in between two different cells. (Unique haploid positions)
[4] SISSOR coverage
[5] CGI/WGS reference coverage
[6] Combined CGI/WGS reference to BAC reference (12)
[7] Internal reference from SISSOR
[8] Maximum error rate in SISSOR

Table 3.3: Summary of differences in individual cells

	Cell 1	Cell 2	Cell 3
Total Unique Positions (DP>=5) [1]	53,956,666	70,285,423	30,654,766
Positions included in CGI/WGS reference[2]	53,203,331	69,220,980	30,306,948
SNP counts	41400	54832	23477
Difference to CGI/WGS reference[2]	14	75	9
Difference to CGI/WGS/BAC reference[3]	14	71	9
Difference to CGI/WGS/BAC/3rd chamber[4] and unconfirmed variants	14	45	9
Error rate (upper bound) [5]	2.63E-07	6.50E-07	2.97E-07

[1] SISSOR coverage
[2] CGI/WGS reference coverage
[3] Combined CGI/WGS reference to BAC reference (12)
[4] Internal reference from SISSOR
[5] Maximum error rate in SISSOR

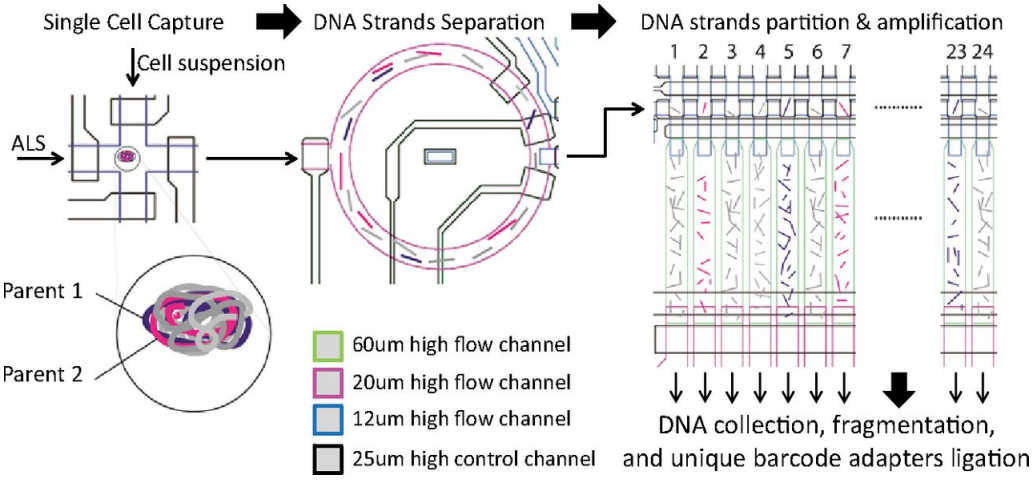

Figure 3.1: [figure and caption from publication [24]] An overview of the experimental process of SISSOR technology. A single cell in suspension was identified by imaging and captured. The cell was lysed, and chromosomal DNA molecules were separated into single-stranded form using ALS. The single-stranded DNA molecules were randomly distributed and partitioned in 24 chambers. Each partition was pushed into an air-filled MDA chamber using a neutralization buffer, followed by an MDA reaction solution. MDA reaction was carried out by heating the entire device at 30 C overnight. The amplified product in each individual chamber was collected out of the device and processed into the barcoded sequencing library.

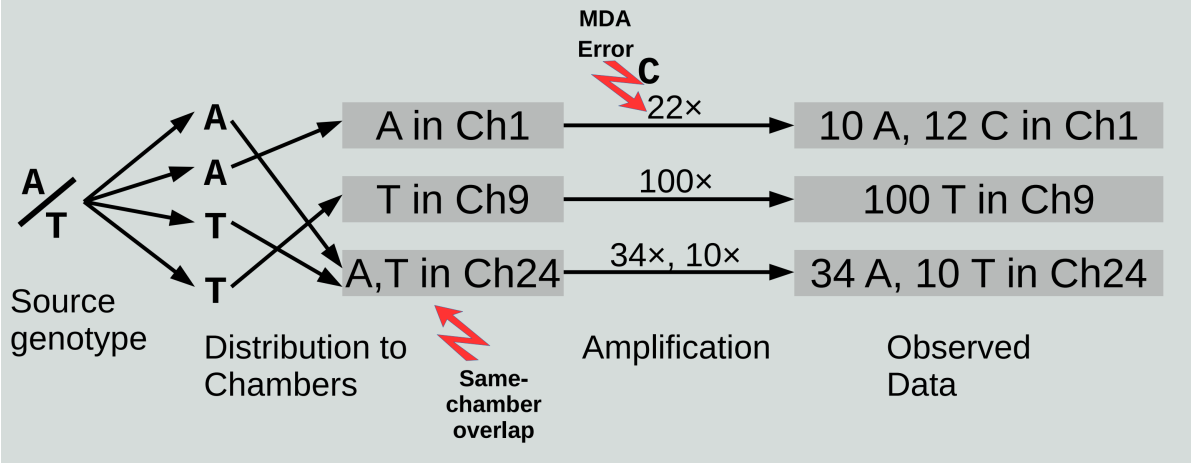

Figure 3.2: The SISSOR variant calling algorithm considers all possible ways that data for multiple reaction chambers could have been generated, accounting for sources of error such as same-chamber allele overlap and amplification (MDA) error. The graphic shows one possible data generation event considered by the SISSOR variant calling algorithm for bases observed at a single variant site. In this example, the observed data includes a mixture of two alleles (A and C) in chamber 1, many observations of a single allele (T) in chamber 9, and a mixture of two different alleles (A and T) in chamber 24. One likely way that this data could have been generated is if the source genotype were A/T and fragments from different haplotypes co-occurred in chamber 24. The observed C alleles in chamber 1 could be the result of amplification error.

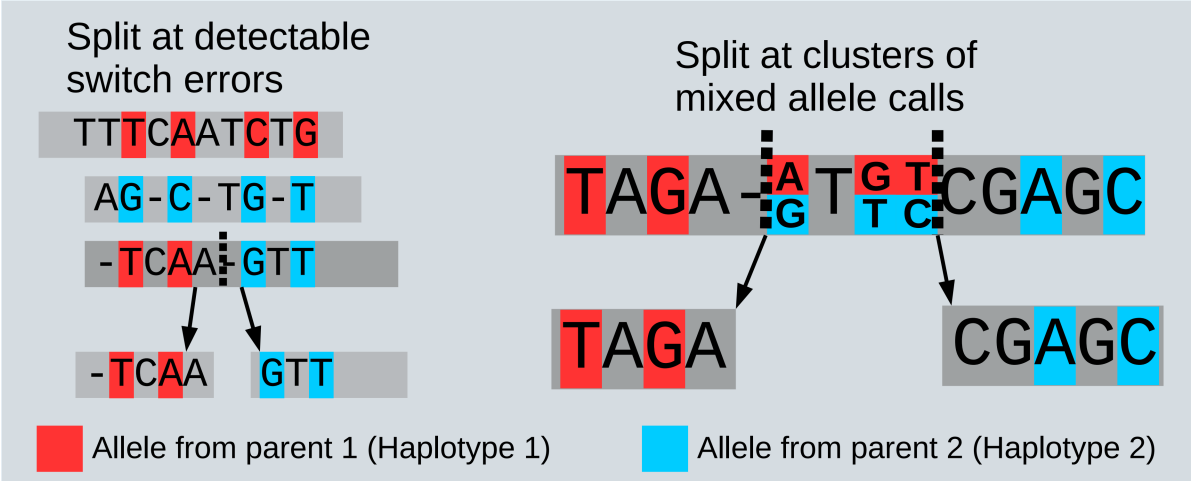

Figure 3.3: After variant calling, SISSOR fragments are processed prior to haplotype assembly to remove errors. If a fragment contains a "switch error" (switching from one haplotype sequence to the other) that is detectable by comparison to other fragments, the fragment is split at that location. Similarly, if a fragment contains a region with multiple sites called as "mixed" (same-chamber allele overlap), the fragment is split and the mixed region is removed. Both switch errors and mixed allele calls are evidence of the two haplotypes overlapping in the same chamber.

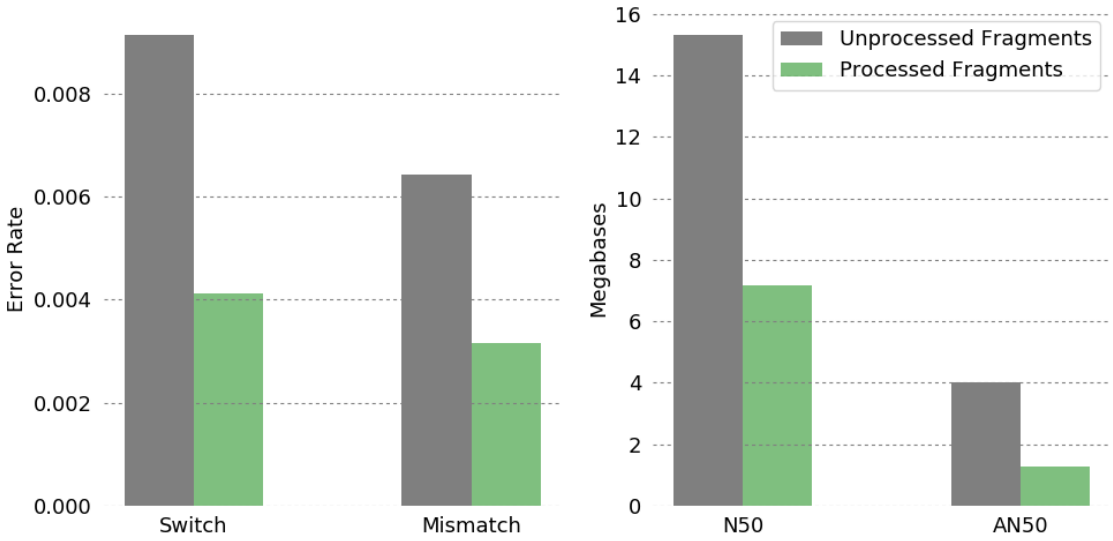

Figure 3.4: SISSOR fragments were processed to remove haplotype errors prior to haplotype assembly, by splitting fragments at detectable switch errors and regions with multiple mixed-allele calls. The error rates (in terms of switch and mismatch errors) and completeness (in terms of N50 and AN50) are shown for the haplotypes assembled using the processed and unprocessed fragments for 3 cells.

Chapter 4

Longshot enables accurate variant calling in diploid genomes from single-molecule long read sequencing

4.1 Abstract

Whole-genome sequencing using sequencing technologies such as Illumina enables the accurate detection of small-scale variants but provides limited information about haplotypes and variants in repetitive regions of the human genome. Single-molecule sequencing (SMS) technologies such as Pacific Biosciences and Oxford Nanopore generate long reads that can potentially address the limitations of short read sequencing. However, the high error rate of SMS reads makes it challenging to detect small-scale variants in diploid genomes. We introduce a variant calling method, Longshot, that leverages the haplotype information present in SMS reads to accurately detect and phase single nucleotide variants (SNV) in diploid genomes. We demonstrate that Longshot achieves very high accuracy for SNV detection using whole-genome Pacific Biosciences data, outperforms existing variant calling methods, and enables variant

detection in duplicated regions of the genome that cannot be mapped using short reads.

4.2 Introduction

The availability of second-generation DNA sequencing technologies such as Illumina short reads has made the resequencing of human genomes routine [76]. Both single nucleotide variants (SNVs), the most abundant form of variation in the human genome, and small indel variants can be reliably detected using whole-genome Illumina sequencing using sequence coverage of 30–40$\times$ [77, 78]. Nevertheless, sequencing human genomes using short-read sequencing technologies has many limitations. First, humans are diploid organisms with two copies (maternal and paternal) of each autosomal chromosome. Haplotypes, or the sequence of alleles that occur on an individual chromosome, can be computationally assembled from whole-genome sequencing using overlaps between reads that span multiple heterozygous variants [39, 32, 20]. However, due to the low rate of heterozygosity of human genomes [79], Illumina reads derived from paired-end sequencing of short fragment libraries (200–500 bp in length) typically cover only a single variant site, and do not provide long-range haplotype information. Second, approximately 3.6% of the genome consists of long and highly similar duplicated sequences where short-reads cannot be uniquely mapped and hence SNVs cannot be detected. These regions overlap hundreds of coding genes, including many disease associated genes such as *PMS2* and *STRC* [80].

Third-generation single molecule sequencing (SMS) technologies such as Pacific Biosciences (PacBio) and Oxford Nanopore (ONT) generate long sequence reads; average read lengths for the PacBio Single Molecule, Real-Time (SMRT) technology are 10–30 kilobases [81]. These long reads have the potential to overcome many of the limitations of short read sequencing technologies including haplotyping and detection of structural variation. Indeed, SMS data has been successfully used for de novo assembly of human genomes [38, 82], identifying complex structural variation [83] and haplotype assembly of human genomes [68, 38]. However, compared

to short read sequencing technologies such as Illumina, the per-base accuracy of SMS reads is low with an error rate exceeding 10% (primarily due to insertion/deletion errors) [81]. This high error rate makes the detection of small sequence variants such as SNVs, particularly heterozygous variants, difficult.

With the decreasing cost of SMS technologies and their increasing use for sequencing human genomes, accurate short variant calling methods for long read SMS data can be valuable in many ways. Current benchmarks for variant calling in human genomes, developed by the the Genome in a Bottle (GIAB) Consortium [84, 85], are based on short read sequence data and cover $\sim$90.8% of the reference human genome sequence. These high-confidence variant calls are immensely valuable for developing new variant calling methods and sequencing technologies. However, these variant call sets are biased towards regions of the genome that are easy-to-call using short reads [86]. Accurate SNV calling using long read SMS data can provide independent validation of short read SNV calls leading to reduction in false positives and increased understanding of systematic errors and artifacts. Furthermore, SNV calling using SMS reads can enable the generation of high-confidence variant calls in repetitive regions of the genome that include segmental duplications. The ability to call variants in repetitive regions that are inaccessible to short read sequencing technologies can also advance the use of SMS technologies for detection of disease causing mutations in duplicated genes via whole-genome or targeted sequencing [87].

Haplotype-resolved SNV detection from SMS reads can also enable the discovery of other types of human genetic variation, such as structural variants (SV) via separation of reads using haplotypes. Huddleston et al. [88] used an assembly-based approach, SMRT-SV, to identify thousands of SVs from whole-genome PacBio data of two haploid genomes, 89% of which were not reported by the 1000 Genomes Project [89]. However, the sensitivity of SV detection using SMRT-SV was only 41% in diploid genomes. Chaisson et al performed dense whole-genome haplotyping of a human genome using multiple sequencing technologies, and were able to call

structural variants successfully on each group of haplotype-separated SMS reads [90].

Variant calling tools such as GATK HaplotyperCaller [64] and FreeBayes [65] developed for short read data analysis are not well-suited for SNV detection using PacBio data for two reasons: (i) short reads have low error rates ($< 0.5\%$) and these methods do not model the high indel error rate of SMS reads which makes it difficult to distinguish true SNVs from errors and (ii) these methods analyze reads in short windows (typically a few hundred bases) and are not designed to leverage the haplotype information present in SMS reads. This haplotype information can be invaluable in distinguishing true variants from errors since observations of a true variant segregate with the reads originating from the haplotype on which it occurs, whereas sequencing errors are unlikely to segregate. Recently, several methods for variant calling from long reads and deep-learning based variant calling methods have been developed [91, 92, 93]. However, the accuracy of these methods for SNV calling on SMS data is currently much lower than that using Illumina whole-genome sequencing (WGS) [93, 92].

We describe a diploid SNV calling method, Longshot, that harnesses long SMS reads to jointly perform SNV detection and haplotyping. For this, it uses our read-based haplotype phasing method HapCUT2 [68]. To overcome the high error rate of SMS reads, it utilizes a pair-Hidden Markov Model to average over the uncertainty in the local alignments and estimate accurate base quality values that can be used for calculating genotype likelihoods. We benchmarked Longshot using simulated data and whole-genome SMS data for multiple human individuals sequenced using the PacBio SMRT and Oxford Nanopore sequencing technologies [84, 85, 26]. LongShot achieves very high accuracy for SNV detection (precision $\geq$0.992 and recall $\geq$0.96) on PacBio SMS datasets and outperforms current variant calling methods in accuracy and run-time. We find that Longshot can also call SNVs with high accuracy using whole-genome Oxford Nanopore data.

4.3 Results

4.3.1 Overview of method

Alignments of SMS reads suffer from reference bias which can cause a SNV allele to be obscured by gaps (insertions and deletions) in the alignments (Supplementary Figure B1). Nevertheless, a true SNV is likely to have at least a few correctly aligned reads with the alternate allele. The first step in the Longshot algorithm identifies potential SNV sites using a standard pileup-based genotyping calculation [63] (Fig. 4.1a). A low variant quality threshold is used to select SNVs in order to minimize false negatives. Next, for each candidate SNV, we determine the most likely allele for each read covering the SNV and the corresponding estimate of the quality of the allele call (Fig. 4.1b). This allelotyping is done by local realignment of a segment of the read to short haplotype sequences (one for each of the two alleles at a biallelic SNV site). In low-complexity regions of the genome (e.g. homopolymers), there is significant ambiguity in the placement of gaps for SMS reads and many alignments are equally likely [94]. Therefore, we use the forward algorithm on a sequence alignment pair-HMM [95] to perform the local realignment by averaging all possible local alignments of a read to a given haplotype.

After estimating the allele call and quality value for each read overlapping a SNV site, we estimate phased genotypes for all SNVs simultaneously using a haplotype-based likelihood model (see Methods). SMS reads typically cover multiple heterozygous sites and this haplotype information is useful since a SNV on a haplotype is expected to segregate with reads from the same haplotype (while random sequencing errors are not). In Longshot, heterozygous SNVs are assembled into haplotypes using HapCUT2 and a local update procedure is used to estimate the most likely phased genotype for each SNV given the current haplotypes for all other SNVs (Fig. 4.1c). This procedure is repeated for a few iterations until the likelihood stops improving. Finally, the variants are filtered for maximum read coverage, excessive variant density and minimum Genotype Quality (GQ) score, where the GQ score is estimated using the phased genotype

likelihoods.

4.3.2 Accurate SNV calling using simulated data

First, we used simulations to assess the accuracy of SNV calling using Longshot and also compared the precision and recall to short-read variant calling. We simulated a diploid genome by adding SNVs to the reference human genome, and simulated paired-end Illumina reads and PacBio SMS reads from this genome (maximum coverage of 60×). Subsequently, we aligned the reads to the reference genome using BWA-MEM [61] (Illumina) and BLASR [96](PacBio) and called SNVs using FreeBayes and Longshot respectively. Across the entire genome, the precision was consistently high ($\geq$0.9999) at all read coverages (20–60×) for both short read and SMS read-based SNV calling (Supplementary Figure B2). Short reads achieved greater recall than SMS reads at lower coverage ($\leq$30×), while SMS reads had marginally greater recall at higher coverage ($\geq$40×). SMS reads are expected to have better mappability in repetitive regions of the genome compared to Illumina reads, particularly in long segmental duplications with high sequence identity. Indeed, the recall for SMS reads in segmental duplications with high sequence similarity ($\geq$95%, 127.5 Mb of DNA sequence) was significantly higher (0.86 at 40× coverage) compared to that using short reads (0.57 at 40× coverage) and increased with increasing coverage (Supplementary Figure B2).

We also compared the precision/recall of SNV calling using BLASR with several long-read mapping tools: NGMLR [97], BWA-MEM [61], and MINIMAP2 [98]. All tools showed high precision and recall when considering SNVs across the whole genome, but BLASR had significantly higher recall (maximum of 0.88) than all other aligners (0.72 using Minimap2) in segmental duplications (Supplementary Figure B2). Therefore, we utilized BLASR for the analysis of real datasets.

We used the simulated datasets to estimate the theoretical fraction of the genome that is callable with SMS long reads compared to short reads at 60× coverage. We found that SMS

reads were able to span 99.4% of the genome (non-N bases on chromosomes 1-22 with at least 30x coverage and at least 90% of reads well-mapped at each position (Supplementary Figure B3)). In comparison, Illumina reads covered 96.3% of the genome under these same criteria, a difference of 3.1%.

4.3.3 Accurate SNV calling using whole-genome PacBio data

We used Longshot to call SNVs using whole-genome human PacBio data for four human genomes from the Genome in a Bottle (GIAB) consortium [84]. Specifically, we used WGS data for the NA12878 individual (45×) and a mother-father-child trio of Ashkenazi ancestry (NA24385 at 64×, NA24149 at 29×, and NA24143 at 27×). For each dataset, a genotype quality threshold that was linearly proportional to the median read depth was used for filtering variants (see Methods). For comparison, we also called SNVs using Illumina short-read WGS data (∼30× coverage) for each individual.

Longshot identified 3.51 to 3.65 million SNVs per genome (on chromosomes 1-22 only) and required 35 hours on average to process ∼28× whole-genome data on a single-core (Supplementary Table B1). To assess the precision and recall of SNV calling, we utilized the GIAB high-confidence variant call set for each individual [84, 85]. The comparison of SNV calls was limited to GIAB high-confidence regions for each genome [85]. The precision and recall for NA12878 were 0.9942 and 0.9592 respectively at 30× coverage and the recall improved to 0.9734 at 45× coverage. The precision and recall and the precision-recall curves (Supplementary Figure B4) were highly consistent across the four genomes at 27-30× coverage (Figure 4.2a and b), demonstrating the robustness of our method. To assess the improvement in precision/recall as a function of sequence coverage, we sub-sampled data for the AJ son individual (NA24385), who was sequenced to 64× coverage. The recall improved steadily from 0.9608 (28×) to 0.9798 (64×) while the precision only changed moderately with increasing coverage (0.9930 to 0.9936). The precision and recall for SNV calling using SMS reads was slightly lower than Illumina based

variant calling (Figure 4.2). Nevertheless, the ability of Longshot to consistently achieve high recall (only 2-3% lower than Illumina WGS for the same depth of coverage) while achieving a low false discovery rate (average = 0.7%) was remarkable given the significantly high error rate of SMS reads ($\sim$10%) compared to Illumina reads.

In contrast with simulated data, the precision of of Longshot on real SMS reads was slightly lower than short read variant calling. To determine the source of false positive calls, we analyzed if such calls were enriched in specific sequence contexts or overlapped with indels. For the NA12878 dataset, (Supplementary Table B2) we observed that the vast majority (71.4%) of false positive SNVs are located within 5 bp of a true indel. These false positive SNVs are called since the current implementation of Longshot does not consider indels as potential variants. Filtering SNV calls located near known indels (using the Mills + 1000 Genomes Gold Standard Indels set from the GATK resource bundle [64]), reduced the number of false positives by $34-45\%$ for the four GIAB genomes (Supplementary Table B3) while only slightly decreasing the recall. Analysis of false negative SNVs showed that 19.5% of false negative SNVs occurred inside homopolymer sequences of length 5 or greater, which is $3.4\times$ the expected value. This follows naturally from the fact that these regions have low information content; insertion and deletion errors could plausibly lie anywhere along the length of a homopolymer. Therefore, allele calls inside homopolymers receive lower quality scores from the pair-HMM realignment, which reduces the power to call SNVs in such regions.

To compare Longshot's accuracy on SMS data with other methods, we considered existing variant calling methods for short read data including GATK and FreeBayes. However, the GATK HC tool did not generate variant calls on the NA12878 PacBio dataset, consistent with previous evaluations of these methods on SMS data [92]. Recently, a deep learning based method for variant calling has been developed that can process both Illumina and SMS long-read data [92]. Although we were unable to perform a direct comparison with DeepVariant due to unavailability of trained models for PacBio continuous long read (CLR) data, comparison of the reported

precision and recall for DeepVariant on the NA12878 dataset (aligned with the same tool) showed that Longshot had better precision than DeepVariant while the recall was similar (Supplementary Table B4). At a genotype quality cutoff of 36, Longshot had the same recall as DeepVariant but higher precision (0.9939 versus 0.9819).

We directly compared the accuracy of Longshot with a deep learning based method Clairvoyante [99] and WhatsHap [93], using whole-genome SMS data for four individual genomes. We used reads aligned with the NGMLR aligner [97] for evaluation since Clairvoyante provides trained models for this aligner (see Supplementary Methods for details). For WhatsHap, we used the potential variants identified in step 1 of Longshot as input since the current version of this tool (version 0.18) does not support potential variant identification. On the NA12878 dataset, although the precision and recall for Longshot were higher than both Clairvoyante and WhatsHap (Table 4.1). In particular, Longshot achieved very high precision or a low false discovery rate (FDR) of 0.5%. In comparison, the FDR for Clairvoyante was 3-fold higher, 1.6%. Comparison of the precision-recall curves for three methods on the NA12878 dataset showed that Longshot outperforms both competing methods for all precision values greater than 0.98 (Supplementary Figure B5). Similarly, analysis of variant calls for two other GIAB genomes (NA24143 and NA24149) showed that Longshot had the best precision and recall among the three methods (4.1). On the high-coverage NA24385, Clairvoyante's recall and precision were marginally better than Longshot (0.3-0.4% higher). Nevertheless, the precision (0.994) and recall (0.980) for Longshot on this dataset using the BLASR alignments were better than Clairvoyante (0.990 and 0.969 respectively). Longshot was also the most computationally efficient of three methods in terms of run-time (4.1). For the NA12878 dataset, the maximum memory usage for Longshot was 5.5 GB compared to 6.2 and 12.7 GB for WhatsHap and Clairvoyante respectively.

The phased genotyping or haplotype assembly step of Longshot distinguishes it from state-of-the-art variant callers for short read data [64, 65] and recent deep learning based methods for variant calling [99, 92]. We investigated the importance of the phased genotyping for

the accuracy of Longshot by running it on the NA12878 PacBio dataset (downsampled to 30× coverage) without phased genotyping (essentially skipping step 3 of the algorithm). We found that skipping the phased genotyping reduced Longshot's recall significantly from 0.959 to 0.905 (genotype quality threshold of 30) while the precision remained virtually unchanged (Supplementary Figure B6).

4.3.4 Accuracy of Longshot haplotypes

Next, we assessed the accuracy and completeness of haplotype assembly using Longshot for two GIAB individuals, NA12878 and NA24385, by comparison to gold-standard haplotypes for these individuals inferred using pedigree data (see Methods). The median read lengths for these two datasets were 3,587 and 7,235 bp, respectively. The Longshot haplotypes for NA12878 had an N50 length of 217.4 kb (with respect to the phased portion of the genome) and were very accurate, with a combined switch error rate of 0.05% (Figure 4.2c and d). Similarly, the haplotypes for NA24385 (30× coverage) had an N50 length of 299.9 kb and a combined switch error rate equal to 0.04%. In comparison, haplotypes assembled using short reads had a N50 length less than 2 kb for both genomes (Figure 4.2d). We also used HapCUT2 and WhatsHap to assemble haplotypes for NA12878 and NA24385 using SMS reads and SNVs identified using ∼30× coverage Illumina sequencing [68]. We found that the haplotype accuracy and completeness were comparable between the three methods while HapCUT2 had the lowest switch and mismatch error rates (Supplementary Figure B7). Separation of SMS reads using SNV haplotypes can enable discovery of non-SNV variants such as indels and structural variants using methods such as SMRT-SV [82] that work well on haploid genomes. For the NA12878 dataset (chromosome 1 only), 51.1% of reads (weighted by length) could be assigned to a haplotype with high confidence. The ability to assign reads to haplotypes was dependent on read length: the haplotype-assigned reads had a median length of 4.3 kb while the unassigned reads had a median length of 2.6 kb only.

4.3.5 SNV calling using Oxford Nanopore reads

Recently, reads from Oxford Nanopore Technologies' (ONT) MinION sequencer were used to assemble a human genome [26]. Nanopore reads have a similar error profile to PacBio SMRT reads, however, the total per-base error rate of ONT reads is reported to be higher than for PacBio SMRT [100] and the errors are dependent on sequence context [101]. We applied the Longshot algorithm to call SNVs using a whole-genome Oxford Nanopore dataset for a human individual (NA12878, 37$\times$ coverage). We observed that the candidate set of SNVs considered by Longshot contained a significant fraction of false positives due to the context-specific errors in Nanopore reads. To amelioriate this, we implemented a simple filter to remove potential SNVs for which the allele observations show a significant strand bias (Fisher's exact test p-value < 0.01), prior to haplotype assembly. On the latest version of this ONT dataset, LongShot achieved a precision equal to 0.991 and recall value equal to 0.933 at a GQ threshold of 65 for SNV calling (see Supplementary Figure B8 for a precision-recall curve). For comparison, we called variants using Nanopolish, a software tool for signal-level analysis of Oxford Nanopore data [101]. Nanopolish required more than 43 hours to call variants on chromosome 20 using 4 cores and achieved a best F1 score of 0.93 (Supplementary Figure B8). In contrast, Longshot had a best F1 score of 0.967 and took only 5 hours and 13 minutes for variant calling (using a single core). In addition, the accuracy of Longshot on Oxford Nanopore data was better than the reported accuracy of other methods (Supplementary Table B4).

4.3.6 Analysis of SNV calls in repetitive regions

As demonstrated with simulations, the recall of variant calling using SMS reads in segmental duplications with high sequence similarity ($\geq 95\%$, Figure 2) is significantly higher (0.86) compared to short reads (0.57). These regions correspond to 102.8 Mb of the genome (excluding the sex chromosomes). However, 97.7% of these regions are excluded from the GIAB

high-confidence variants, making it challenging to assess the accuracy of SNV calling using real SMS data. We compared SNV calls in segmental duplications for the NA12878 genome made using short read Illumina data (33× coverage) and SMS reads (30× coverage). In segmental duplications with $\geq 95\%$ similarity, 180,889 SNVs were called using SMS reads, 55.0% more than those using Illumina reads (Table 4.2). The fewer calls using Illumina reads likely reflect the inability to map in segmental duplications. For example, Illumina reads cannot be mapped uniquely in a significant portion of the *STRC* gene, resulting in 52.3% fewer variants called compared to SMS reads (Fig. 4.3). We found that in total, 1.66 Mb of the bases in segmental duplications with $\geq 95\%$ similarity overlap with coding exons and 90.3% of these bases were well-mapped in the 45× PacBio dataset (each position having at least 20× coverage and $\geq 90\%$ of reads aligned to the position having MAPQ ≥ 30). The difference was more stark in segmental duplications with $\geq 99\%$ similarity: 78,851 SNVs were called with SMS reads compared to only 18,684 with Illumina reads (4.2 fold difference). The Transition/Transversion (Ts/Tv) ratio for the SNVs called only using SMS reads in these regions was 1.99, slightly lower than the ratio for the SNV calls in GIAB confident regions ($\sim$2.1). This is consistent with the expectation that the Ts/Tv ratio is usually $\sim 2.0-2.1$ for SNVs across the whole genome [102]. In contrast, the Ts/Tv ratio for Illumina-only calls in segmental duplications with $\geq$99% similarity was 1.55, much lower than the expected value (Table 4.2).

Next, we assessed the Mendelian consistency of SNV calls for the mother-father-child trio of Ashkenazi ancestry from the GIAB project. To minimize discordance due to false negative calls, only sites with at least 20× read coverage in every individual were considered. SMS calls in the high confidence GIAB regions had higher concordancy (98.88%) compared to calls outside GIAB confident regions (96.17%). Within segmental duplications ($\geq 95\%$ similarity), 4.99% of the SNVs in the child were discordant with Mendelian inheritance. Many of the discordant SNVs were clustered in contiguous blocks, indicating that they are the result of mismatched reads or structural variation in one or more individuals.

Finally, we compared Longshot SNV calls for NA12878 to the Platinum Genomes small variant call set for this genome that have been generated using Illumina WGS and validated using haplotype inheritance on a 17-member pedigree [103]. In GIAB high confidence regions, 95.2% of the PG SNVs were also called by Longshot. The PG calls cover a significant fraction of the genome (330.7 Mb) that is excluded from the GIAB high-confidence calls. In these regions, only 79.6% of the PG SNVs were shared with Longshot and 74,641 SNVs were unique to the PG calls (Supplementary Figure B10). The low concordance in regions outside the GIAB high-confidence regions highlights the challenge of accurate variant calling in these regions. Longshot's ability to call SNVs accurately using SMS reads provides an orthogonal validation for SNVs called using short reads. In-depth analysis of variant calls made using short-read and SMS data in these regions can enable the expansion of confidently called regions for reference human genomes.

4.4 Discussion

Our results demonstrate that highly accurate detection of SNVs is feasible even from long-read sequence data with high error rates. Combined with recent work demonstrating the ability to detect and genotype structural variants from SMRT-seq data, our results indicate that long-read sequencing can be used to accurately detect all forms of genetic variation in human genomes. Recently, Li et al wrote that "although PacBio assembly is accurate at the base-pair level for haploid genomes, it is currently not accurate enough to confidently call heterozygotes in diploid mammalian genomes" [86]. We have demonstrated that heterozygous SNVs can be called accurately in diploid genomes, by combining sensitive allelotyping of reads at SNV sites with haplotype-informed genotyping. Our method has a very low false discovery rate (0.5-0.8%) across multiple whole-genome PacBio datasets that is 2-4 fold lower than other variant calling methods. Furthermore, we find that the FDR can be reduced further to 0.3% by filtering out known common indels.

We have also demonstrated that SMS reads can be used to call SNVs in segmental duplications and other regions of the genome with low short-read mappability. However, correctly mapping PacBio reads in highly similar segmental duplications remains a challenge. As Supplementary Figure B2 shows, there is wide variance in the ability of SMS read mappers to map reads in segmental duplications. This is likely due to the mappers having different strategies for dealing with highly similar mappings that are differentiated by a small number of paralog-specific-variants (PSVs). Despite BLASR performing relatively well using simulated reads, many of the discordant SNVs observed between the AJ trio in segmental duplications appeared to be caused by the presence of multiple mismapped reads. SMS read mapping methods with specific optimizations for segmental duplications could improve the ability to call variants in segmental duplications [98].

The GIAB and Platinum Genomes variant sets used to assess variant calling accuracy in this paper were generated using short read datasets, are therefore biased in favor of short-read technologies [86] and exclude regions where long reads are likely to have better precision and recall. Therefore, in an unbiased genome-wide comparison, Longshot may achieve even better accuracy than short read variant calling methods. Furthermore, some of the false positives calls by Longshot may actually correspond to false negatives in the GIAB high-confidence call sets. A recent graph-based read alignment approach identified thousands of variants that were absent in the GIAB call-sets [104]. In the NA12878 genome, Longshot identified 5900 SNVs that are located in GIAB high-confidence regions and do not overlap indels present in the GIAB variant calls. Many of these variants are located in variant-dense genomic regions that are problematic for mapping using short reads but should be callable using long single-molecule reads. Further analysis of these variants will be helpful in improving the recall of gold-standard variant call-sets for human genomes.

Longshot offers the ability to assemble haplotypes without prior knowledge of SNVs and leverage the haplotypes to separate SMS reads by haplotype. This opens up a wide range

of possibilities for SMS read analysis, given that many SMS analysis tools work much better on haploid samples. For example, the haplotype-separated reads could be used to call structural variants with greater sensitivity using a tool such as SMRT-SV [82]. A similar approach was recently used to profile structural variation genome-wide after extensive haplotype assembly with multiple sequencing technologies and computational separation of the SMS reads by haplotype [90]. Currently, Longshot uses the read pileups to identify candidate SNVs and the vast majority ($\sim$72%) of false positive SNVs identified with Longshot correspond to misclassified indel variants. Using a genomic consensus of haplotype separated reads should improve the accuracy of variant calling using Longshot.

LongShot was also able to call SNVs with high accuracy from Oxford Nanopore long read data without any modification to the likelihood model. Although the precision and recall was lower than PacBio reads at similar coverage, this is expected due to the higher error rate of Nanopore reads. Continued improvements in the sequencing technology and the raw basecalling, and the use of context-specific error models for local realignment are expected to further improve the accuracy of variant calling using Nanopore reads.

In this paper, we focused on the detection and phasing of single nucleotide variants alone since accurate calling of short indels using SMS reads is challenging due to the high insertion/deletion error rate. A recently developed deep-learning based variant caller [92] had low precision (0.589) and recall (0.12) for short indel calling on PacBio WGS data. In comparison to CLR reads, PacBio circular consensus sequencing (CCS) produces reads with greater accuracy by sequencing multiple times around the same DNA template. Recent improvements have enabled the generation of highly accurate long reads (10-15 kilobases read lengths and error rates $< 1\%$) using CCS [105]. We expect that using Longshot with these low-error reads will improve the accuracy of SNV calling and also enable accurate short indel calling. As the cost of SMS technologies continues to decrease, these technologies are likely to see widespread use in human disease studies in the near future. In particular, whole-genome SMS can enable the detection of

disease-associated structural variants and variants in repetitive regions of the genome that cannot be identified using standard Illumina WGS [106, 107, 108]. Tools such as LongShot will be valuable for realizing the potential of SMS technologies for the comprehensive detection of all forms of genetic variation in such studies.

4.5 Methods

4.5.1 Identification of candidate SNVs

The first step in the Longshot algorithm is to identify positions in the genome that may contain a SNV. Potential SNVs are identified from the pileup of aligned reads by performing a genotype likelihood calculation similar to Samtools or other NGS variant calling methods [78] (see Supplementary Methods). The prior probabilities for genotypes are defined using a slight modification of the approach of Li et al [8] (see Supplementary Methods). SNV sites for which the posterior probability of a non-reference genotype is greater than 0.01 are considered as candidate SNVs for the next step of the algorithm. The sites are also filtered for mininum read coverage (6 by default), minimum alternate allele count and fraction (3 and 0.125 by default).

4.5.2 Local realignment using pair-HMMs

For an SMS read that overlaps a candidate bi-allelic SNV site with two alleles 'ref' and 'alt', we want to determine which allele is the most likely observation (allele call) and also assign a probability of error to this observation (quality value). To accomplish this, we perform local realignment of a short sequence from the read to the reference and to the alternate sequence (with the SNV allele added, see Fig 4.1B). This local realignment is performed using a pair-Hidden Markov Model (pair-HMM) [95]. The parameters for the HMM are estimated directly from the aligned reads prior to realignment (see Supplementary Methods).

It is sufficient to perform the local realignment within a short window covering the SNV site. This window is defined using the nearest non-repetitive anchor sequences of length k (default $k = 6$), to the left and right of the SNV where the read sequence matches the reference sequence perfectly (see Supplementary Methods). Once the window W is identified, we use the forward algorithm to calculate $p_{\text{ref}} = P(\text{read(W)} \mid \text{ref(W)})$ and $p_{\text{alt}} = P(\text{read(W)} \mid \text{alt(W)})$ where $\text{read}(W)$ is the sequence of the read in the window W defined by the two anchors. We select the allele $a_{max} \in \{\text{ref}, \text{alt}\}$ for which the probability is higher, as the observed allele and use phred $\left(1 - \frac{p_{a_{max}}}{p_{\text{ref}} + p_{\text{alt}}}\right)$ as the allele quality score.

When multiple candidate SNVs are located in close proximity, we define the window to include all such SNVs and use a generalization of the calculation described above to determine alleles and estimate base quality values (see Supplementary Methods). For computational efficiency, a banded version of the forward algorithm is used. This reduces the complexity to $O(mb)$ where b is the width of the band and m is the length of the window (50-200 bp). Allele observations with phred-scaled quality score below a threshold (7.0 by default) are discarded. This reduces the effective read depth for SMS reads (Supplementary Figure B9).

In order to remove false variants resulting from strand-specific sequencing errors, we filter potential SNVs whose allele observations are over-represented in reads from one strand. For each potential SNV, we build a contingency table of the counts of the reference and alternate allele on reads from the forward strand and reverse strand respectively. Variants for which the Fisher's exact test p-value (2-tailed) is less than 0.01 are not considered for haplotype-informed genotyping.

4.5.3 Haplotype-informed genotyping

Longshot achieves accurate variant calling using SMS reads by performing phased genotyping for all candidate SNVs jointly. Given a set of candidate SNVs V and the allele calls (and quality values) for each read $r \in R$, we aim to maximize the likelihood $p(R|H)$ where H is a pair

of haplotypes (H_1, H_2) over the variant set V. Longshot optimizes the likelihood function using an iterative approach that uses (i) the HapCUT2 algorithm [68] to estimate the most likely pair of haplotypes for variants with heterozygous genotypes and (ii) local updates to estimate the most likely phased genotype for each variant given the current haplotype pair (Fig. 1C).

Assuming independence between reads, the likelihood function $p(R|H)$ can we written as [68]:

$$p(R|H) = \prod_r p(r|H) = \prod_r \frac{p(r|H_1) + p(r|H_2)}{2}$$

$p(r|H_1)$ for any read r can be calculated using the pair-HMM probabilities for each (read,variant) pair. Let G be the set of possible phased genotypes for a biallelic variant: $\{0|0, 0|1, 1|0, 1|1\}$ (homozygous reference, the two heterozygous states, and homozygous alternate). Let H refer to the current estimate of the most likely haplotype pair, and $H^{i,g}$ refer to the haplotype pair H with the ith SNV altered to have the phased genotype g. Given H, we can calculate the posterior probability for the phased genotype g as follows:

$$p(H[i] = g|R, H) = \frac{p(g)p(R|H^{i,g})}{\sum_{g' \in G} p(g')p(R|H^{i,g'})} \tag{4.1}$$

The optimization starts with the initial set of variants identified from the pileup-based likelihood calculation and the unphased genotypes for each variant estimated using the local realignment. The iterative phase of the Longshot algorithm consists of the following steps:

For $i = 1 \ldots k$

1. $L = p(R|H)$
2. Let V' be the set of heterozygous SNVs in V
3. $H(V') = \text{HapCUT2}(R, V')$

4. Repeat:

   (a) For each variant $v \in V$: update $H[v]$ using equation 4.1

   (b) If no genotype was updated in (a), BREAK

5. $L' = p(R|H)$

6. If $\frac{\log(L')-\log(L)}{\log(L)} < \Delta$: BREAK

In Step 3, HapCUT2 is used to phase the current set of heterozygous variants. Then, the haplotype scaffold is used to refine the genotypes of each variant in step 4. This serves to remove false heterozygous variants and identify new heterozygous variants that can be phased by HapCUT2 in the next iteration. Steps 1-5 are repeated until the relative log-likelihood of the data between consecutive iterations is smaller than Δ (default $= 1 \times 10^{-5}$).

4.5.4 Variant filtering

The raw variant calls were subjected to three types of filters to reduce false positives. SNVs were first filtered according by genotype quality (GQ) estimated by the variant caller. The GQ cutoff was fixed at 50 for short reads. For Longshot, we used a variable GQ cutoff (matched to the median read coverage) for filtering variants. This was done to reduce the number of false SNVs due to true indel variants that have high GQ. For simulations, which do not have indel variants, we used a fixed genotype quality cutoff of 50.

To filter false SNVs due to copy number amplifications, a maximum read depth filter similar to what has been used previously for short-read based variant calling was used [109]. Variants with read depth greater than $d + 5\sqrt{d}$, where d is the median read depth across the entire dataset were filtered out. We also observed that for SMS reads, many false positive SNVs occur nearby each other in dense clusters. These dense clusters may result from systematically mismapped reads due to missing sequence in the reference genome, or are indicative of structural variations such as

CNVs. We used a simple density filter ($>$ 10 SNVs in a window of 500 base pairs) to filter out such false variants for variants called with Longshot. For the AJ trio, variants in the Delly exclusion regions (available from https://github.com/tobiasrausch/delly/blob/master/excludeTemplates/human.hg38.excl.tsv) were also filtered out for the analysis of Mendelian consistency.

4.5.5 Simulations

We simulated a diploid genome using the reference human genome sequence with heterozygous SNVs (rate = 0.001) and homozygous SNVs (rate = 0.0005) (see Supplementary Methods for details). Paired-end 100bp reads were generated from the simulated genome with a substitution error rate of 0.001 [110]. The short reads were aligned to the human reference (hs37d5) using BWA-MEM, and variants were called using FreeBayes [65]. Similarly, we used SimLoRD [111] to generate PacBio SMS reads (median length = 7.5 kb) from the simulated genome using the default error rates of 0.11 for insertion, 0.04 for deletion, and 0.01 for substitution [111]. The `-mp 1` option was used to force each read to only have a single sequencing pass, so that the error profile of the reads resembles PacBio continuous long reads (lower accuracy) as opposed to circular consensus reads (greater accuracy). We aligned the SMS reads to the human reference (hs37d5) using the long-read alignment tools BLASR (`v5.3.2`, options `--nproc 16 --bestn 1 --bam`), MiniMap2 (`v2.11-r797`, options `-t 16 -ax map-pb`), BWA-MEM (`v0.7.17`, options `-x pacbio -t 16 -T 0`) and NGMLR (`v0.2.7`, options `-t 16 -x pacbio`).

4.5.6 Whole-Genome Sequencing data

45$\times$ coverage Pacific Biosciences Single Molecule Real Time (SMRT) reads for NA12878, aligned to the hs37d5 reference genome using BLASR, were obtained from the Genome in a Bottle consortium [48]. PacBio read data for the AJ trio was also obtained from the GIAB ftp

site and aligned to the hg38 reference genome using BLASR [96], using the same parameters used for aligning the simulated reads. Oxford Nanopore reads for NA12878 were obtained from the Nanopore WGS Consortium [26] and aligned to hg38 using minimap2. Illumina WGS data for NA12878 and the AJ Trio (NA24385, NA24143, NA24149), sequenced on the HiSeq 2500 (30× and 60× coverage respectively, 148 bp paired-end reads), was obtained from the GIAB. The 60× coverage datasets were downsampled to half coverage. The reads were downloaded in BAM format aligned to as hs37d5 using bwa-mem (NA12878) and hg38 using NovoAlign (AJ trio). Variant calling on Illumina WGS data was performed using FreeBayes [65] (`v1.0.2-33-gdbb6160`) with `--standard-filters` and `--genotype-qualities` turned on). BED files for segmental duplications and repeat elements in the human genome were obtained from the UCSC table browser [112].

4.5.7 Assessment of variant calling and phasing accuracy

High confidence variant call sets generated by the GIAB project were used for assessing accuracy of SNV calling [48, 85]. For NA12878, SNVs were compared against the GrCh37 (for Illumina and PacBio) or GrCh38 (for Oxford Nanopore) version of the GIAB high-confidence call set (release v3.3.2). For the AJ trio, SNVs were compared against the GrCh38 version of the GIAB high-confidence call set (release v3.3.2). For comparing the accuracy of Longshot with Clairvoyante and WhatsHap, the GrCh37 version of the calls were used (release v3.3.2). For each individual, the comparison of SNV calls was limited to high-confidence regions (provided in a bed file). Precision and Recall were calculated using RTGtools (`v3.9.1`) vcfeval.

For NA12878, we compared the accuracy of the Longshot haplotypes using the Platinum Genomes haplotypes for the same individual as ground truth. For NA24385, we generated high-quality haplotypes from a consensus of the GIAB trio-based phased genotypes and 10X Genomics phased variant calls and used the resulting haplotypes for assessment of haplotyping accuracy. The haplotypes were compared at all unfiltered SNVs that were called heterozygous in

both the assembled haplotypes and the ground truth. The errors were tabulated in terms of the total combined rate of switch and mismatch errors, also known as long-switch and short-switch errors respectively [68, 43]. The N50 metric - defined as the length N in base pairs such that half of the phased portion of the genome is in haplotype blocks of length N or greater - was used to measure the completeness of haplotype blocks.

For the AJ trio, Mendelian consistency of the SNV calls was assessed using RTGtools [113]. For this, SAMtools [63] and BEDtools [114] were used to obtain a set of regions that have high coverage ($> 20\times$) of well-mapped SMS reads ($MAPQ > 30$ and filter `-F 3844` applied) in all three individuals. These regions were further intersected with a bed file for the region being investigated (either GIAB confident regions, outside GIAB confident regions, or 95% similar segmental duplications). The individual VCFs for the trio were merged into a single VCF and filtered so that all records have a genotype quality greater than 50.

4.5.8 Server configuration

All experiments in this study were performed on CentOS 6.6 with Intel Xeon CPU E5-2670 0 @ 2.60GHz, with jobs managed by a Torque/PBS system.

4.6 Data Availability

The PacBio and Illumina sequence datasets and variant calls used in this paper are publicly available from the GIAB ftp site: ftp://ftp-trace.ncbi.nlm.nih.gov/giab/ftp/. The sub-folders for each individual are as follows:

NA12878:

data/NA12878/NA12878_PacBio_MtSinai,

release/NA12878_HG001/NISTv3.3.2/GRCh37/,

data/NA12878/NIST_NA12878_HG001_HiSeq_300x

NA24385:

data/AshkenazimTrio/HG002_NA24385_son/PacBio_MtSinai_NIST,
release/AshkenazimTrio/HG002_NA24385_son/NISTv3.3.2/GRCh38,
data/AshkenazimTrio/HG002_NA24385_son/NIST_HiSeq_HG002_Homogeneity-10953946

NA24149:

data/AshkenazimTrio/HG003_NA24149_father/PacBio_MtSinai_NIST/,
release/AshkenazimTrio/HG003_NA24149_father/NISTv3.3.2/GRCh38,
data/AshkenazimTrio/HG003_NA24149_father/NIST_HiSeq_HG003_Homogeneity-12389378

NA24143:

/data/AshkenazimTrio/HG003_NA24143_mother/PacBio_MtSinai_NIST,
release/AshkenazimTrio/HG004_NA24143_mother/NISTv3.3.2/GRCh38,
data/AshkenazimTrio/HG004_NA24143_mother/NIST_HiSeq_HG004_Homogeneity-14572558

For the direct comparison between methods, BAMs aligned using NGMLR from the Clairvoyante study were used[99]. The BAMs were obtained from http://www.bio8.cs.hku.hk/clairvoyante/bamUsed/.

The Oxford Nanopore sequence dataset is publicly available from the Nanopore WGS Consortium: https://github.com/nanopore-wgs-consortium/NA12878/blob/master/Genome.md. The NA12878 genome was sequenced using the Oxford Nanopore MinION with version R9.4 version of the chemistry on 39 flowcells. We used the rel6 version of the base calls (called using ONT Guppy basecalling software version 2.3.8+498297c). All other relevant data are available upon request. The source data underlying Supplementary Figure 9 is provided as a Source Data file.

4.7 Code availability

Longshot is implemented in the Rust programming language, uses the rust-bio and rust-htslib libraries [115] and the HapCUT2 C code [68]. It is freely available for download at `https://github.com/pjedge/longshot`. It is also available on Bioconda [116]. A Snakemake workflow [75] for automatically generating all of the results of the paper is available at `https://github.com/pjedge/longshot_study`.

4.8 Author Contributions

P.E. designed and implemented the algorithm, performed the analyses and wrote the manuscript. V.B. conceived the project, designed the algorithm, performed analyses and wrote the manuscript.

4.9 Competing Interests

The authors declare no competing financial interests.

4.10 Acknowledgments

The research was supported by the National Human Genome Research Institute of the National Institute of Health (award number R01HG010149).

Chapter 4, in full, is a reprint of the material as it appears in Longshot enables accurate variant calling in diploid genomes from single-molecule long read sequencing, 2019. Edge, Peter; Bansal, Vikas. Nature Communications, 10(1), pp.1-10. Nature Publishing Group, 2019. The dissertation author was the primary author of this paper.

the material for this dissertation. The license is available at http://creativecommons.org/licenses/by/4.0/. No significant changes to the material were made, but the material was reformatted where necessary to be a chapter of the dissertation instead of a standalone article.

4.11 Figures and Tables

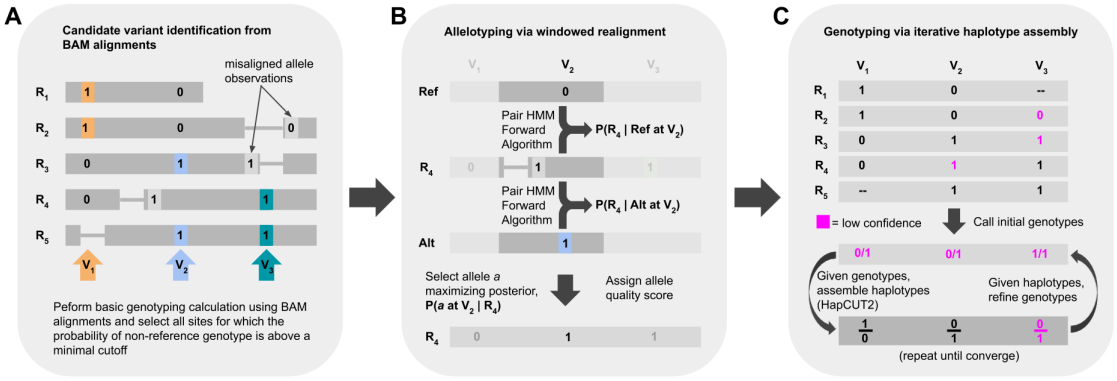

Figure 4.1: Overview of the Longshot algorithm. (a) Candidate variants are identified using the pileup of the original alignments and a standard genotype likelihood calculation is used to determine if the site is a potential variant. (b) To determine the allele for each read at each potential SNV site it overlaps, a window is formed around the variant and the probability of the observed read sequence given each allele is calculated using the forward algorithm on a Pair Hidden Markov Model. The most likely allele and quality score is chosen based on the relative likelihoods of the two alleles. (c) Using the alleles and quality values for each read at variant sites, phased genotypes for all variants are determined jointly by performing haplotype assembly using HapCUT2 (on heterozygous variants) and local updates of the phased genotypes in an iterative manner.

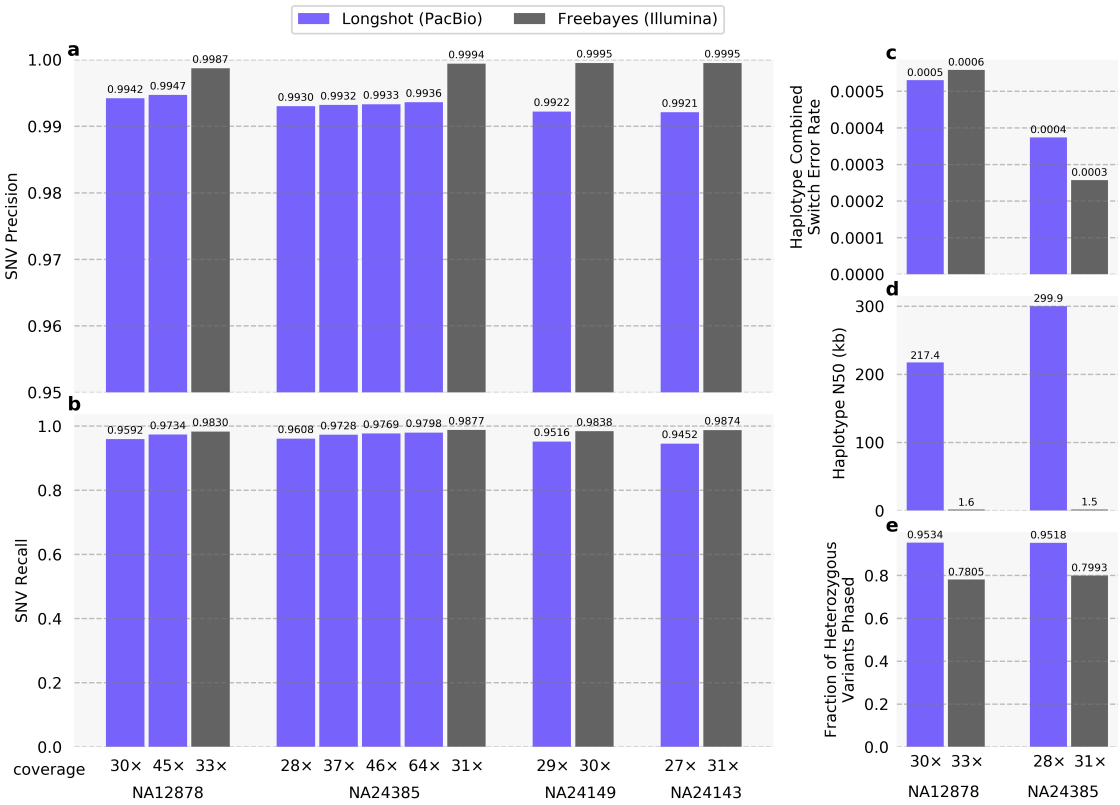

Figure 4.2: Accuracy and completeness of Longshot SNV calls on whole-genome SMS data. Longshot was used to call single nucleotide variants (SNVs) using SMS data from the GIAB project for four human genomes: NA12878 (30× and 45× coverage), NA24385 (28×,37×,46×, and 64× coverage), NA24149 (29× coverage), and NA24143 (27× coverage). For each individual, variants were also called using FreeBayes applied to ∼30× coverage Illumina short reads. (a) Precision of the SNV calls calculated using the GIAB high-confidence variant call set, (b) Recall of the SNV calls, (c) The combined switch error rate (total rate of switch errors and mismatch errors) of the Longshot and Illumina short-read based haplotypes, (d) N50 length of the haplotypes, and (e) The fraction of heterozygous variants phased in each dataset.

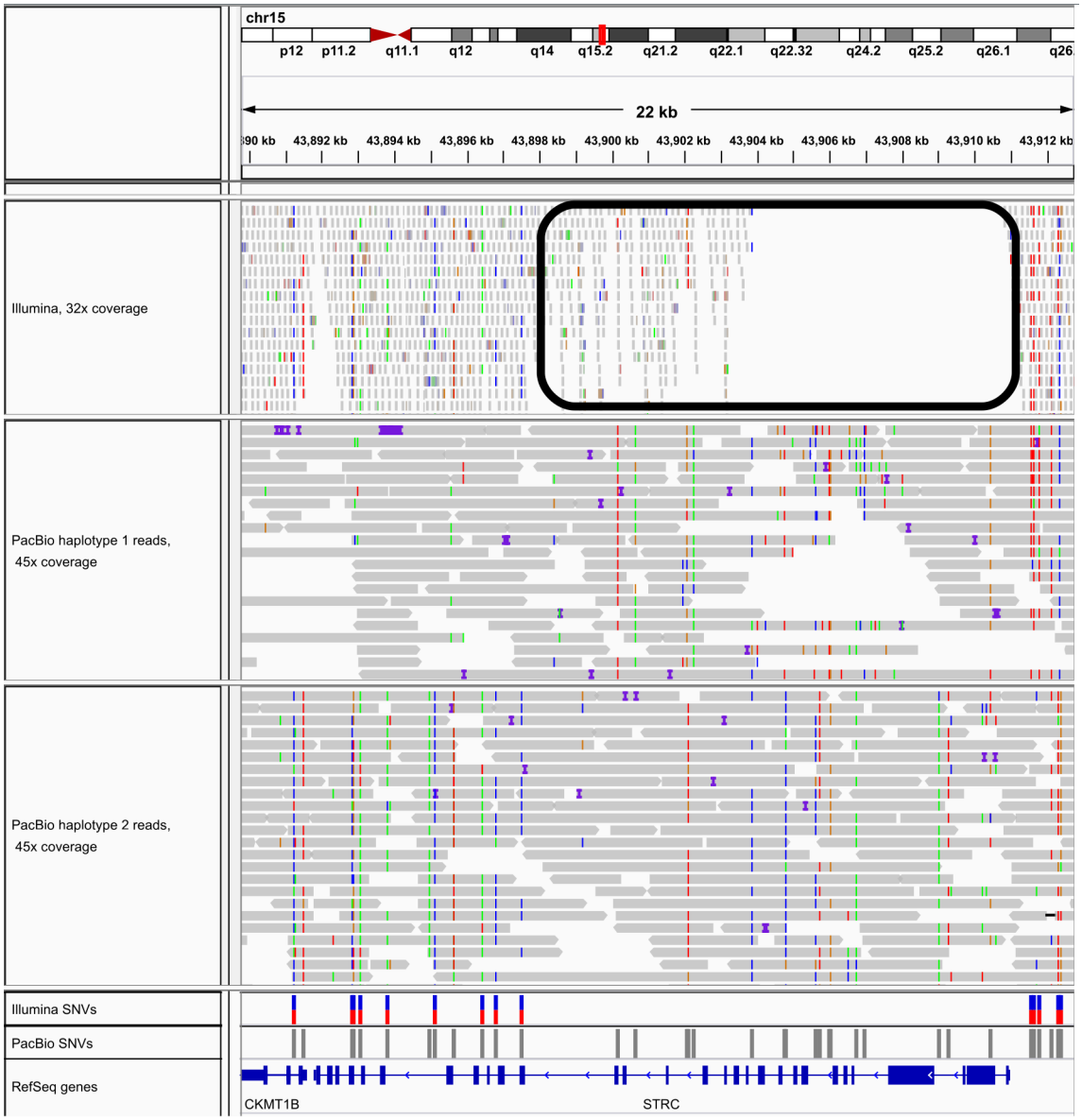

Figure 4.3: Accurate variant calling using SMS reads and Longshot in the duplicated gene *STRC*. An Integrated Genomics Viewer (IGV) view of mapped reads shows that a long segment of the gene (circled in black) has low coverage using uniquely mapped Illumina reads due to the presence of a long segmental duplication with high sequence similarity ($> 99.8\%$) that spans the entire gene. PacBio reads (separated by haplotype using Longshot phased SNVs) have consistent coverage of mapped reads across the entire gene, allowing Longshot to call 42 SNVs of which 20 are shared with short reads, and 22 are unique to Longshot.

Table 4.1: Comparison of accuracy for variant calling methods on whole-genome SMS data. All methods were run on BAM files generated using the NGMLR aligner and precision and recall values were calculated using the GIAB high-confidence variant calls. The runtime listed is the total walltime to process all chromosomes individually. Clairvoyante supports multi-threading and was run using four threads per chromosome.

Genome	Read Coverage	Method	Precision	Recall	Run time (hours)
NA12878	44	Longshot	0.995	0.968	23:31
		WhatsHap	0.972	0.975	27:47
		Clairvoyante	0.984	0.957	21:44 (x4)
NA24385	62	Longshot	0.987	0.965	41:55
		WhatsHap	0.976	0.974	32:09
		Clairvoyante	0.990	0.969	22:25 (x4)
NA24385	27	Longshot	0.981	0.927	20:03
		WhatsHap	0.959	0.941	22:54
		Clairvoyante	0.960	0.927	21:09 (x4)
NA24143	27	Longshot	0.993	0.941	18:51
		WhatsHap	0.962	0.949	22:06
		Clairvoyante	0.960	0.920	21:42 (x4)
NA24149	23	Longshot	0.993	0.924	16:59
		WhatsHap	0.959	0.934	20:30
		Clairvoyante	0.938	0.904	23:59 (x4)

Table 4.2: Comparison of Pacbio and Illumina SNV calls for NA12878. Variants were called using short reads (33× coverage) with FreeBayes, and using SMS long reads (30× coverage) with Longshot. The number of variants called by each technology, the number of variants shared between the two technologies, and the corresponding transition/transversion (Ts/Tv) ratios are shown for the whole-genome and various subsets of the genome including GIAB high-confidence regions and segmental duplications with high sequence identity.

		Genome (1-22)	Inside GIAB Confident	Outside GIAB Confident	Segmental Dup. ($\geq$ 95% similar)	Segmental Dup. ($\geq$ 99% similar)
Region size		2.8 Gb	2.4 Gb	330.7 Mb	102.8 Mb	47.5 Mb
PacBio	# SNVs	3,518,530	3,002,660	515,870	180,889	78,851
	Ts/Tv	2.08	2.14	1.75	1.95	1.99
Illumina	# SNVs	3,563,787	3,065,573	498,214	116,649	18,684
	Ts/Tv	2.03	2.1	1.66	1.84	1.79
Unique to PacBio	# SNVs	254,428	63,848	190,580	103,621	69,705
	Ts/Tv	1.63	1.83	1.57	1.9	1.99
Unique to Illumina	# SNVs	299,733	126,763	172,970	39,409	9,538
	Ts/Tv	1.3	1.26	1.33	1.53	1.55
Shared Illumina & PacBio	# SNVs	3,264,078	2,938,812	325,266	77,241	9,146
	Ts/Tv	2.12	2.15	1.85	2.01	2.04

Appendix A

Supplemental Material for Chapter 2

A.1 Supplemental Methods for Chapter 2

A.1.1 Maximum Likelihood cut heuristic

Maximum-Likelihood-Cut(H,R)

Initialization: $c = 0,\ S^* = \varnothing$

Iteration: for $i = 1 \ldots M$:

1. Chose a pair of vertices (u, v)
2. Initialize $S_1 = \{u\}$, $S_2 = \{v\}$ and $S = S_1 \cup S_2$
3. **While** $|S| < |V|$
   (a) Let $w' = \arg\max_{w \in V-S} |L(w)|$.
   (b) **If** $L(w') < 0$, $S_1 = S_1 \cup \{w'\}$
   (c) **else if** $L(w') > 0$, $S_2 = S_2 \cup \{w'\}$
   (d) **else** add w' uniformly at random to S_1 or S_2
4. **If** $p(R|q, H(S_1)) > p(R|q, H(S^*))$
   (a) $S^* = S_1$, $c = 0$

   else $c = c + 1$
5. If $c > C$: break

Return: S^*

A.1.2 Implementation of HapCUT2

HapCUT2 operates on each connected component or haplotype block of the read data to search for good haplotypes iteratively using the maximum-likelihood-cut heuristic. Each

fragment is stored as a list of blocks that cover consecutive variants. This compact representation is efficient at storing long reads as well as paired-end reads that span a large number of variants. For short read data, HapCUT2 stores all pairs of edges corresponding to each fragment in the read-haplotype graph. However, for long read datasets, HapCUT2 reduces the number of edges in the graph by only storing edges for adjacent variants in each fragment. This is sufficient for determining the connected components in the read-haplotype graph and also for selecting an edge to initialize the cut in the Maximum-Likelihood-Cut heuristic. Note that HapCUT2 still has to consider all pairs of edges per fragment in order to search for good cuts. Therefore, the computational complexity of HapCUT2 scales as V^2 where V is the maximum number of variants covered by a fragment.

The first step in the Maximum-Likelihood-Cut heuristic is to select a pair of vertices to initialize the cut. In the original HapCUT method [39], edges were selected at random from the read-haplotype graph to initialize the cut. This requires a large number of iterations (proportional to number of edges in the graph) to ensure that 'good' edges are considered. An alternate greedy approach (also used in the RefHap algorithm [41]) is to identify edges such that the current phase between the pair of vertices (as defined by the haplotype H) is highly inconsistent with the fragment data. Such edges can be found by sorting the list of edges in the read-haplotype graph by weight and selecting the lowest weight edges. HapCUT2 uses a hybrid approach (combination of K lowest weight edges (default $K = 5$) and $\min(N/10, 100)$ randomly sampled edges where N is the number of variants in the block) to initialize the cut in the maximum-likelihood-cut heuristic. Therefore, the maximum number of iterations for the maximum-likelihood-cut routine is $M = K + \min(N/10, 100)$.

A.1.3 Likelihood-based variant pruning

Following haplotype assembly, HapCUT2's variant pruning scheme makes a single pass over the haplotype H in which it considers each variant i of H separately. The goal is two-

fold: firstly, if the likelihood of the haplotype can be improved by changing the haplotype or genotype assignment at i, then the allele or genotype is reassigned. Secondly, if the haplotype is low confidence at i even after being reassigned, position i is pruned from the solution. While considering position i, it is assumed that that the rest of the haplotype $H[i' \neq i]$ is correct. At variant i, each of four possibilities are considered for the two alleles in the ordered pair of haplotypes: $\{10, 01, 11, 00\}$. Let $H_{i \to x}$ denote H that has been modified to have phasing $x \in \{10, 01, 11, 00\}$ at position i. Optionally, HapCUT2 supports obtaining the prior probabilities of the unordered genotype configurations $(00), (01), (11)$ from the VCF genotype likelihoods. The results obtained in this paper used the default behavior, which sets prior probabilities of 0 for the homozygous configurations (00) and (11), such that genotype calls are assumed to be correct. The prior probabilities for the two haplotype configurations (01) and (10) are set to be equal. Then, the posterior probability of each possibility can be calculated as:

$$P(H_{i \to x} | q, R, H) = \frac{p(x) p(R | q, H_{i \to x})}{\sum_{y \in \{10,01,11,00\}} p(y) p(R | q, H_{i \to y})} \tag{A.1}$$

The posterior probability of the most likely configuration is:

$$P_H[i] = \max_{x \in \{10,01,11,00\}} P(H_{i \to x} | q, R, H) \tag{A.2}$$

For a given position, the haplotype is assigned to the configuration that maximizes $P_H[i]$. If $P_H[i] < \alpha$ for some user-defined threshold $\alpha \in [0.5, 1]$, the variant is pruned from the final result. The default value of α is 0.8. HapCUT2 also offers the RefHap heuristic as an alternative to the likelihood based method, which may be preferable when quality scores are not accurate.

A.1.4 Block Splitting

HapCUT2 includes an optional scheme for splitting blocks at low-confidence sites. Let $H_{s(i)}$ denote H that has been edited to have a switch starting at position i. That is, every position

from i onwards is flipped with respect to those before i. Similarly to before, we assume that $H[1..(i-1)]$ and $H[i..N]$ are correct.

$$P(H_{s(i)}|q,R,H) = \frac{p(R|q,H_{s(i)})}{p(R|q,R,H) + p(R|q,R,H_{s(i)})} \tag{A.3}$$

Under the assumption that $H[1..(i-1)]$ and $H[i..N]$ are correct, this is equivalent to computing a Bayesian posterior probability of a switch at i with equal priors. After computing the posterior probability of each phasing, a block is split at i if $1 - P(H_{s(i)}|q,R,H) < \alpha_2$ for some user-defined threshold $\alpha_2 \in [0.5, 1]$

A.1.5 Estimating $\tau(I)$ for Hi-C reads

In order to properly model h-trans error we must know the probability that a read pair with insert size I is h-trans, i.e. the two ends of the paired-end read originate from different homologous chromosomes. Selvaraj et al. estimated these probabilities using mouse Hi-C data where the haplotypes was known. We estimated the function τ for the NA12878 MboI data using the known trio phase and observed that τ varies from chromosome to chromosome, with certain chromosomes such as chromosome 17 and 19 having rates of h-trans error several times larger than others. It is possible that the rate of h-trans error may also vary across different cell types. Therefore, it would be ideal to estimate the rate of h-trans error directly from the data as a part of the haplotype assembly process.

Assume that we have assembled the haplotypes (H) from the Hi-C reads using HapCUT2-Assemble. Our goal is to estimate the probability $\tau(I)$ that a paired-end read with distance between the two inner ends equal to I represents an h-trans read. We assume that this probability is the same for all reads that have insert size I. For one such read R, let us assume without loss of generality that the haplotype pair for the two variants covered by the read is $(00, 11)$. If the read sequence is also 00 or 11, the read matches the haplotype pair H. This can happen if the read is a

cis-read and has 0 or 2 sequencing errors or if it is a trans-read and has a sequencing error at only one end. The probability of this is:

$$(1-\tau(I))\left[(1-q_1)(1-q_2)+q_1q_2\right]+\tau(I)\left[(1-q_1)q_2+(1-q_2)q_1\right]=(1-\tau(I))a+(\tau(I))(1-a)$$

where q_1 and q_2 are sequencing error probabilities and a is the probability that the read pair has 0 or 2 sequencing errors (at the variant sites). Conversely, if the read sequence at the two variants is 01 or 10, the read can be either (i) a cis-read with one sequencing error, or (ii) a trans-read with 0 or 2 sequencing errors. The likelihood of the read in this case is:

$$(\tau(I))a+(1-\tau(I))(1-a)$$

The likelihood of each read can be calculated using the above two expressions. The joint likelihood of all reads with an insert length equal to I is simply the product of individual read likelihoods and is a function of the variable $\tau(I)$. To get a maximum likelihood estimate of $\tau(I)$, we simply find the value of $\tau(I)$ that maximizes this likelihood function. It is not difficult to show that the maximum likelihood estimate of $\tau(I)$ is:

$$\frac{\sum_{R_i,R_i=H} b_i+\sum_{R_i,R_i\neq H} a_i}{N_I} \tag{A.4}$$

where N_I is the total number of reads with insert length I, a_i is the probability that read

pair i has 0 or 2 sequencing errors, and b_i is the probability that read pair i has 1 sequencing error.

HapCUT2-HiC-Mode(R)

Initialization: $H = H^0, H_{old} = H^0$

$\tau(I) = 0$ for all I

Iteration: while $p(R|q,H,\tau) \geq p(R|q,H_{old},\tau)$:

1. $H_{old} = H$
2. $H =$ HapCUT2-HiC-Assemble(R,τ)
3. estimate $\tau(I) = \frac{\sum_{R_i,R_i=H} b_i + \sum_{R_i,R_i \neq H} a_i}{N_I}$ for each insert size I using all reads with insert size I

Return: H

HapCUT2-HiC-Assemble, used as a subroutine by HapCUT2-HiC-Mode, is the exact same as HapCUT2-Assemble described in the main text, except that it incorporates τ into all read likelihood calculations. Therefore, the algorithm works by iteratively assembling a complete haplotype H from the reads and τ using an "h-trans aware" version of the assembly algorithm, estimating a new τ from H, and repeating. Note that the haplotypes assembled by HapCUT2 are expected to have some errors, particularly at low coverage. If we can calculate the posterior probability of the phasing between each pair of variants, we could estimate τ using an exact EM approach. However, the posterior probability of the phase between a pair of variants depends on the errors in the paths in the read-haplotype graph between the two variants and is computationally infeasible to calculate. Nevertheless, we found that this EM-like approach was able to accurately estimate the h-trans error rates at sufficient coverage (Figure A5), and the model consistently improved switch error and mismatch accuracy both at modest coverage levels such as 30× and 40× and high coverage levels such as 90×.

A.1.6 Extraction of haplotype informative reads

Most haplotype assembly algorithms use a haplotype "fragment file" as input. This file represents each haplotype informative read or fragment as a list of heterozygous variants (indices of variants in the VCF file) and the corresponding alleles (with quality values) at each variant site. Consecutive heterozygous variants covered by a single read are compressed to form a single block. This format was first utilized in the assembly of the whole-genome Sanger sequence data for HuRef [32]. The ExtractHAIRs (Extract HAplotype Informative Reads) program was created to process aligned reads in a sorted BAM file and heterozygous variants from a VCF file (with variant calls) to create the haplotype "fragment file". It has been available as part of the HapCUT software package since 2011 and can efficiently process paired-end sequence data as well as long read datasets such as PacBio reads. For analyzing Hi-C data, the fragment file format was modified to store extra information for each read including the data type (Hi-C or long read), variant start index of the second read and paired-end insert size.

For 10X Genomics linked-reads, reads with the same barcode may originate from the same DNA molecule or from several other DNA molecules scattered over the genome. For this reason, reads with the same barcode that were separated by 20 kb or more were called as originating from separate molecules. The molecule boundaries were derived from the aligned BAM file using a python script and the haplotype fragment for each molecule was extracted from the BAM file using the extractHAIRS tool (code available at https://github.com/vibansal/hapcut2).

A.1.7 Post processing of alignments for Hi-C reads

We observed that a significant fraction of the reads ($\sim$10-20%, depending on the experiment) contained a chimeric mate resulting from the ligation junction being located towards the ends of DNA fragment. This resulted in one read reading past the ligation point of the Hi-C fragment, making it a chimera of its own sequence and a sequence originating from near the

other read's location. These chimeras appeared in the BWA alignment as primary and secondary alignments. For the purpose of haplotype assembly, it is important to have as much sequence material as possible. For this reason, we repaired chimeric alignments by cutting the chimeric portion and pasting it to the other mate with an added gap. This post-processing script is freely available with HapCUT2. Following Hi-C chimera repair, the single end alignments for each paired end reads were combined and mate information was filled in with samtools fixmate [63]. Subsequently, the BAM files were sorted with samtools sort and PCR duplicates were marked for removal with Picard MarkDuplicates (http://broadinstitute.github.io/picard). Finally, bam files were split by chromosome with BamTools split [117] and converted to HapCUT fragment matrix format using extractHAIRs.

A.1.8 Experiment and Pipeline Management

Processing and experiment pipelines were managed with Snakemake software [75]. A Snakemake snakefile is available with the HapCUT2 software that will reproduce the results of this paper (all main and supplemental figures) from raw online data sources.

A.2 Acknowledgments

Appendix A, in full, is a reprint of the material as it appears in the Supplemental Material of HapCUT2: robust and accurate haplotype assembly for diverse sequencing technologies. Edge, Peter; Bafna, Vineet; Bansal, Vikas. Genome Research, 27(5), pp.801-812. Cold Spring Harbor Laboratory Press, 2017. The dissertation author was the primary author of this paper.

Permission to reuse the article is granted by Genome Research as described at `https://genome.cshlp.org/site/misc/terms.xhtml`.

A.3 Supplemental Figures and Tables for Chapter 2

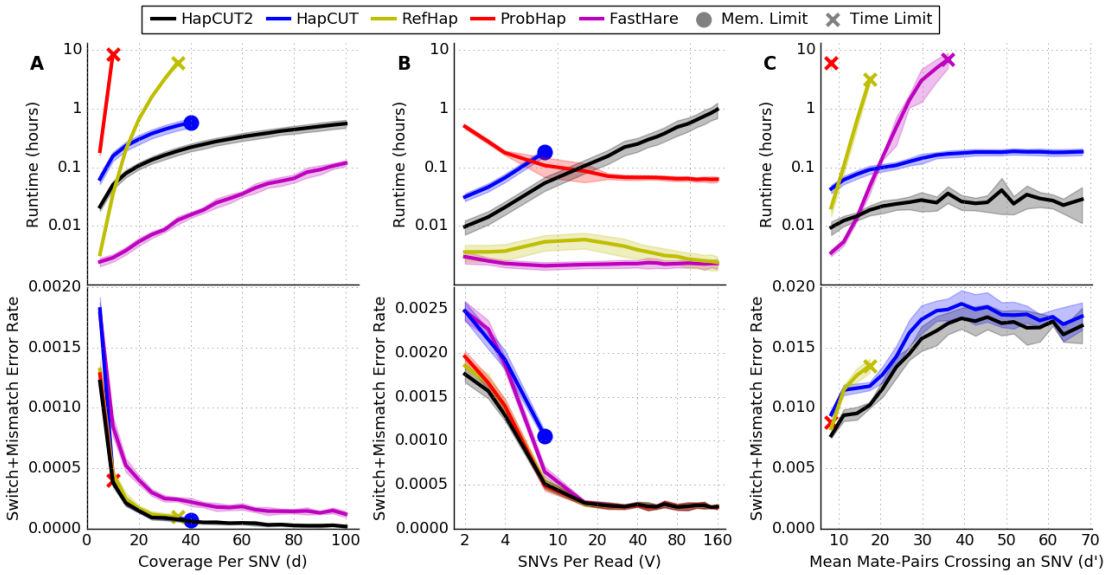

Figure A1: An expanded version of Figure 2.1 with shaded areas added to represent the standard deviation of the 10 replicate experiments. Comparison of runtime (top) and switch+mismatch error rate (bottom) for HapCUT2 with leading methods for haplotype assembly (HapCUT, RefHap, ProbHap, and FastHare) on simulated read data as a function of (A): mean coverage per variant (variants per read fixed at 4), (B): mean variants per read (mean coverage per variant fixed at 5), (C): mean number of paired-end reads crossing a variant (mean coverage per variant fixed at 5, read length 150 base pairs, random insert size up to a variable maximum value). Lines represent the mean of 10 replicate simulations and shaded regions represent the standard deviation. FastHare is not visible on panel C (bottom) due to error rates 10 to 18 times higher than HapCUT2.

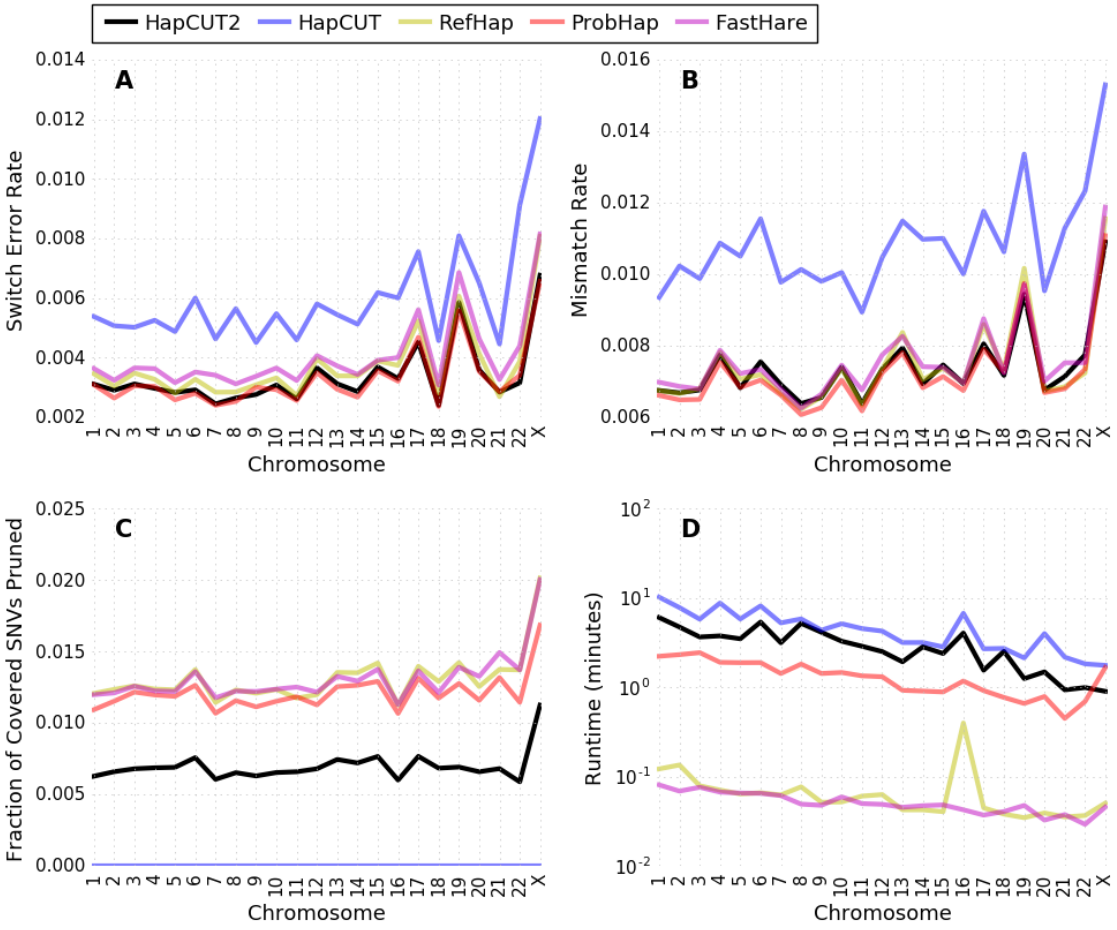

Figure A2: Comparison of the performance of HapCUT2 with other tools on NA12878 fosmid data across all chromosomes: (A) Switch error rate, (B) Mismatch rate, (C) Fraction of covered variants pruned, and (D) Runtime in CPU-minutes. Switch and mismatch errors were calculated using the set of phased variants specific to each tool.

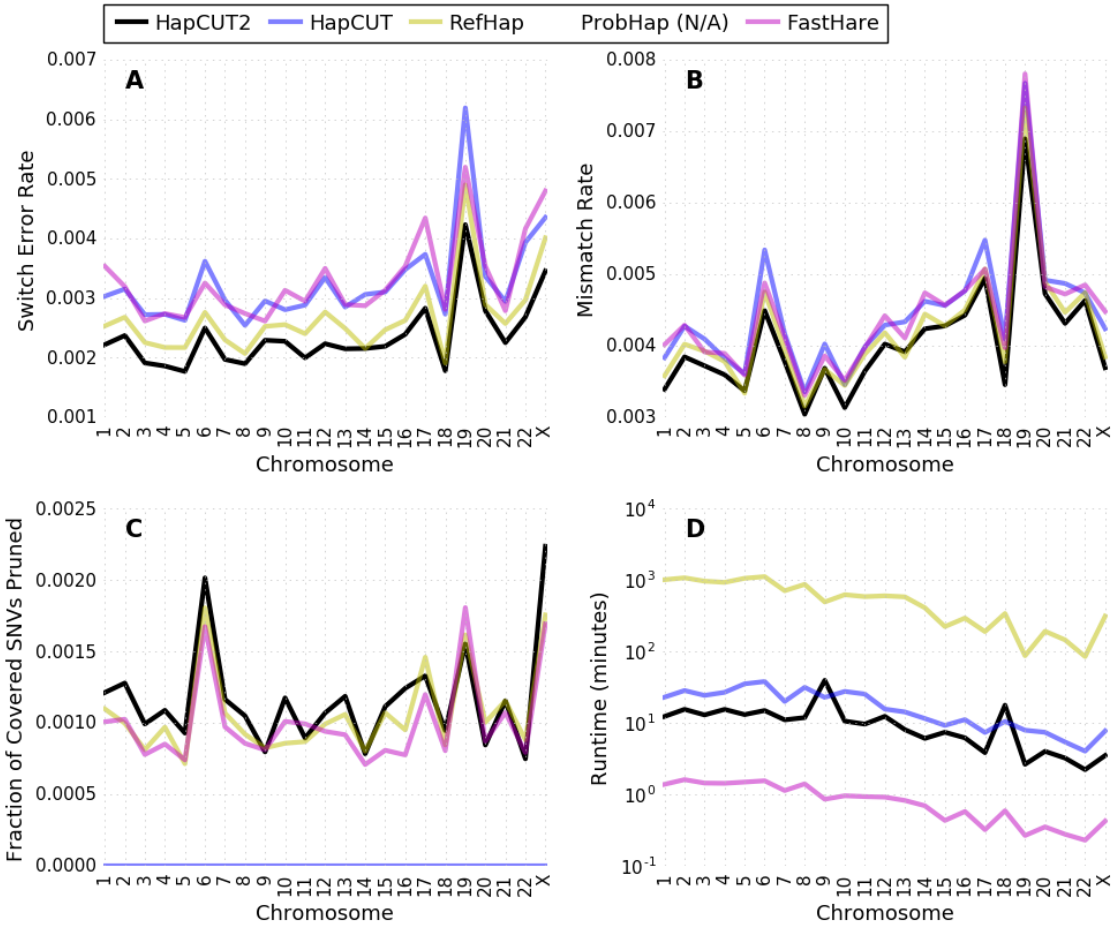

Figure A3: Comparison of the performance of HapCUT2 with other tools on NA12878 44× coverage PacBio SMRT data across all chromosomes: (A) Switch error rate, (B) Mismatch rate, (C) Fraction of covered variants pruned, and (D) Runtime in CPU-minutes. Switch and mismatch errors were calculated using the set of phased variants specific to each tool. ProbHap exceeded 20 CPU-hours for some chromosomes.

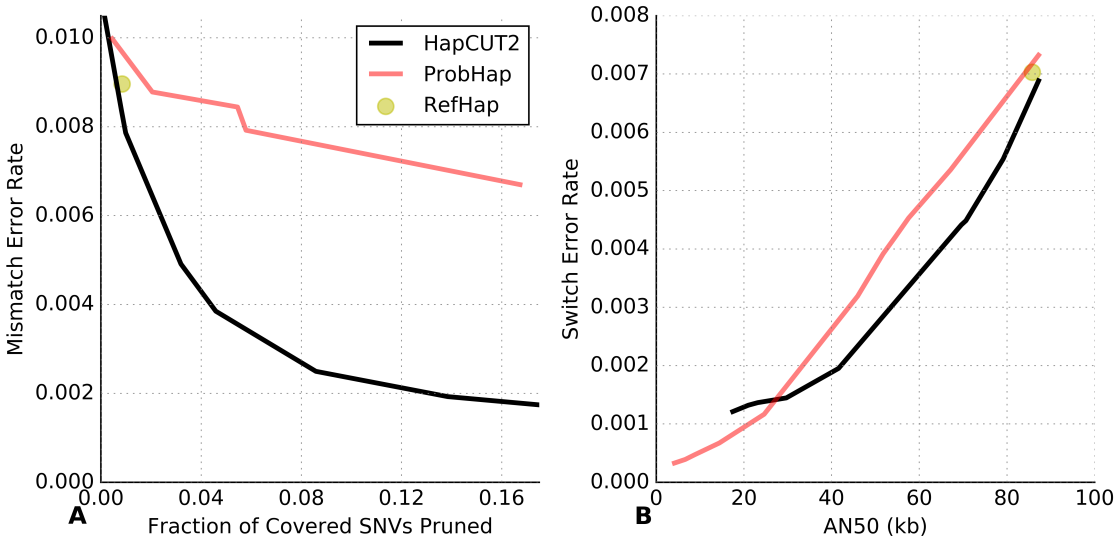

Figure A4: Efficacy of HapCUT2, ProbHap, and RefHaps' post-processing strategies on 11× coverage PacBio SMRT data. (A) Reduction in mismatch error rate by pruning individual low-confidence variants and (B) Reduction in switch error rate by splitting haplotype blocks at possible switch errors (block size represented by the AN50 metric). RefHap is presented as a single point since it does not support variant confidence thresholding or block-splitting.

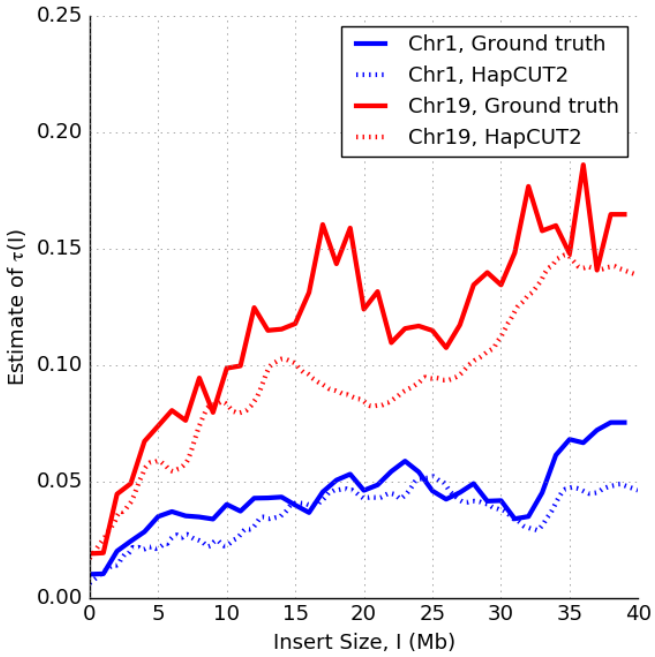

Figure A5: Comparison of the h-trans interaction probabilities $\tau(I)$ estimated by HapCUT2 (for chromosomes 1 and 19 of NA12878 using 90× Hi-C data) against probabilities estimated using knowledge of ground truth haplotypes on the same dataset. Raw HapCUT2 probabilities are smoothed with a Savitzky-Golay filter for visualization, and ground truth probabilities are created using 1 Mb bins.

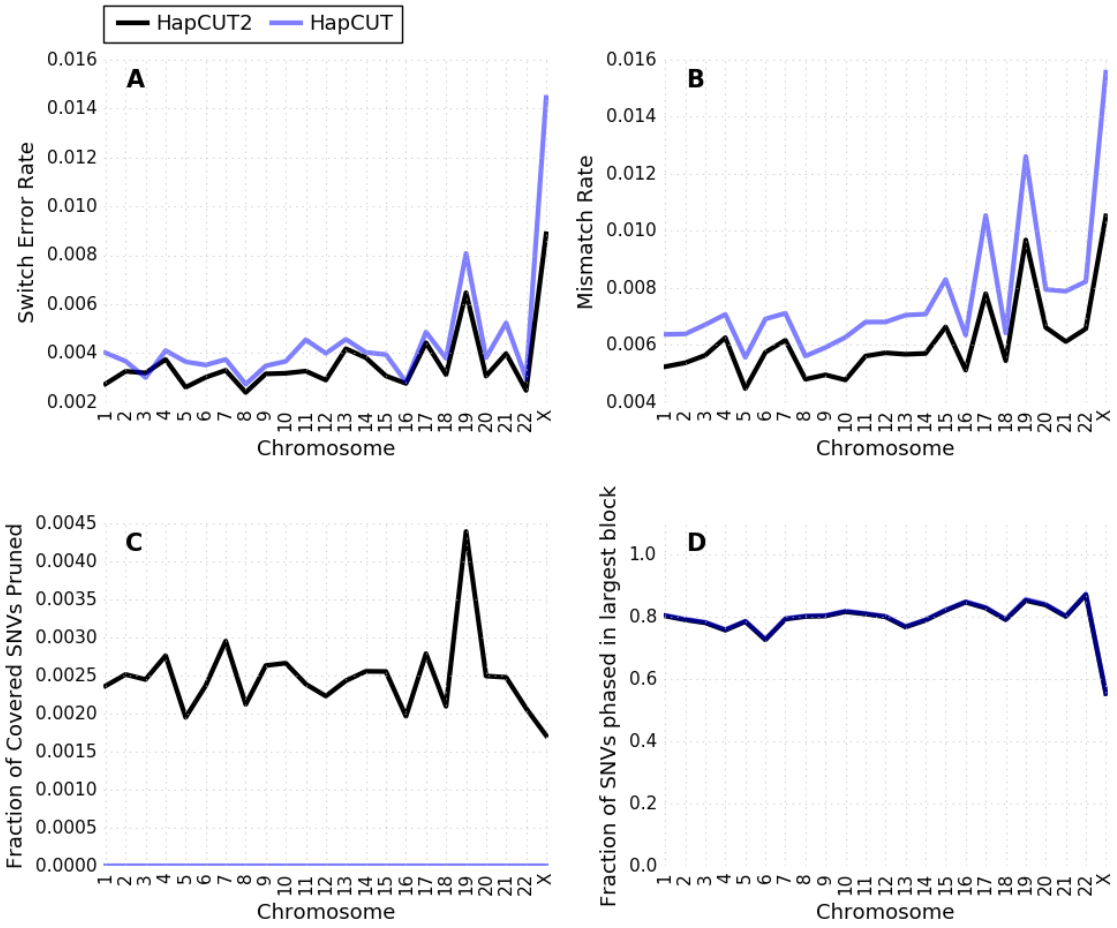

Figure A6: Comparison of HapCUT2 to HapCUT on 90× coverage MboI Hi-C data. (A) Switch error rate (B) Mismatch rate (C) Fraction of covered variants pruned (D) Fraction of variants phased in largest block. Switch and mismatch errors reported here are for all variants phased by a given tool, so the fraction of variants pruned is also reported.

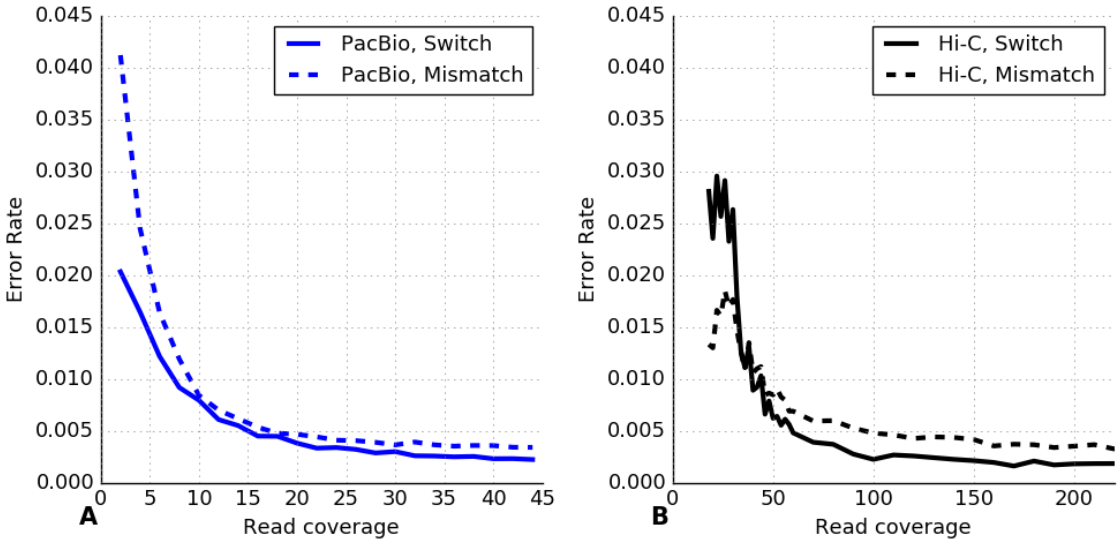

Figure A7: Haplotype accuracy (switch and mismatch error rates) for the NA12878 genome as a function of sequence coverage for (A) PacBio SMRT data and (B) Hi-C (MboI enzyme) data.

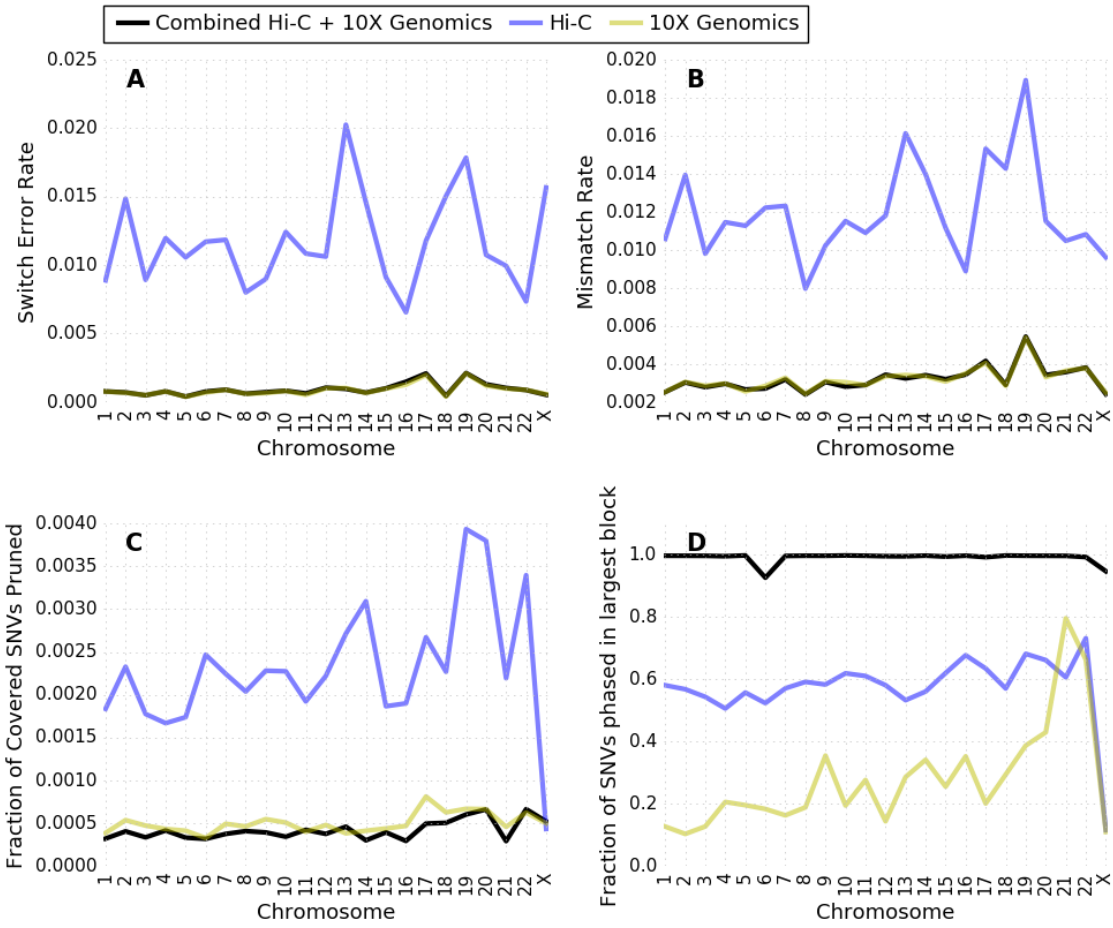

Figure A8: Comparison of haplotypes obtained using HapCUT2 on 40× coverage MboI Hi-C reads combined with 10X Genomics linked-reads (34× short-read coverage) to the haplotypes obtained using the two datasets separately: (A) Switch error rate, (B) Mismatch rate, (C) Fraction of covered variants pruned, and (D) Fraction of variants phased in largest block.

Table A1: Comparison of the total runtime to phase whole-genome Hi-C data (format in hours:minutes) using HapCUT and HapCUT2. To assess the impact of the trans-error probability (τ) modeling on run time, HapCUT2 was run in three different ways: default (no trans-error modeling), Hi-C mode with a pre-computed τ model and Hi-C mode that estimates τ from the data.

Method	**Hi-C (30×)**	**Hi-C (40×)**	**Hi-C (90×)**
HapCUT	0:36	0:46	1:49
HapCUT2 (no trans-errors)	0:10	0:15	0:34
HapCUT2 (fixed τ model)	0:34	0:52	2:02
HapCUT2 (estimate τ from data)	3:05	4:38	9:11

Appendix B

Supplemental Material for Chapter 4

B.1 Supplemental Methods for Chapter 4

B.1.1 Simulating a diploid genome

In order to simulate Illumina and PacBio reads, a diploid genome was generated from the hg19 reference genome (chromosomes 1-22). For simplicity, only SNVs were simulated. Heterozygous SNVs were placed uniformly at random at a rate of 0.001 and homozygous SNVs were placed uniformly at random at a rate of 0.0005. The SNV alleles were selected according to the genotype priors used by Li et al [8]. The phase for heterozygous genotypes was selected uniformly at random. Two fasta sequences for each chromosome were generated using the bcftools consensus command to insert the alleles at SNVs from each haplotype into the reference sequence.

B.1.2 Estimating coverage from aligned reads

For both Illumina and PacBio whole-genome read datasets, the median coverage for each dataset was measured by sampling 100,000 random positions from the genome using `bedtools random` [114] and counting the number of aligned reads covering each position (passing samtools flag filter `3844`). Then, the median value of the measured coverages was taken. For the simulated data, the median coverage was not measured, and instead the reported coverage is the read coverage that was generated for the simulation.

B.1.3 Identification of candidate SNVs

Longshot uses a simple (and standard) genotyping model for identifying candidate SNVs. For each position on the reference, the read bases piled up over that position are considered. The most frequent non-reference base is selected and denoted as the alt (1) allele. The three genotypes $G \in \{0/0, 0/1, 1/1\}$ are considered. Let $\mathbf{A}$ be the vector of allele observations over $\{0, 1\}$.

Using Bayes' rule,

$$p(G|\mathbf{A}) = \frac{p(\mathbf{A}|G)p(G))}{\sum_G p(\mathbf{A}|G)p(G)} \tag{B.1}$$

The probability of the observed pileup alleles is the product of their respective probabilities:

$$p(\mathbf{A}|G) = \prod_{a \in \mathbf{A}} (a|G) \tag{B.2}$$

$$p(a|G) = \begin{cases} 1-\varepsilon & \text{if } a = G_0 = G_1 \\ \varepsilon & \text{if } a \neq G_0 = G_1 \\ \frac{1}{2}(1-\varepsilon)+\frac{1}{2}\varepsilon & \text{otherwise} \end{cases} \tag{B.3}$$

ε is the probability of a sequencing error to a specific base. Since quality values are of limited use for SMS reads, we use our own estimated base-mismatch emission scores for this value (see the subsection 'alignment parameter estimation'). We can compute the probability of a non-reference genotype as:

$$p(0/1|A) + p(1/1|A) = 1 - p(0/0|\mathbf{A})$$

B.1.4 Finding non-repetitive anchors

For an SMS read overlapping a potential SNV site, non-repetitive anchor sequences to the left and right of the potential SNV site are identified to perform local realignment. For this, Longshot searches leftward (and rightward) from the SNV site to find a sequence of length k (default value 6) in the reference sequence where the aligned SMS read matches the reference exactly. This implies that the k-mer in the SMS read was likely sequenced from the template without error and is aligned correctly. This assumption may not hold when the reference sequence is repetitive. For example, consider a 6-mer AAAAAA that matches between reference and SMS

read perfectly. This may be a good anchor sequence if it is the only occurence of AAAAAA in the nearby reference sequence. It is a bad anchor if this is not the case; for instance, if it occurs inside an even larger homopolymer run of A's. We circumvent this issue by ignoring any potential anchor that occurs more than once on the reference sequence within the maximum anchor search window (100 bp by default). The rust-bio implementation of the BNDM algorithm is used to quickly perform this k-mer search[115, 118]. If the leftward or rightward anchor search exceeds half the size of the maximum anchor search window, then that position on the reference and read is used as the anchor regardless of any other factors. This means that in the worst case with default parameters, a realignment window of 100 bp will be formed around the potential SNV. In order to avoid forming realignment windows at a locus with large gaps, if a insertion/deletion/refskip event of length ≥ 20 is encountered near the window, the site is not realigned.

B.1.5 Pair-HMM realignment for clusters of SNVs

When multiple potential SNVs are located in close proximity, the realignment approach should consider the alternate haplotypes defined by these SNVs jointly rather than perform the local realignment for each SNV independently. This is especially important for false potential SNVs – it is common for a single true SNV to be misaligned in the original BAM so that the pileup-based scan identifies it as two or three potential heterozygous SNVs within a few bp of each other. Therefore, we use a simple approach to merge nearby SNVs into SNV clusters. For every pair of adjacent potential SNVs, we merge them into the same SNV cluster (and merge their realignment windows) if their realignment window boundaries overlap. For a cluster with n potential SNVs, we use the pair-HMM forward algorithm to realign against each of the 2^n possible short-haplotype sequences, which we will refer to as the set $\mathcal{H}$. For example, the possible haplotypes in the case of three potential SNVs are $\mathcal{H} = \{000, 001, 010, 100, 110, 101, 011, 111\}$ represented in bitstring form for the 3 SNV sites. We then use a Bayesian calculation similar to the single SNV case to calculate the probability of each possible short-haplotype $h \in \mathcal{H}$

$$p(h \mid \text{read}) = \frac{p(\text{read} \mid h)}{\sum_{h' \in \mathcal{H}} p(\text{read} \mid h')}$$

where $p(\text{read} \mid h)$ is calculated using the forward algorithm between the (multi-SNV) read-window and the haplotype sequence obtained by inserting into the reference sequence window each SNV in h. The short-haplotype h_{max} maximizing $p(h \mid \text{read})$ is selected and used to assign the call for each of the n alleles. The quality value q_i for each allele is calculated independently as:

$$q_i = \text{phred}\left(1 - \sum_{h_c} p(h_c \mid \text{read})\right)$$

where h_c is the set of haplotypes in $\mathcal{H}$ for which $h_c[i] = h_{\max}[i]$. In other words, it is 1 minus the sum of probabilities of all short haplotypes sharing the same best allele call in position i. The total computational complexity of all the realignments is $O(m^2 2^n)$, for read and haplotype windows of length m. For computational efficiency, we limit the cluster size (n) to a maximum value (default = 3) and break large clusters into smaller ones.

B.1.6 Priors on genotypes

Longshot uses the same approach as Li et al. [8] to estimate the prior probability of the genotypes. The prior probabilities for each SNV genotype (given the reference base) are derived assuming that heterozygous SNVs occur at a rate of 0.001 and homozygous SNVs occur at a rate of 0.0005. By default, Longshot differs from the Li et al approach in that a transition (Ts) mutation is assumed to occur at the same rate as a transversion (Tv) mutation. The prior probabilities can be specified as parameters to the software.

B.1.7 Haplotyping and measuring accuracy

Longshot produces a phased VCF as output, using the standard 'phase set' (PS) notation to delineate phased haplotype blocks. To assess the accuracy of the haplotypes assembled by

Longshot, the output VCF was first filtered to remove SNVs with a low phase quality (PQ < 30). This value is similar to the genotype quality (GQ), except that it represents the confidence in the most likely phased genotype (0|0,0|1,1|0,1|1). The GQ, on the other hand, combines the heterozygous phased genotypes together to represent the most likely unphased genotype (0/0,0/1,1/1).

A switch error occurs when, at a single SNV, the phase (e.g. 0|1 or 1|0) of the assembled haplotype differs from the ground-truth haplotype with respect to the previously compared SNVs. A switch error indicates that the phase of the SNVs that follow will be different. If two consecutive switch errors occur such that the phase of only one SNV is incorrect, this corresponds to a 'mismatch' error or a short switch error. We calculated the switch and mismatch error rate separately and report a single error rate by adding these two error rates.

To compare the phasing performance of short reads with long reads using Longshot, we used HapCUT2 to assemble haplotypes using short reads and variants called using FreeBayes on the same set of reads [65]. Variants identified from short read WGS can also be paired with long read data to assemble long haplotypes [68]. This differs from Longshot, which requires no prior knowledge of SNVs (and genotypes) to assemble haplotypes. We compared the phasing accuracy of Longshot with this composite approach. For this, SNVs called using 30× Illumina WGS with the previously described filters were used with the extractHAIRS program to extract haplotype fragments in the PacBio reads mode (`--pacbio 1`). Then, the fragments were used to assemble haplotypes with HapCUT2 (`v1.1`). The resulting haplotypes were filtered for a minimum mismatch quality (similar to the phase quality described for Longshot) of 30.

B.1.8 Separation of reads by haplotype

Let $H = (H_1, H_2)$ be the final pair of haplotypes output by Longshot. Assuming that the prior probability of a read originating from H_1 or H_2 is equal, the probability that a read r was sampled from haplotype H_1 can be calculated as:

$$\frac{p(r|H_1)}{p(r|H_1)+p(r|H_2)}$$

The read is assigned to H_1 if this probability is at least T (default value = 0.99), assigned to H_2 if the probability is $\leq 1-T$ and left unassigned otherwise.

B.1.9 Alignment Parameter Estimation

In order to estimate allele probabilities using the pair-HMM, it is necessary to know the best alignment parameters for SMS reads. Specifically, the parameters are transition probabilities between every state in {MATCH, INSERTION, DELETION} (with outgoing probabilities for a state summing to 1) as well as emission probabilities for a pair of aligned bases. We use a single emission probability for matched bases, and a single emission probability for mismatched bases. In order to use parameters that accurately reflect the data, we estimate these probabilities directly from the alignments in the bam file. While the bam alignments are too inaccurate for sensitive genotyping, we can expect the alignments to roughly reflect the probabilities of insertion, deletion, and base mismatch errors for the reads. This approach also assumes that variants from the reference genome are significantly less common than read errors, which is true for SMS reads from the human genome. We perform a single scan over every CIGAR string and sequence in the BAM, and transition between MATCH, MISMATCH, and DELETION states according to the CIGAR string. The number of observed transitions from each state to itself or other states is counted and converted into probabilities by dividing by the total transitions out of the outgoing state. The aligned bases at each step in the CIGAR are tracked, and the total number of matching and mismatching bases are used to estimate the emission probability for matched vs. mismatched bases.

B.1.10 Variant calling using Clairvoyante and WhatsHap

We installed Clairvoyante 1.02 using the Bioconda method described on the github (`https://github.com/aquaskyline/Clairvoyante`), but encountered a runtime error related to CPU affinity alteration that we avoided with a small fix to the code, described in this github issue (`https://github.com/aquaskyline/Clairvoyante/issues/27`). We used the pre-trained models available at `http://www.bio8.cs.hku.hk/trainedModels.tbz` that were trained at learning rate 1e-3 for 999 epochs. We used the model trained on NA24385 to call variants on NA12878 and the model trained on NA12878 for other genomes. We ran Clairvoyante using commands of the form:

```
clairvoyante.py callVarBam --chkpnt_fn {model} --ref_fn ref.fa \
--bam_fn ngmlr_alignments.bam --ctgName {chrom} \
--call_fn {chrom}.out.vcf --sampleName {sample_name} \
--threshold 0.2 --minCoverage 4 --threads 4
```

Comparison of precision and recall values between methods requires choosing a threshold for the variant quality. For Clairvoyante, following the authors' approach [99], we used the 0.2 allele frequency cutoff and the variant quality threshold that maximized the F1-score.

We used WhatsHap 0.18 (https://bitbucket.org/whatshap/whatshap) using the potential variants discovered in step 1 of the Longshot algorithm as input. The program was run with the following command for each genome:

```
whatshap genotype --ignore-read-groups --reference ref.fa \
-o {chrom}.out.vcf potential_snvs_{chrom}.vcf input.bam
```

After calling variants, the VCF was filtered using the same maximum coverage filter as Longshot. Both methods were run separately on each chromosome on an Intel Xeon CPU E5-2670 0 @ 2.60GHz.

B.1.11 Variant calling using Nanopolish

We used Nanopolish version 0.11.1 to call variants using the rel6 version of the NA12878 ONT data (https://github.com/nanopore-wgs-consortium/NA12878/blob/master/Genome.md). The raw fastq files were downloaded and aligned with minimap2 to hg38 reference genome. First, we downloaded the individual fast5 files and ran nanopolish index with command of the form:

```
nanopolish index \
-d fast5/Bham/FAB39043-3709921973_Multi \
.... \
-d fast5/Bham/FAB41174-3976885577_Multi \
-s fast5/rel_6_sequencing_summary.txt rel_6.fastq.gz
```

For variant calling, we divided chromosome 20 into chunks of size 1 MB and ran nanopolish on the chunks using commands of the form:

```
nanopolish variants --threads 4 --ploidy 2 -q cpg --window chr20:{start}-{end} --
    reads rel_6.fastq.gz --bam minimap2_alignments.bam --genome hg38.fa --outfile
    chr20.{start}.{end}.vcf
```

After this, the VCF's from the individual chunks were recombined. We note that we ran nanopolish in the methylation aware mode since it yielded better results.

B.2 Acknowledgments

Appendix B, in full, is a reprint of the material as it appears in the Supplemental Material of Longshot enables accurate variant calling in diploid genomes from single-molecule long read sequencing, 2019. Edge, Peter; Bansal, Vikas. Nature Communications, 10(1), pp.1-10. Nature Publishing Group, 2019. The dissertation author was the primary author of this paper.

B.3 Supplemental Figures and Tables for Chapter 4

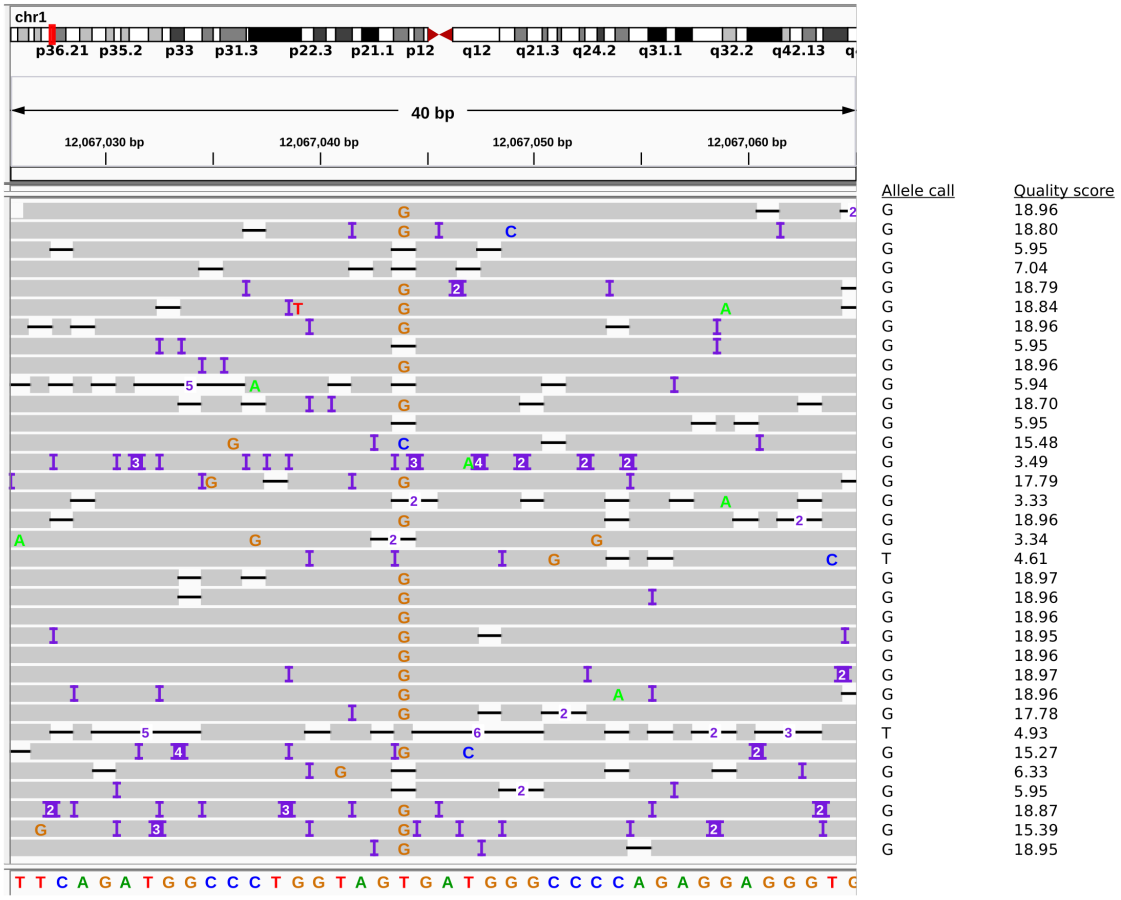

Figure B1: Illustration of reference bias in SMS read alignments. PacBio SMS reads covering a homozygous SNV (chr1:12,067,044 T→G) site in the individual NA12878 are shown (visualized using IGV). Frequent indel errors in the SMS reads, combined with a bias in the alignment algorithm to favor the reference allele, results in 3 reads that contain the reference allele at the SNV site, 1 read containing a C, and 9 reads that contain a deletion. As a result, the variant can incorrectly be called as a heterozygous SNV. Realigning each read to both the reference allele and the alternate allele and selecting the most likely alignment can ameliorate this bias. To the right of each read, the most likely allele call and quality value calculated by Longshot using the Pair-HMM realignment strategy is shown. All reads, except two reads with very low quality values (< 5), support the 'G' allele resulting in the correct homozygous SNV call.

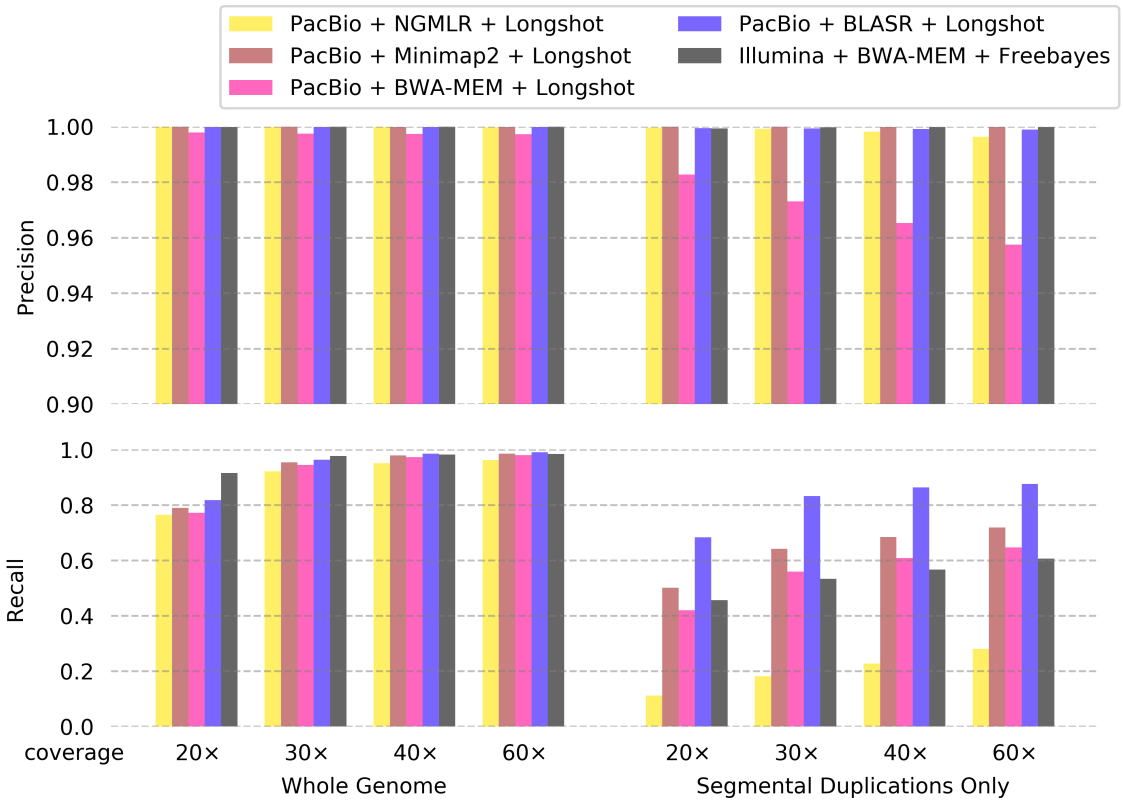

Figure B2: Comparison of precision and recall of SNV calling using different long-read mapping tools. Simulated Illumina and PacBio reads were generated at multiple coverages (20×, 30×, 40× and 60×) from chromosomes 1-22 (hs37d5 reference genome) and variants were called using either Longshot (PacBio) or FreeBayes (Illumina). SMS reads were mapped with multiple mapping tools (NGMLR, BWA-MEM, MINIMAP2, and BLASR) for comparison. Precision **(top)** and Recall **(bottom)** of called variants were assessed across the entire chromosome **(left)** and within segmental duplications with 95% or greater sequence identity **(right)**.

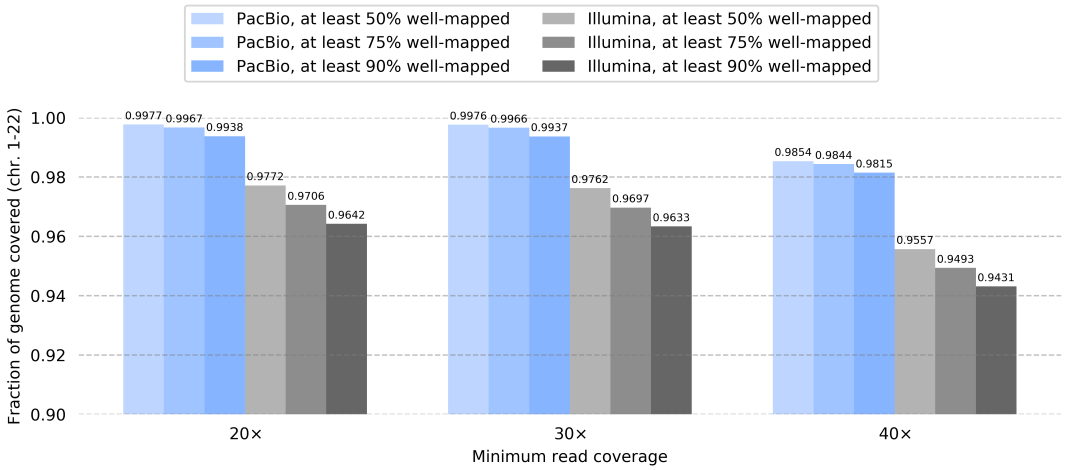

Figure B3: Comparison of the mappability of short reads with long reads using simulated data. Illumina short reads and PacBio SMS long reads were each simulated at 60× coverage and mapped to the genome with BWA-MEM and BLASR, respectively. For every position in the genome, the coverage of primary read mappings was assessed. The positions in the genome were filtered for those with at least 20×,30×, and 40× coverage of primary read mappings. Of those positions, it was determined what fraction of the mappings were "well-mapped", or passing standard filters and having MAPQ $\geq$ 30. The number of positions meeting the minimum coverage cutoff and also meeting a minimum "well-mapped read" cutoff of at least 50%, 75%, 90% are shown as a fraction of total genomic positions (excluding "N" positions).

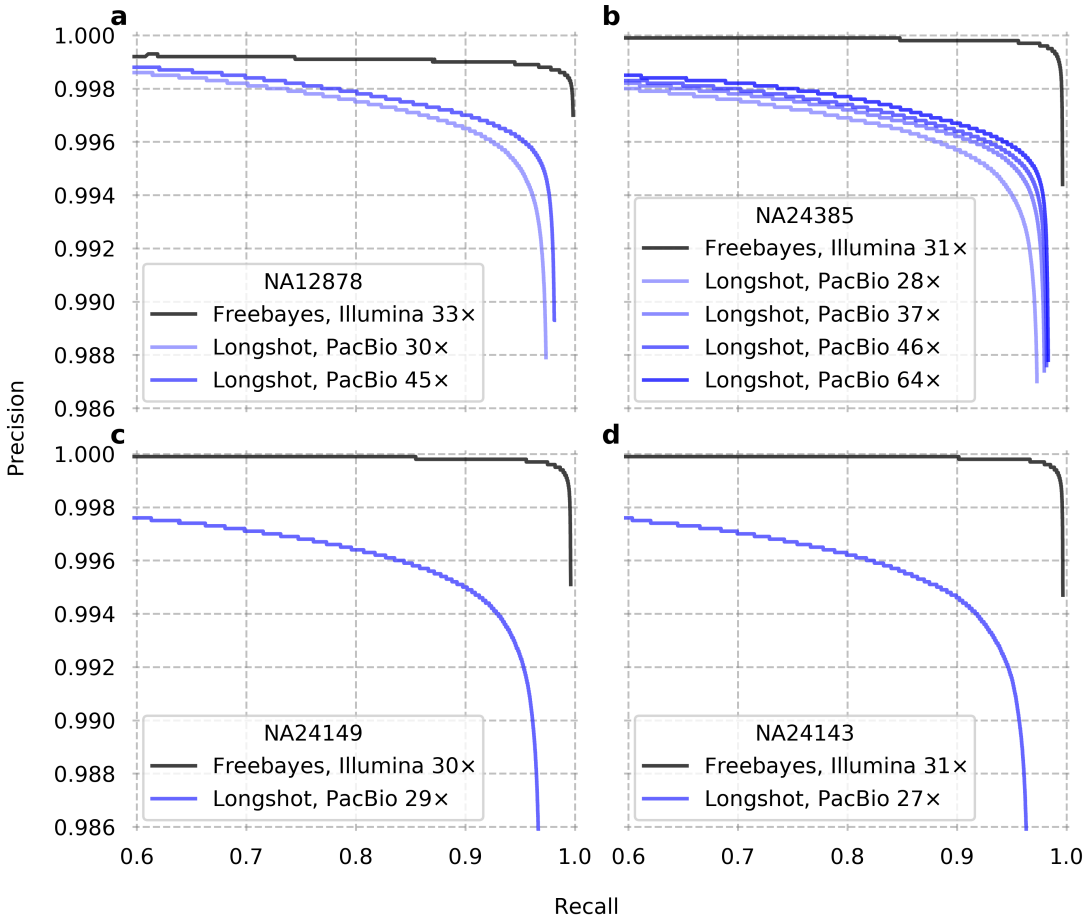

Figure B4: Precision-Recall curve for SNV calling on four individuals: (a) NA12878, (b) NA24385, (c) NA24149, and (d) NA24143. For each individual, variants were called from Illumina short reads using FreeBayes and from whole-genome PacBio SMS reads using Longshot. Precision and recall were calculated using GIAB SNV calls within high-confidence regions. Points on each curve are obtained by varying the minimum Genotype Quality (GQ) cutoff.

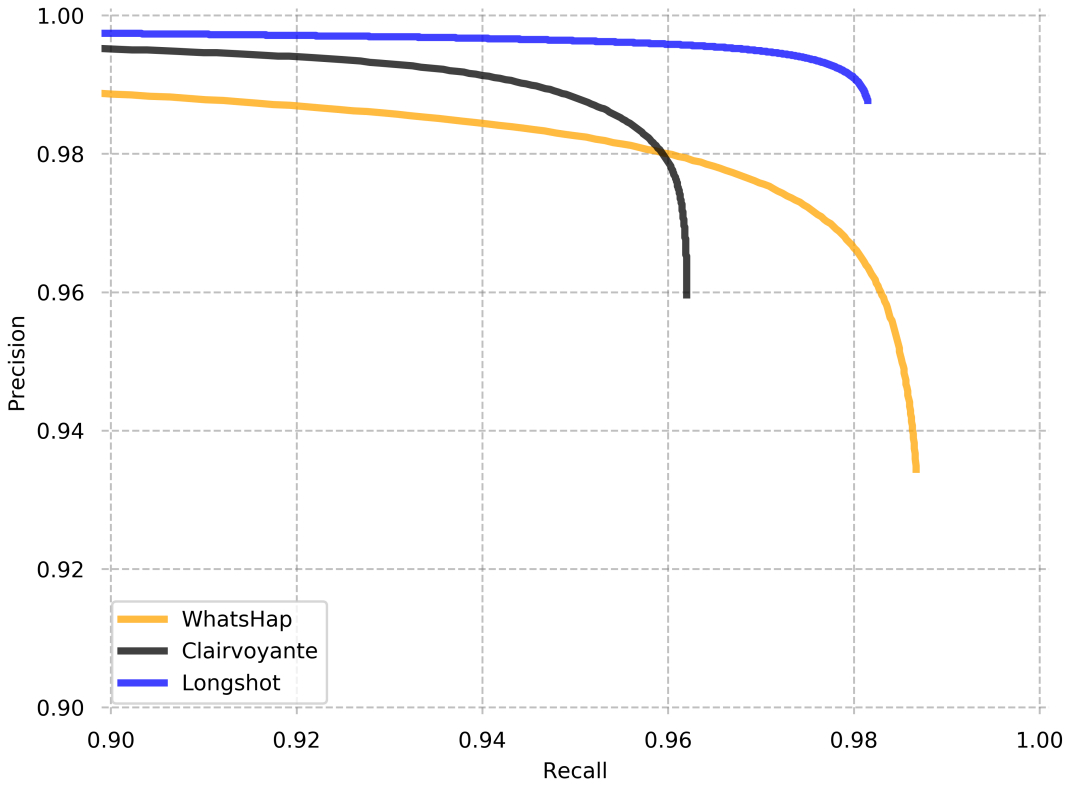

Figure B5: Comparison of precision and recall of SNV calling using different variant calling methods (Longshot, Clairvoyante and WhatsHap), on the NA12878 PacBio dataset. Reads aligned using the NGMLR tool were used for variant calling using each method. Precision and recall were calculated using GIAB SNV calls within high-confidence regions.

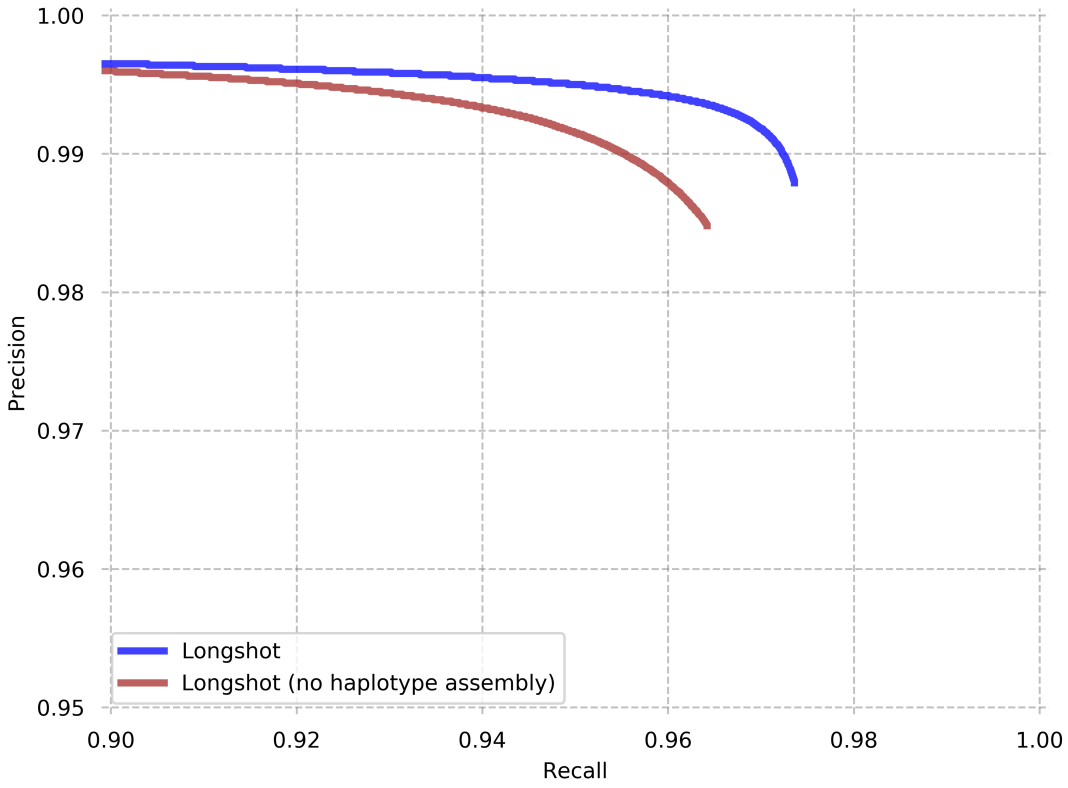

Figure B6: Precision-Recall Curves for Longshot with and without phased genotyping. Longshot was run as normal on the whole-genome NA12878 PacBio data (downsampled to 30× coverage), as well as with "no haplotype assembly" (skipping step 3 of the algorithm).

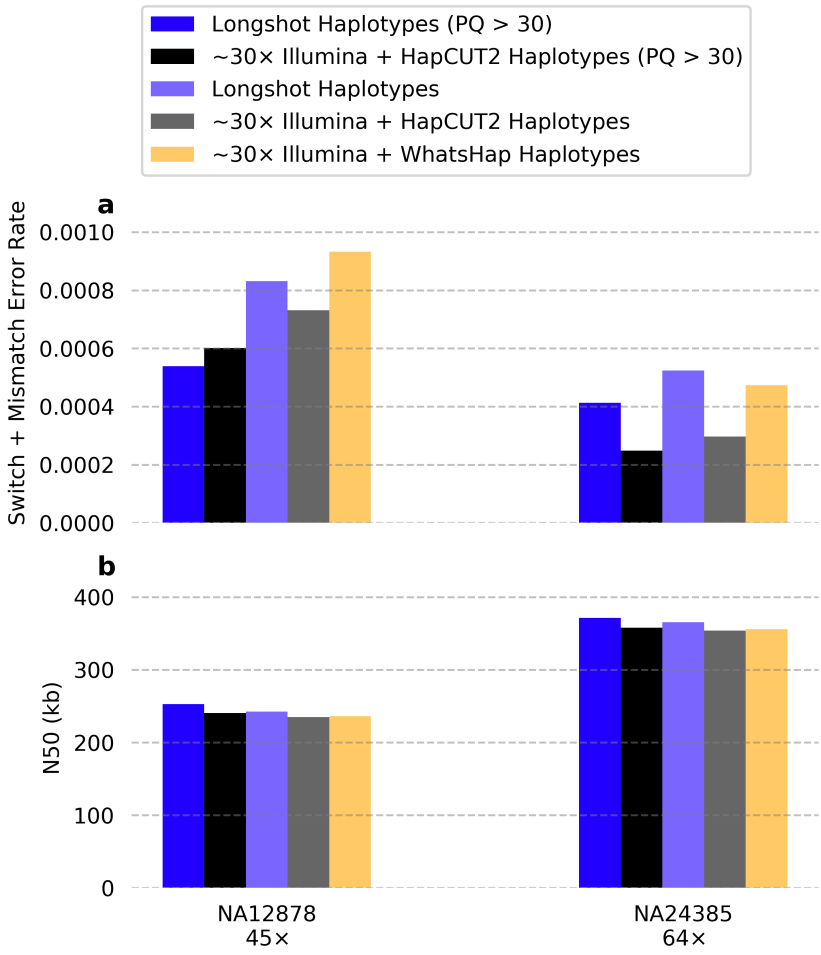

Figure B7: Comparison of the accuracy of haplotypes assembled with Longshot, HapCUT2 and WhatsHap for two genomes: NA12878 (45×) and NA24385 (64×). For running HapCUT2 and WhatsHap, variants identified from ∼30× coverage Illumina sequencing were used as input for phasing. The accuracy of the resulting haplotypes **(a)** was measured using the combined switch error rate (long switches and mismatches), and the completeness **(b)** was measured using the N50 length of the haplotypes. Results are also shown for Longshot and HapCUT2 after filtering for SNVs with Phase Quality (PQ) greater than 30.

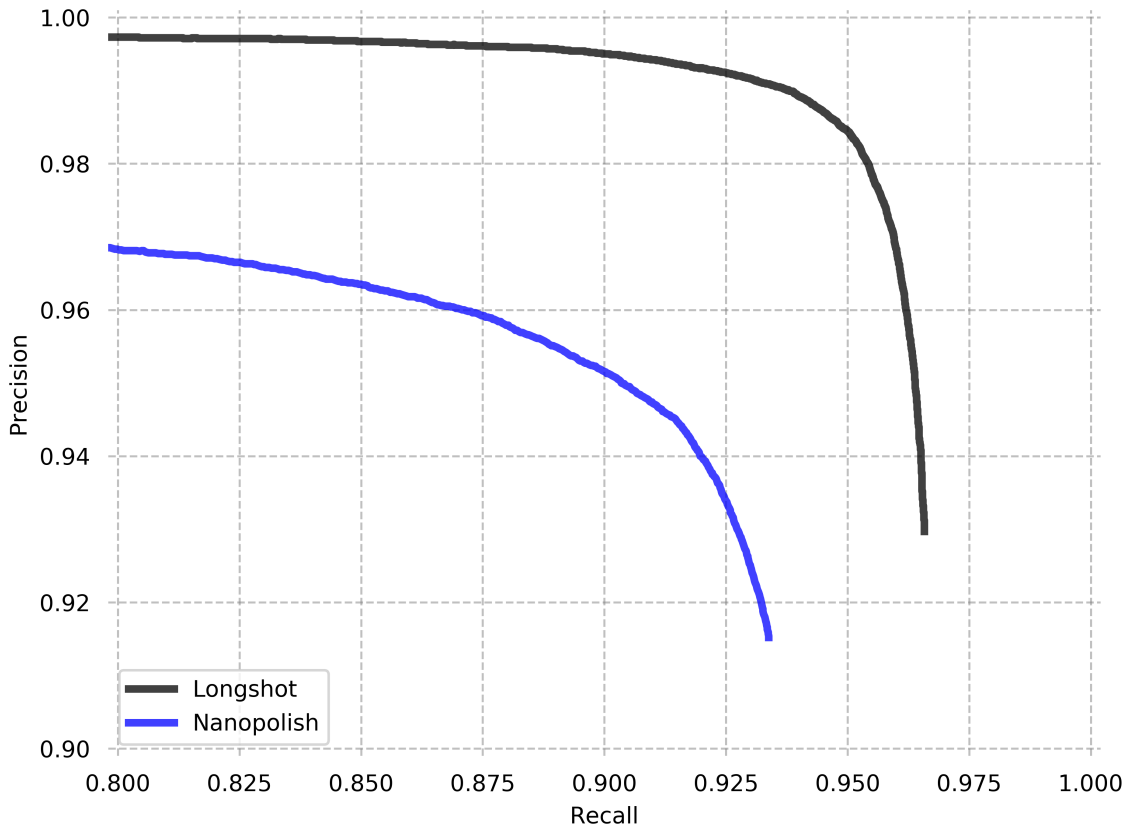

Figure B8: Precision-Recall curve for SNV calling using whole-genome Oxford Nanopore data for NA12878 ($\sim$ 37$\times$ coverage). Precision and recall were calculated using GIAB SNV calls within high-confidence regions of the genome. Points on the curve were obtained by varying the cutoff of the Genotype Quality (GQ) score for Longshot and the QUAL score for Nanopolish. Results shown are based on chromosome 20 only.

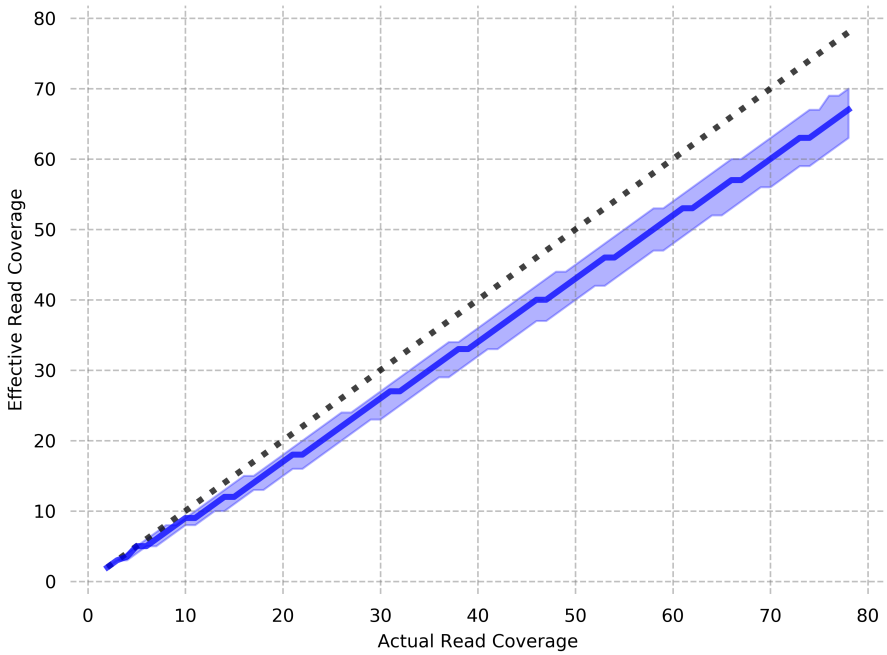

Figure B9: Actual vs effective read coverage in PacBio SMS data. At each SNV site, alleles for which the quality value estimated by the allelotyping method in Longshot is lower than a threshold (default = 7) are discarded; this reduces the effective read coverage. The data shown here is for SNV sites from the 45× coverage NA12878 PacBio dataset (chromosome 1 only). The median effective read coverage (as well as 1st and 3rd quartiles) is plotted for variant sites with the same actual read coverage. Source data is provided as a Source Data file.

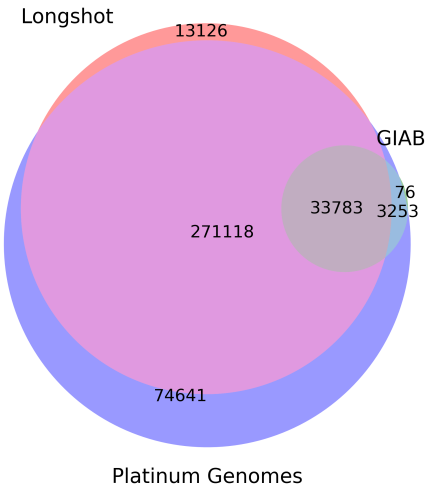

Figure B10: Comparison of Platinum Genomes variant calls (outside GIAB confident regions) with Longshot variants for NA12878. Variants were called on chr1-22 with Longshot using 45× coverage PacBio SMS reads for NA12878. The Longshot variants, GIAB variants, and Platinum Genomes variants were each filtered for regions that are outside GIAB confident regions, but inside the Platinum Genome called regions. 79.6% of SNVs in these difficult-to-call regions are shared between the Platinum Genomes calls and the Longshot calls.

Table B1: Summary of SNVs called using Longshot on whole-genome PacBio SMS data for multiple individuals. Only variants called on the autosomes (chromosome 1-22) are reported. For the NA24385 and NA12878 individuals, variants were called at multiple levels of coverage by downsampling. Precision and recall were calculated inside GIAB high-confidence (GIAB-HC)) regions using the GIAB variant set for each individual as the ground truth. Longshot was run separately on each chromosome each using a single CPU core of an Intel Xeon CPU E5-2670 0 @ 2.60GHz. The run-time is the total measured walltime to process all chromosomes (sum of all individual chromosome walltimes). The mean memory (averaged across chromosomes), and also the maximum memory usage (across chromosomes) are also provided.

Genome	Read Coverage	SNVs called	Precision	Recall	SNVs outside GIAB-HC regions	Run time (hours)	Memory (mean, GB)	Memory (max, GB)
NA12878	30×	3518530	0.994	0.959	515870	27:31	3.23	4.99
NA12878	45×	3567475	0.995	0.973	520375	36:41	3.71	5.76
NA24385	28×	3585223	0.993	0.961	660938	38:30	3.26	5.08
NA24385	37×	3628953	0.993	0.973	666769	47:03	3.66	5.73
NA24385	46×	3642443	0.993	0.977	666733	68:40	3.87	5.71
NA24385	64×	3651477	0.994	0.980	666447	92:48	4.56	7.00
NA24149	29×	3529531	0.992	0.952	776106	37:32	3.25	4.76
NA24143	27×	3528836	0.992	0.945	772344	35:22	3.16	4.66

Table B2: Fractions of False Positive (FP) and False Negative (FN) variant calls that were misgenotyped or coincide with genomic features. Variants were called using Longshot on chr1 using 45× coverage SMS reads for NA12878. The FP and FN variants were determined with respect to the ground truth (GIAB variant set within GIAB confident regions). In order to reason about why the FPs and FNs occurred, the fraction of those variants that fall into different categories was counted. For comparison, 100,000 random positions from chr1 were selected, filtered by the GIAB confident regions and subjected to the same analysis. The fold enrichment compared to the random positions is shown as a measure of the significance.

Genome	False Positives (FP)	FP Enrichment	False Negatives (FN)	FN Enrichment
Misgenotyped SNV	0.050	-	0.014	-
Near Indel	0.714	176.99	0.053	13.17
In homopolymer	0.323	5.70	0.195	3.44
In homopolymer but not near indel	0.025	0.46	0.163	2.94
In STR	0.008	27.90	0.002	7.20
In LINE	0.286	1.32	0.215	0.99
In SINE	0.122	0.81	0.221	1.46

Table B3: Improvement in variant precision by filtering out SNVs near known indel variants. Most false positive (FP) variants called with Longshot occur at true indel variant sites that are mistaken as SNVs. Filtering out SNVs occuring within 5 bp of known indel sites (Mills + 1000G Indel variant set from the GATK bundle) results in improved precision at the cost of a small reduction in recall.

Genome	Read Coverage	Precision (known indels not filtered)	Precision (known indels filtered out)	Recall (known indels not filtered)	Recall (known indels filtered out)
NA12878	45×	0.995	0.997	0.974	0.969
NA24385	64×	0.994	0.996	0.979	0.974
NA24149	29×	0.993	0.995	0.948	0.943
NA24143	27×	0.993	0.995	0.941	0.936

Table B4: SNV calling accuracy for different methods on PacBio and Oxford Nanopore data for NA12878. The precision and recall values for DeepVariant on the PacBio data were obtained from Poplin et al. [92] and based on three chromosomes (20, 21 and 22). For a direct comparison, we calculated the precision and recall for Longshot using the BLASR-aligned bams on these three chromosomes only. The precision/recall for Clairvoyante was obtained from Supplementary Data (Luo et al. [99]) and was based on rel3 release of the ONT data aligned with NGMLR. The accuracy values for MarginPhase were obtained from [93].

Genome	Technology	Method	Precision	Recall
NA12878	PacBio	Longshot	0.9947	0.9701
NA12878	PacBio	DeepVariant	0.9819	0.9739
NA12878	ONT	MarginPhase	0.809	0.769
NA12878	ONT	Clairvoyante	0.9148	0.7518

Bibliography

[1] Human genome project faq. https://www.genome.gov/human-genome-project/Completion-FAQ. Accessed: 2019-10-15.

[2] Evolution of illumina sequencing. https://www.illumina.com/science/technology/next-generation-sequencing/illumina-sequencing-history.html. Accessed: 2019-10-15.

[3] The $1000 genome. https://www.illumina.com/content/dam/illumina-marketing/documents/company/featured-articles/the-1000-dollar-genome.pdf. Accessed: 2019-10-15.

[4] 1000 Genomes Project Consortium. An integrated map of genetic variation from 1,092 human genomes. *Nature*, 491, 2012.

[5] Michal Janitz. *Next-generation genome sequencing: towards personalized medicine*. John Wiley & Sons, 2011.

[6] Margaret A Hamburg and Francis S Collins. The path to personalized medicine. *New England Journal of Medicine*, 363(4):301–304, 2010.

[7] Francis S Collins, Lisa D Brooks, and Aravinda Chakravarti. A dna polymorphism discovery resource for research on human genetic variation. *Genome research*, 8(12): 1229–1231, 1998.

[8] Ruiqiang Li, Yingrui Li, Xiaodong Fang, Huanming Yang, Jian Wang, Karsten Kristiansen, and Jun Wang. SNP detection for massively parallel whole-genome resequencing. *Genome Res.*, 19(6):1124–1132, 2009.

[9] Barkur S Shastry. Snp alleles in human disease and evolution. *Journal of human genetics*, 47(11):561, 2002.

[10] Yaping Yang, Donna M. Muzny, Jeffrey G Reid, Matthew N. Bainbridge, Alecia Willis, Patricia A. Ward, A. Caperton Braxton, Joke Beuten, Fan Xia, Zhiyv Niu, Matthew Thomas Hardison, Richard E. Person, Mir Reza Bekheirnia, Magalie S Leduc, Amelia Kirby, Peter Pham, Jennifer Scull, Min Wang, Yan Ding, Sharon E. Plon, James R Lupski, Arthur L Beaudet, Richard A. Gibbs, and Christine M. Eng. Clinical whole-exome sequencing

for the diagnosis of mendelian disorders. *New England Journal of Medicine*, 369(16): 1502–1511, 2013.

[11] Eric S Lander and Nicholas J Schork. Genetic dissection of complex traits. *Science*, 265 (5181):2037–2048, 1994.

[12] Robert Plomin, Claire MA Haworth, and Oliver SP Davis. Common disorders are quantitative traits. *Nature reviews genetics*, 10(12):872, 2009.

[13] Mark J Chaisson and Pavel A Pevzner. Short read fragment assembly of bacterial genomes. *Genome research*, 18(2):324–330, 2008.

[14] RH Lekanne Deprez, Jose J Muurling-Vlietman, Jarda Hruda, MJH Baars, LCD Wijnaendts, Irene Stolte-Dijkstra, Mariel Alders, and Johanna M van Hagen. Two cases of severe neonatal hypertrophic cardiomyopathy caused by compound heterozygous mutations in the mybpc3 gene. *Journal of medical genetics*, 43(10):829–832, 2006.

[15] Francesca De Luca, Giada Rotunno, Francesca Salvianti, Francesca Galardi, Marta Pestrin, Stefano Gabellini, Lisa Simi, Irene Mancini, Alessandro Maria Vannucchi, Mario Pazzagli, Angelo Di Leo, and Pamela Pinzani. Mutational analysis of single circulating tumor cells by next generation sequencing in metastatic breast cancer. *Oncotarget*, 7(18):26107, 2016.

[16] Niels Van der Aa, Masoud Zamani Esteki, Joris R Vermeesch, and Thierry Voet. Preimplantation genetic diagnosis guided by single-cell genomics. *Genome medicine*, 5(8):71, 2013.

[17] Felix Schmidt and Thomas Efferth. Tumor heterogeneity, single-cell sequencing, and drug resistance. *Pharmaceuticals*, 9(2):33, 2016.

[18] Yusi Fu, Chunmei Li, Sijia Lu, Wenxiong Zhou, Fuchou Tang, X Sunney Xie, and Yanyi Huang. Uniform and accurate single-cell sequencing based on emulsion whole-genome amplification. *Proceedings of the National Academy of Sciences*, 112(38):11923–11928, 2015.

[19] Peter Edge and Vikas Bansal. Longshot enables accurate variant calling in diploid genomes from single-molecule long read sequencing. *Nature communications*, 10(1):1–10, 2019.

[20] J. Duitama, G. K. McEwen, T. Huebsch, S. Palczewski, S. Schulz, K. Verstrepen, E. K. Suk, and M. R. Hoehe. Fosmid-based whole genome haplotyping of a HapMap trio child: evaluation of Single Individual Haplotyping techniques. *Nucleic Acids Res.*, 40(5): 2041–2053, Mar 2012.

[21] Grace X Y Zheng, Billy T Lau, Michael Schnall-Levin, Mirna Jarosz, John M Bell, Christopher M Hindson, Sofia Kyriazopoulou-Panagiotopoulou, Donald A Masquelier, Landon Merrill, Jessica M Terry, Patrice A Mudivarti, Paul W Wyatt, Rajiv Bharadwaj, Anthony J Makarewicz, Yuan Li, Phillip Belgrader, Andrew D Price, Adam J Lowe, Patrick Marks,

Gerard M Vurens, Paul Hardenbol, Luz Montesclaros, Melissa Luo, Lawrence Greenfield, Alexander Wong, David E Birch, Steven W Short, Keith P Bjornson, Pranav Patel, Erik S Hopmans, Christina Wood, Sukhvinder Kaur, Glenn K Lockwood, David Stafford, Joshua P Delaney, Indira Wu, Heather S Ordonez, Susan M Grimes, Stephanie Greer, Josephine Y Lee, Kamila Belhocine, Kristina M Giorda, William H Heaton, Geoffrey P McDermott, Zachary W Bent, Francesca Meschi, Nikola O Kondov, Ryan Wilson, Jorge A Bernate, Shawn Gauby, Alex Kindwall, Clara Bermejo, Adrian N Fehr, Adrian Chan, Serge Saxonov, Kevin D Ness, Benjamin J Hindson, and Hanlee P Ji. Haplotyping germline and cancer genomes with high-throughput linked-read sequencing. *Nat. Biotechnol.*, 34(3): 303–311, Mar 2016.

[22] J. M. Belton, R. P. McCord, J. H. Gibcus, N. Naumova, Y. Zhan, and J. Dekker. Hi-C: a comprehensive technique to capture the conformation of genomes. *Methods*, 58(3): 268–276, Nov 2012.

[23] S. Selvaraj, J. R Dixon, V. Bansal, and B. Ren. Whole-genome haplotype reconstruction using proximity-ligation and shotgun sequencing. *Nat. Biotechnol.*, 31(12):1111–1118, Dec 2013.

[24] Wai Keung Chu, Peter Edge, Ho Suk Lee, Vikas Bansal, Vineet Bafna, Xiaohua Huang, and Kun Zhang. Ultraaccurate genome sequencing and haplotyping of single human cells. *Proceedings of the National Academy of Sciences*, 114(47):12512–12517, 2017.

[25] Smrt sequencing: Read lengths. `https://www.pacb.com/smrt-science/smrt-sequencing/read-lengths`. Accessed: 2018-10-04.

[26] Miten Jain, Sergey Koren, Karen H Miga, Josh Quick, Arthur C Rand, Thomas A Sasani, John R Tyson, Andrew D Beggs, Alexander T Dilthey, Ian T Fiddes, Sunir Malla, Hannah Marriott, Tom Nieto, Justin O'Grady, Hugh E Olsen, Brent S Pedersen, Arang Rhie, Hollian Richardson, Aaron R Quinlan, Terrance P Snutch, Louise Tee, Benedict Paten, Adam M Phillippy, Jared T Simpson, Nicholas J Loman, and Matthew Loose. Nanopore sequencing and assembly of a human genome with ultra-long reads. *Nat. Biotechnol.*, 36 (4):338–345, 2018.

[27] R. Tewhey, V. Bansal, A. Torkamani, E. J. Topol, and N. J. Schork. The importance of phase information for human genomics. *Nat. Rev. Genet.*, 12(3):215–223, Mar 2011.

[28] G. Glusman, H. C. Cox, and J. C. Roach. Whole-genome haplotyping approaches and genomic medicine. *Genome Med*, 6(9):73, 2014.

[29] S. Schiffels and R. Durbin. Inferring human population size and separation history from multiple genome sequences. *Nat. Genet.*, 46(8):919–925, Aug 2014.

[30] M. W. Snyder, A. Adey, J. O. Kitzman, and J. Shendure. Haplotype-resolved genome sequencing: experimental methods and applications. *Nat. Rev. Genet.*, 16(6):344–358, Jun 2015.

[31] S. R. Browning and B. L. Browning. Haplotype phasing: existing methods and new developments. *Nat. Rev. Genet.*, 12(10):703–714, Oct 2011.

[32] Samuel Levy, Granger Sutton, Pauline C Ng, Lars Feuk, Aaron L Halpern, Brian P Walenz, Nelson Axelrod, Jiaqi Huang, Ewen F Kirkness, Gennady Denisov, Yuan Lin, Jeffrey R MacDonald, Andy Wing Chun Pang, Mary Shago, Timothy B Stockwell, Alexia Tsiamouri, Vineet Bafna, Vikas Bansal, Saul A Kravitz, Dana A Busam, Karen Y Beeson, Tina C McIntosh, Karin A Remington, Josep F Abril, John Gill, Jon Borman, Yu-Hui Rogers, Marvin E Frazier, Stephen W Scherer, Robert L Strausberg, and J. Craig Venter. The diploid genome sequence of an individual human. *PLoS Biol.*, 5:e254, Sep 2007.

[33] 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.

[34] 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.

[35] Brock A. Peters, Bahram G. Kermani, Andrew B. Sparks, Oleg Alferov, Peter Hong, Andrei Alexeev, Yuan Jiang, Fredrik Dahl, Y. Tom Tang, Juergen Haas, Kimberly Robasky, Alexander Wait Zaranek, Je-Hyuk Lee, Madeleine Price Ball, Joseph E. Peterson, Helena Perazich, George Yeung, Jia Liu, Linsu Chen, Michael I. Kennemer, Kaliprasad Pothuraju, Karel Konvicka, Mike Tsoupko-Sitnikov, Krishna P. Pant, Jessica C. Ebert, Geoffrey B. Nilsen, Jonathan Baccash, Aaron L. Halpern, George M. Church, and Radoje Drmanac. Accurate whole-genome sequencing and haplotyping from 10 to 20 human cells. *Nature*, 487(7406):190–195, Jul 2012.

[36] 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.

[37] S. Amini, D. Pushkarev, L. Christiansen, E. Kostem, T. Royce, C. Turk, N. Pignatelli, A. Adey, J. O. Kitzman, K. Vijayan, M. Ronaghi, J. Shendure, K. L. Gunderson, and F. J. Steemers. Haplotype-resolved whole-genome sequencing by contiguity-preserving transposition and combinatorial indexing. *Nat. Genet.*, 46(12):1343–1349, Dec 2014.

[38] M. Pendleton, R. Sebra, A. W. Pang, A. Ummat, O. Franzen, T. Rausch, A. M. Stutz, W. Stedman, T. Anantharaman, A. Hastie, Alex Hastie, Heng Dai, Markus Hsi-Yang Fritz, Han Cao, Ariella Cohain, Gintaras Deikus, Russell E Durrett, Scott C Blanchard, Roger Altman, Chen-Shan Chin, Yan Guo, Ellen E Paxinos, Jan O Korbel, Robert B Darnell, W Richard McCombie, Pui-Yan Kwok, Christopher E Mason, Eric E Schadt, and Ali Bashir. Assembly and diploid architecture of an individual human genome via single-molecule technologies. *Nat. Methods*, 12(8):780–786, Aug 2015.

[39] V. Bansal and V. Bafna. HapCUT: an efficient and accurate algorithm for the haplotype assembly problem. *Bioinformatics*, 24(16):i153–159, Aug 2008.

[40] D. He, A. Choi, K. Pipatsrisawat, A. Darwiche, and E. Eskin. Optimal algorithms for haplotype assembly from whole-genome sequence data. *Bioinformatics*, 26(12):i183–190, Jun 2010.

[41] Jorge Duitama, Thomas Huebsch, Gayle McEwen, Eun-Kyung Suk, and Margret R. Hoehe. Refhap: A reliable and fast algorithm for single individual haplotyping. In *Proceedings of the First ACM International Conference on Bioinformatics and Computational Biology*, BCB '10, pages 160–169, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0438-2. doi: 10.1145/1854776.1854802.

[42] D. Aguiar and S. Istrail. HapCompass: a fast cycle basis algorithm for accurate haplotype assembly of sequence data. *J. Comput. Biol.*, 19(6):577–590, Jun 2012.

[43] V. Kuleshov. Probabilistic single-individual haplotyping. *Bioinformatics*, 30(17):i379–385, Sep 2014.

[44] 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, Sep 2013.

[45] Alessandro Panconesi and Mauro Sozio. Fast hare: A fast heuristic for single individual snp haplotype reconstruction. In *International Workshop on Algorithms in Bioinformatics*, pages 266–277. Springer, 2004.

[46] H. Matsumoto and H. Kiryu. MixSIH: a mixture model for single individual haplotyping. *BMC Genomics*, 14 Suppl 2:S5, 2013.

[47] E. Berger, D. Yorukoglu, J. Peng, and B. Berger. HapTree: a novel Bayesian framework for single individual polyplotyping using NGS data. *PLoS Comput. Biol.*, 10(3):e1003502, Mar 2014.

[48] Justin M. Zook, David Catoe, Jennifer McDaniel, Lindsay Vang, Noah Spies, Arend Sidow, Ziming Weng, Yuling Liu, Christopher E. Mason, Noah Alexander, Elizabeth Henaff, Alexa B.R. McIntyre, Dhruva Chandramohan, Feng Chen, Erich Jaeger, Ali Moshrefi, Khoa Pham, William Stedman, Tiffany Liang, Michael Saghbini, Zeljko Dzakula, Alex Hastie, Han Cao, Gintaras Deikus, Eric Schadt, Robert Sebra, Ali Bashir, Rebecca M. Truty, Christopher C. Chang, Natali Gulbahce, Keyan Zhao, Srinka Ghosh, Fiona Hyland, Yutao Fu, Mark Chaisson, Chunlin Xiao, Jonathan Trow, Stephen T. Sherry, Alexander W. Zaranek, Madeleine Ball, Jason Bobe, Preston Estep, George M. Church, Patrick Marks, Sofia Kyriazopoulou-Panagiotopoulou, Grace X.Y. Zheng, Michael Schnall-Levin, Heather S. Ordonez, Patrice A. Mudivarti, Kristina Giorda, Ying Sheng, Karoline Bjarnesdatter Rypdal, and Marc Salit. Extensive sequencing of seven human genomes to characterize benchmark reference materials. *Sci Data*, 3:160025, 2016.

[49] S. S. Rao, M. H. Huntley, N. C. Durand, E. K. Stamenova, I. D. Bochkov, J. T. Robinson, A. L. Sanborn, I. Machol, A. D. Omer, E. S. Lander, and E. L. Aiden. A 3D map of the human genome at kilobase resolution reveals principles of chromatin looping. *Cell*, 159 (7):1665–1680, Dec 2014.

[50] S. Goodwin, J. Gurtowski, S. Ethe-Sayers, P. Deshpande, M. C. Schatz, and W. R. McCombie. Oxford Nanopore sequencing, hybrid error correction, and de novo assembly of a eukaryotic genome. *Genome Res.*, 25(11):1750–1756, Nov 2015.

[51] N. H. Putnam, B. L. O'Connell, J. C. Stites, B. J. Rice, M. Blanchette, R. Calef, C. J. Troll, A. Fields, P. D. Hartley, C. W. Sugnet, D. Haussler, D. S. Rokhsar, and R. E. Green. Chromosome-scale shotgun assembly using an in vitro method for long-range linkage. *Genome Res.*, 26(3):342–350, Mar 2016.

[52] V. Kuleshov, D. Xie, R. Chen, D. Pushkarev, Z. Ma, T. Blauwkamp, M. Kertesz, and M. Snyder. Whole-genome haplotyping using long reads and statistical methods. *Nat. Biotechnol.*, 32(3):261–266, Mar 2014.

[53] O. Delaneau, J. F. Zagury, and J. Marchini. Improved whole-chromosome phasing for disease and population genetic studies. *Nat. Methods*, 10(1):5–6, Jan 2013.

[54] David Sims, Ian Sudbery, Nicholas E Ilott, Andreas Heger, and Chris P Ponting. Sequencing depth and coverage: key considerations in genomic analyses. *Nature Reviews Genetics*, 15(2):121–132, 2014.

[55] A. Patel, P. Edge, S. Selvaraj, V. Bansal, and V. Bafna. InPhaDel: integrative shotgun and proximity-ligation sequencing to phase deletions with single nucleotide polymorphisms. *Nucleic Acids Res.*, 44(12):e111, Jul 2016.

[56] Ross Lippert, Russell Schwartz, Giuseppe Lancia, and Sorin Istrail. Algorithmic strategies for the single nucleotide polymorphism haplotype assembly problem. *Briefings in Bioinformatics*, 3(1):23–31, 2002. doi: 10.1093/bib/3.1.23. URL `http://dx.doi.org/10.1093/bib/3.1.23`.

[57] L. M. Li, J. H. Kim, and M. S. Waterman. Haplotype reconstruction from SNP alignment. *J. Comput. Biol.*, 11(2-3):505–516, 2004.

[58] 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.

[59] C. Lo, A. Bashir, V. Bansal, and V. Bafna. Strobe sequence design for haplotype assembly. *BMC Bioinformatics*, 12 Suppl 1:S24, 2011.

[60] Hao Zhao, Zhifu Sun, Jing Wang, Haojie Huang, Jean-Pierre Kocher, and Liguo Wang. CrossMap: a versatile tool for coordinate conversion between genome assemblies. *Bioinformatics (Oxford, England)*, 30(7):1006–1007, April 2014. ISSN 1367-4811. doi: 10.1093/bioinformatics/btt730.

[61] Heng Li. Aligning sequence reads, clone sequences and assembly contigs with bwa-mem. page preprint at https://arxiv.org/abs/1303.3997, 2013.

[62] Yong Wang and Nicholas E Navin. Advances and applications of single-cell sequencing technologies. *Molecular cell*, 58(4):598–609, 2015.

[63] 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 2009.

[64] 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, 2010.

[65] Erik Garrison and Gabor Marth. Haplotype-based variant detection from short-read sequencing. page preprint at https://arxiv.org/abs/1207.3907, 2012.

[66] Hamim Zafar, Yong Wang, Luay Nakhleh, Nicholas Navin, and Ken Chen. Monovar: single-nucleotide variant detection in single cells. *Nature methods*, 13(6):505, 2016.

[67] Xiao Dong, Lei Zhang, Brandon Milholland, Moonsook Lee, Alexander Y Maslov, Tao Wang, and Jan Vijg. Accurate identification of single-nucleotide variants in whole-genome-amplified single cells. *Nature methods*, 14(5):491, 2017.

[68] Peter Edge, Vineet Bafna, and Vikas Bansal. Hapcut2: robust and accurate haplotype assembly for diverse sequencing technologies. *Genome Res.*, 27(5):801–812, 2017.

[69] Vikas Bansal. An accurate algorithm for the detection of dna fragments from dilution pool sequencing experiments. *Bioinformatics*, 34(1):155–162, 2017.

[70] Cheng-Zhong Zhang, Viktor A Adalsteinsson, Joshua Francis, Hauke Cornils, Joonil Jung, Cecile Maire, Keith L Ligon, Matthew Meyerson, and J Christopher Love. Calibrating genomic and allelic coverage bias in single-cell sequencing. *Nature communications*, 6: 6822, 2015.

[71] J Guillermo Paez, Ming Lin, Rameen Beroukhim, Jeffrey C Lee, Xiaojun Zhao, Daniel J Richter, Stacey Gabriel, Paula Herman, Hidefumi Sasaki, David Altshuler, Cheng Li, Matthew Meyerson, and William R. Sellers. Genome coverage and sequence fidelity of ϕ29 polymerase-based multiple strand displacement whole genome amplification. *Nucleic Acids Research*, 32(9):e71–e71, 2004.

[72] ENCODE Project Consortium. An integrated encyclopedia of dna elements in the human genome. *Nature*, 489(7414):57, 2012.

[73] Petr Danecek, Adam Auton, Goncalo Abecasis, Cornelis A Albers, Eric Banks, Mark A DePristo, Robert E Handsaker, Gerton Lunter, Gabor T Marth, Stephen T Sherry, Gilean McVean, Richard Durbin, and 1000 Genomes Project Analysis Group. The variant call format and vcftools. *Bioinformatics*, 27(15):2156–2158, 2011.

[74] Madeleine P. Ball, Joseph V. Thakuria, Alexander Wait Zaranek, Tom Clegg, Abraham M. Rosenbaum, Xiaodi Wu, Misha Angrist, Jong Bhak, Jason Bobe, Matthew J. Callow, Carlos Cano, Michael F. Chou, Wendy K. Chung, Shawn M. Douglas, Preston W. Estep, Athurva Gore, Peter Hulick, Alberto Labarga, Je-Hyuk Lee, Jeantine E. Lunshof, Byung Chul Kim, Jong-Il Kim, Zhe Li, Michael F. Murray, Geoffrey B. Nilsen, Brock A. Peters, Anugraha M. Raman, Hugh Y. Rienhoff, Kimberly Robasky, Matthew T. Wheeler, Ward Vandewege, Daniel B. Vorhaus, Joyce L. Yang, Luhan Yang, John Aach, Euan A. Ashley, Radoje Drmanac, Seong-Jin Kim, Jin Billy Li, Leonid Peshkin, Christine E. Seidman, Jeong-Sun Seo, Kun Zhang, Heidi L. Rehm, , and George M. Church. A public resource facilitating clinical use of genomes. *Proceedings of the National Academy of Sciences*, 109(30): 11920–11927, 2012.

[75] Johannes Köster and Sven Rahmann. Snakemake—a scalable bioinformatics workflow engine. *Bioinformatics*, 28(19):2520–2522, 2012.

[76] S. Goodwin, J. D. McPherson, and W. R. McCombie. Coming of age: ten years of next-generation sequencing technologies. *Nat. Rev. Genet.*, 17(6):333–351, 2016.

[77] David Sims, Ian Sudbery, Nicholas E Ilott, Andreas Heger, and Chris P Ponting. Sequencing depth and coverage: key considerations in genomic analyses. *Nat. Rev. Genet.*, 15(2): 121, 2014.

[78] R. Nielsen, J. S. Paul, A. Albrechtsen, and Y. S. Song. Genotype and SNP calling from next-generation sequencing data. *Nat. Rev. Genet.*, 12(6):443–451, Jun 2011.

[79] K. Bryc, N. Patterson, and D. Reich. A novel approach to estimating heterozygosity from low-coverage genome sequence. *Genetics*, 195(2):553–561, Oct 2013.

[80] D. Mandelker, R. J. Schmidt, A. Ankala, K. McDonald Gibson, M. Bowser, H. Sharma, E. Duffy, M. Hegde, A. Santani, M. Lebo, and B. Funke. Navigating highly homologous genes in a molecular diagnostic setting: a resource for clinical next-generation sequencing. *Genet. Med.*, 18(12):1282–1289, 2016.

[81] S. Ardui, A. Ameur, J. R. Vermeesch, and M. S. Hestand. Single molecule real-time (SMRT) sequencing comes of age: applications and utilities for medical diagnostics. *Nucleic Acids Res.*, 46(5):2159–2168, Mar 2018.

[82] Mark J.P. Chaisson, John Huddleston, Megan Y. Dennis, Peter H. Sudmant, Maika Malig, Fereydoun Hormozdiari, Francesca Antonacci, Urvashi Surti, Richard Sandstrom, Matthew Boitano, Jane M. Landolin, John A. Stamatoyannopoulos, Michael W. Hunkapiller, Jonas Korlach, , and Evan E. Eichler. Resolving the complexity of the human genome using single-molecule sequencing. *Nature*, 517(7536):608–611, 2015.

[83] M. J. Chaisson, S. Mukherjee, S. Kannan, and E. E. Eichler. Resolving multicopy duplications de novo using polyploid phasing. *Res Comput Mol Biol*, 10229:117–133, May 2017.

[84] J. M. Zook, B. Chapman, J. Wang, D. Mittelman, O. Hofmann, W. Hide, and M. Salit. Integrating human sequence data sets provides a resource of benchmark SNP and indel genotype calls. *Nat. Biotechnol.*, 32(3):246–251, Mar 2014.

[85] Justin Zook, Jennifer McDaniel, Hemang Parikh, Haynes Heaton, Sean A Irvine, Len Trigg, Rebecca Truty, Cory Y McLean, Francisco M De La Vega, and Marc Salit. Reproducible integration of multiple sequencing datasets to form high-confidence snp, indel, and reference calls for five human genome reference materials. page preprint at https://doi.org/10.1101/281006, 2018.

[86] Heng Li, Jonathan M Bloom, Yossi Farjoun, Mark Fleharty, Laura Gauthier, Benjamin Neale, and Daniel MacArthur. A synthetic-diploid benchmark for accurate variant-calling evaluation. *Nat. Methods*, 15(8):595–597, 2018.

[87] D. M. Borras, R. H. A. M. Vossen, M. Liem, H. P. J. Buermans, H. Dauwerse, D. van Heusden, R. T. Gansevoort, J. T. den Dunnen, B. Janssen, D. J. M. Peters, M. Losekoot, and S. Y. Anvar. Detecting PKD1 variants in polycystic kidney disease patients by single-molecule long-read sequencing. *Hum. Mutat.*, 38(7):870–879, 07 2017.

[88] J. Huddleston, M. J. P. Chaisson, K. M. Steinberg, W. Warren, K. Hoekzema, D. Gordon, T. A. Graves-Lindsay, K. M. Munson, Z. N. Kronenberg, L. Vives, P. Peluso, M. Boitano, C. S. Chin, J. Korlach, R. K. Wilson, and E. E. Eichler. Discovery and genotyping of structural variation from long-read haploid genome sequence data. *Genome Res.*, 27(5): 677–685, 2017.

[89] Peter H. Sudmant, Tobias Rausch, Eugene J. Gardner, Robert E. Handsaker, Alexej Abyzov, John Huddleston, Yan Zhang, Kai Ye, Goo Jun, Markus Hsi-Yang Fritz, Miriam K. Konkel, Ankit Malhotra, Adrian M. Sttz, Xinghua Shi, Francesco Paolo Casale, Jieming Chen, Fereydoun Hormozdiari, Gargi Dayama, Ken Chen, Maika Malig, Mark J. P. Chaisson, Klaudia Walter, Sascha Meiers, Seva Kashin, Erik Garrison, Adam Auton, Hugo Y. K. Lam, Xinmeng Jasmine Mu, Can Alkan, Danny Antaki, Taejeong Bae, Eliza Cerveira, Peter Chines, Zechen Chong, Laura Clarke, Elif Dal, Li Ding, Sarah Emery, Xian Fan, Madhusudan Gujral, Fatma Kahveci, Jeffrey M. Kidd, Yu Kong, Eric-Wubbo Lameijer, Shane McCarthy, Paul Flicek, Richard A. Gibbs, Gabor Marth, Christopher E. Mason, Androniki Menelaou, Donna M. Muzny, Bradley J. Nelson, Amina Noor, Nicholas F. Parrish, Matthew Pendleton, Andrew Quitadamo, Benjamin Raeder, Eric E. Schadt, Mallory Romanovitch, Andreas Schlattl, Robert Sebra, Andrey A. Shabalin, Andreas Untergasser, Jerilyn A. Walker, Min Wang, Fuli Yu, Chengsheng Zhang, Jing Zhang, Xiangqun Zheng-Bradley, Wanding Zhou, Thomas Zichner, Jonathan Sebat, Mark A. Batzer, Steven A. McCarroll, The 1000 Genomes Project Consortium, Ryan E. Mills, Mark B. Gerstein, Ali Bashir, Oliver Stegle, Scott E. Devine, Charles Lee, Evan E. Eichler, and Jan O. Korbel.

An integrated map of structural variation in 2,504 human genomes. *Nature*, 526(7571): 75–81, 2015.

[90] M. J. P. Chaisson, A. D. Sanders, X. Zhao, A. Malhotra, D. Porubsky, T. Rausch, E. J. Gardner, O. L. Rodriguez, L. Guo, R. L. Collins, X. Fan, J. Wen, R. E. Handsaker, S. Fairley, Z. N. Kronenberg, X. Kong, F. Hormozdiari, D. Lee, A. M. Wenger, A. R. Hastie, D. Antaki, T. Anantharaman, P. A. Audano, H. Brand, S. Cantsilieris, H. Cao, E. Cerveira, C. Chen, X. Chen, C. S. Chin, Z. Chong, N. T. Chuang, C. C. Lambert, D. M. Church, L. Clarke, A. Farrell, J. Flores, T. Galeev, D. U. Gorkin, M. Gujral, V. Guryev, W. H. Heaton, J. Korlach, S. Kumar, J. Y. Kwon, E. T. Lam, J. E. Lee, J. Lee, W. P. Lee, S. P. Lee, S. Li, P. Marks, K. Viaud-Martinez, S. Meiers, K. M. Munson, F. C. P. Navarro, B. J. Nelson, C. Nodzak, A. Noor, S. Kyriazopoulou-Panagiotopoulou, A. W. C. Pang, Y. Qiu, G. Rosanio, M. Ryan, A. Stutz, D. C. J. Spierings, A. Ward, A. E. Welch, M. Xiao, W. Xu, C. Zhang, Q. Zhu, X. Zheng-Bradley, E. Lowy, S. Yakneen, S. McCarroll, G. Jun, L. Ding, C. L. Koh, B. Ren, P. Flicek, K. Chen, M. B. Gerstein, P. Y. Kwok, P. M. Lansdorp, G. T. Marth, J. Sebat, X. Shi, A. Bashir, K. Ye, S. E. Devine, M. E. Talkowski, R. E. Mills, T. Marschall, J. O. Korbel, E. E. Eichler, and C. Lee. Multi-platform discovery of haplotype-resolved structural variation in human genomes. *Nat Commun*, 10(1):1784, 2019.

[91] Fei Guo, Dan Wang, and Lusheng Wang. Progressive approach for snp calling and haplotype assembly using single molecular sequencing data. *Bioinformatics*, 34(12): 2012–2018, 2018.

[92] R. Poplin, P. C. Chang, D. Alexander, S. Schwartz, T. Colthurst, A. Ku, D. Newburger, J. Dijamco, N. Nguyen, P. T. Afshar, S. S. Gross, L. Dorfman, C. Y. McLean, and M. A. DePristo. A universal SNP and small-indel variant caller using deep neural networks. *Nat. Biotechnol.*, 36(10):983–987, Nov 2018.

[93] J. Ebler, M. Haukness, T. Pesout, T. Marschall, and B. Paten. Haplotype-aware diplotyping from noisy long reads. *Genome Biol.*, 20(1):116, 06 2019.

[94] M. Jain, I. T. Fiddes, K. H. Miga, H. E. Olsen, B. Paten, and M. Akeson. Improved data analysis for the MinION nanopore sequencer. *Nat. Methods*, 12(4):351–356, Apr 2015.

[95] Richard Durbin, Sean R Eddy, Anders Krogh, and Graeme Mitchison. *Biological sequence analysis: probabilistic models of proteins and nucleic acids*. Cambridge University Press, 1998.

[96] Mark J Chaisson and Glenn Tesler. Mapping single molecule sequencing reads using basic local alignment with successive refinement (blasr): application and theory. *BMC bioinformatics*, 13(1):238, 2012.

[97] F. J. Sedlazeck, P. Rescheneder, M. Smolka, H. Fang, M. Nattestad, A. von Haeseler, and M. C. Schatz. Accurate detection of complex structural variations using single-molecule sequencing. *Nat. Methods*, 15(6):461–468, 2018.

[98] H. Li. Minimap2: pairwise alignment for nucleotide sequences. *Bioinformatics*, 34(18): 3094–3100, 09 2018.

[99] Ruibang Luo, Fritz J Sedlazeck, Tak-Wah Lam, and Michael C Schatz. A multi-task convolutional deep neural network for variant calling in single molecule sequencing. *Nat Commun*, 10(1):998, 2019.

[100] E Karlsson, A Lärkeryd, Andreas Sjödin, M Forsman, and Per Stenberg. Scaffolding of a bacterial genome using minion nanopore sequencing. *Scientific reports*, 5:11996, 2015.

[101] N. J. Loman, J. Quick, and J. T. Simpson. A complete bacterial genome assembled de novo using only nanopore sequencing data. *Nat. Methods*, 12(8):733–735, Aug 2015.

[102] Mark A DePristo, Eric Banks, Ryan Poplin, Kiran V Garimella, Jared R Maguire, Christopher Hartl, Anthony A Philippakis, Guillermo Del Angel, Manuel A Rivas, Matt Hanna, Aaron McKenna, Tim J Fennell, Andrew M Kernytsky, Andrey Y Sivachenko, Kristian Cibulskis, Stacey B Gabriel, David Altshuler, and Mark J Daly. A framework for variation discovery and genotyping using next-generation dna sequencing data. *Nat. Genet.*, 43(5): 491–498, 2011.

[103] Michael A Eberle, Epameinondas Fritzilas, Peter Krusche, Morten Källberg, Benjamin L Moore, Mitchell A Bekritsky, Zamin Iqbal, Han-Yu Chuang, Sean J Humphray, Aaron L Halpern, Kruglyak S, Margulies EH, McVean G, and Bentley DR. A reference data set of 5.4 million phased human variants validated by genetic inheritance from sequencing a three-generation 17-member pedigree. *Genome Res.*, 27(1):157–164, 2017.

[104] G. Rakocevic, V. Semenyuk, W. P. Lee, J. Spencer, J. Browning, I. J. Johnson, V. Arsenijevic, J. Nadj, K. Ghose, M. C. Suciu, S. G. Ji, G. Demir, L. Li, B. C. Topta, A. Dolgoborodov, B. Pollex, I. Spulber, I. Glotova, P. Komar, A. L. Stachyra, Y. Li, M. Popovic, M. Kallberg, A. Jain, and D. Kural. Fast and accurate genomic analyses using genome graphs. *Nat. Genet.*, 51(2):354–362, 02 2019.

[105] A. M. Wenger, P. Peluso, W. J. Rowell, P. C. Chang, R. J. Hall, G. T. Concepcion, J. Ebler, A. Fungtammasan, A. Kolesnikov, N. D. Olson, A. Topfer, M. Alonge, M. Mahmoud, Y. Qian, C. S. Chin, A. M. Phillippy, M. C. Schatz, G. Myers, M. A. DePristo, J. Ruan, T. Marschall, F. J. Sedlazeck, J. M. Zook, H. Li, S. Koren, A. Carroll, D. R. Rank, and M. W. Hunkapiller. Accurate circular consensus long-read sequencing improves variant detection and assembly of a human genome. *Nat. Biotechnol.*, Aug 2019.

[106] J. D. Merker, A. M. Wenger, T. Sneddon, M. Grove, Z. Zappala, L. Fresard, D. Waggott, S. Utiramerur, Y. Hou, K. S. Smith, S. B. Montgomery, M. Wheeler, J. G. Buchan, C. C. Lambert, K. S. Eng, L. Hickey, J. Korlach, J. Ford, and E. A. Ashley. Long-read genome sequencing identifies causal structural variation in a Mendelian disease. *Genet. Med.*, 20 (1):159–163, Jan 2018.

[107] T. Mizuguchi, T. Suzuki, C. Abe, A. Umemura, K. Tokunaga, Y. Kawai, M. Nakamura, M. Nagasaki, K. Kinoshita, Y. Okamura, S. Miyatake, N. Miyake, and N. Matsumoto. A 12-kb structural variation in progressive myoclonic epilepsy was newly identified by long-read whole-genome sequencing. *J. Hum. Genet.*, 64(5):359–368, May 2019.

[108] H. Ishiura, K. Doi, J. Mitsui, J. Yoshimura, M. K. Matsukawa, A. Fujiyama, Y. Toyoshima, A. Kakita, H. Takahashi, Y. Suzuki, S. Sugano, W. Qu, K. Ichikawa, H. Yurino, K. Higasa, S. Shibata, A. Mitsue, M. Tanaka, Y. Ichikawa, Y. Takahashi, H. Date, T. Matsukawa, J. Kanda, F. K. Nakamoto, M. Higashihara, K. Abe, R. Koike, M. Sasagawa, Y. Kuroha, N. Hasegawa, N. Kanesawa, T. Kondo, T. Hitomi, M. Tada, H. Takano, Y. Saito, K. Sanpei, O. Onodera, M. Nishizawa, M. Nakamura, T. Yasuda, Y. Sakiyama, M. Otsuka, A. Ueki, K. I. Kaida, J. Shimizu, R. Hanajima, T. Hayashi, Y. Terao, S. Inomata-Terada, M. Hamada, Y. Shirota, A. Kubota, Y. Ugawa, K. Koh, Y. Takiyama, N. Ohsawa-Yoshida, S. Ishiura, R. Yamasaki, A. Tamaoka, H. Akiyama, T. Otsuki, A. Sano, A. Ikeda, J. Goto, S. Morishita, and S. Tsuji. Expansions of intronic TTTCA and TTTTA repeats in benign adult familial myoclonic epilepsy. *Nat. Genet.*, 50(4):581–590, 2018.

[109] H. Li. Toward better understanding of artifacts in variant calling from high-coverage samples. *Bioinformatics*, 30(20):2843–2851, Oct 2014.

[110] N Homer. Dwgsim:whole genome simulator for next-generation sequencing. `https://github.com/nh13/DWGSIM`, 2010.

[111] Bianca K Stöcker, Johannes Köster, and Sven Rahmann. Simlord: simulation of long read data. *Bioinformatics*, 32(17):2704–2706, 2016.

[112] R. M. Kuhn, D. Haussler, and W. J. Kent. The UCSC genome browser and associated tools. *Brief. Bioinformatics*, 14(2):144–161, Mar 2013.

[113] J. G. Cleary, R. Braithwaite, K. Gaastra, B. S. Hilbush, S. Inglis, S. A. Irvine, A. Jackson, R. Littin, S. Nohzadeh-Malakshah, M. Rathod, D. Ware, L. Trigg, and F. M. De La Vega. Joint variant and de novo mutation identification on pedigrees from high-throughput sequencing data. *J. Comput. Biol.*, 21(6):405–419, 2014.

[114] Aaron R Quinlan and Ira M Hall. Bedtools: a flexible suite of utilities for comparing genomic features. *Bioinformatics*, 26(6):841–842, 2010.

[115] Johannes Köster. Rust-bio: a fast and safe bioinformatics library. *Bioinformatics*, 32(3): 444–446, 2015.

[116] B Grüning, R Dale, A Sjödin, BA Chapman, J Rowe, CH Tomkins-Tinch, R Valieris, J Köster, and Team Bioconda. Bioconda: sustainable and comprehensive software distribution for the life sciences. *Nat. Methods*, 15(7):475–476, 2018.

[117] Derek W Barnett, Erik K Garrison, Aaron R Quinlan, Michael P Strömberg, and Gabor T Marth. Bamtools: a c++ api and toolkit for analyzing and managing bam files. *Bioinformatics*, 27(12):1691–1692, 2011.

[118] Gonzalo Navarro and Mathieu Raffinot. A bit-parallel approach to suffix automata: Fast extended string matching. In *Annual Symposium on Combinatorial Pattern Matching*, pages 14–33. Springer, 1998.