UNIVERSITY OF CALIFORNIA
SANTA CRUZ

IMPROVING SEQUENCE ALIGNMENT AND VARIANT CALLING THROUGH THE PROCESS OF POPULATION AND PEDIGREE-BASED GRAPH ALIGNMENT

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

DOCTOR OF PHILOSOPHY

in

BIOMOLECULAR ENGINEERING & BIOINFORMATICS

by

Charles J. Markello

March 2022

The Dissertation of Charles J. Markello
is approved:

Professor Benedict Paten, Chair

Professor David Haussler

Professor Beth Shapiro

Professor R. Edward Green

Peter F. Biehl
Vice Provost and Dean of Graduate Studies

Table of Contents

List of Figures

List of Tables

Abstract

Improving sequence alignment and variant calling through the process of population and pedigree-based graph alignment

by

Charles J. Markello

In current sequencing methodology, a linear genome reference is used to detect genetic-variants based on collections of sequence reads. The linear reference introduces potential misalignment of reads that don't exactly match the reference or the copy number of sequences in the reference doesn't match the sample correctly. This is known as *reference bias*. In the field of clinical genetics for rare diseases, a resulting reduction in genotyping accuracy in some regions has likely prevented the resolution of some cases. Pangenome graphs embed population variation into a reference structure to reduce reference bias. While this helps to reduce reference bias, further performance improvements are possible with the aid of pedigree information. In this dissertation I present my research on the methods developed to build programs that apply pangenome graphs to solve these problems. First, I share the work I've contributed towards streamlining a single-sample pangenome software workflow and the accuracy enhancements I've contributed within the pangenome effort. Next, I share my methods in incorporating pedigree information within the pangenome framework and show how performance is improved over standard pangenomes. I describe an extension of this work to demonstrate the clinical application of this workflow. Finally, I cover various projects I've contributed to that catalogue and use detected variants for deleterious classification.

To my parents,

Tom and Barbara Markello,

who have always been there to support me every step of the way,

and to my sister,

Jay Markello,

for giving perspective and balance to my life.

Acknowledgments

It should be said that there are too many people for me to thank for getting me through the graduate phase of my life. To start, I want to thank the members of my committee, Benedict Paten, David Haussler, Beth Shapiro and Ed Green for the support, patience, critiques and opportunities that they have given me over the years for the research being presented in this dissertation. I would also very much like to thank the members of the National Institutes of Health Undiagnosed Diseases Program, in particular William A. Gahl, Tim Gall, Charles Huang, Alex Rodriguez, Alex Brandt, Elise Flynn, Jeremy Elson, Andy Hsieh, Cynthia Tifft, and David Adams, whose support has made all of this work possible. Much of what I have learned about how to conduct science, how to write software in robust and usable ways and how to communicate and share those results are the product of their guidance and feedback.

In addition, I would like to thank my graduate, post-graduate and staff colleagues Adam Novak, Audrey Musselman-Brown, Brandon Saint-John, Colleen Bosworth, David Steinberg, Glenn Hickey, Hannes Schmidt, James Casaletto, Jean Monlong, Joel Armstrong, John Vivian, Jonas Sibbesen, Jordan Eizenga, Jouni Sirén, Kishwar Shafin, Marina Haukness, Mike Lin, Robin Rounthwaite, Ryan Lorig-Roach, Sean Blum, Thomas Ng, Trevor Pesout, Xian Chang, and Yohei Rosen. They have been on the (metaphorical) ground-floor with me who have advised me in my academic career and taught me much about the research techniques and obscure technologies that I wouldn't have known without them. They have been there for me for the longest time and have shown me how to live a more balanced life in tandem with the time spent doing graduate work. I also want to thank my friends who include not just my lab

colleagues, but also the friends I've made in college, their friends, my old housemates, and my sisters friends. Specifically, I want to thank Aryan Sarparast, Michael Nguyen, Nick Giampietro, Chris Sundahl, Wyatt Fluckiger, Eric Allen Carrillo, Dan Carrillo, Adam Carrillo, Corey Pigott, and Karen Alarcon. They have done the most heavy-lifting in terms of maintaining my sanity and have significantly contributed to expanding my experiences with travel, food, climbing, and strange alcoholic beverages.

I also want to thank my family who have all helped me through each stage of this journey. My mom and dad for giving me the emotional and intellectual support that I needed to complete this work. To my sister and sister-in-law, Jay and Kelly, for being there for me while adjusting to life in California and teaching me their mystic ways in the culinary arts and showing me that rats actually make for really great pets.

To summarize, the work done in this thesis has been made possible because of the guidance and assistance provided by everyone mentioned here. Graduate school is hard, but it would have been impossible without their help.

Part I

Introduction and Background

Chapter 1

Introduction

The current fields of biology and clinical medicine have gravitated towards investigating the role that genes play in the landscape of evolution and disease. Biologists are increasingly asking more granular questions about the nature of these biological mechanisms. These questions cover a wide array of disciplines that include the analysis of DNA, RNA, gene regulation, protein folding and function. As a consequence, large amounts of data are required to answer questions that relate these domains of knowledge into a more comprehensive picture of biology. Large data requires computational solutions in order to understand and synthesise that information in a reasonable amount of time. The field of Bioinformatics attempts to bridge these gaps by applying computational methods to effectively answer questions that require processing big-data. Since the initial draft of the Human Genome was completed by the Human Genome Project in 2003, translational and personalized medicine, the fields of medicine that focuses on the application of genetic and biological information towards solving the individual aspect of disease, have taken on a larger role in the clinical community. One of the largest areas

of attention in personalized and translational medicine is on understanding how genetic mechanisms connect with the phenotypic expression of organisms, also commonly referred to as the genotype-to-phenotype mapping problem.

One component of the genotype-to-phenotype mapping problem is the role of rare diseases as resulting from combinations of alleles at the point of conception and somatic mutations at the early stages of embryological development. The National Institutes of Health Undiagnosed Diseases Program (NIHUDP) is challenged to discover and understand the role of rare genetic variation and mechanisms and how they relate to the expression of rare diseases in their patient cohort. The driving motivation of this thesis is to develop methodologies that leverage the latest techniques in Bioinformatic algorithms and programs along with pedigree and clinical database information in order to provide the tools that can aid in the discovery of genetic variants that contribute to rare disease. The primary technology covered in this thesis concerns the the role of mathematical-graph based reference sequences, referred to as *pangenome references*, and how they are used to enhance detection of genetic-variants.

The first part of this thesis covers the background and tools used in parts 2, 3 and 4 of the dissertation. Chapter 2 focuses on the background of various sequencing technologies, the use of various genomes for sequence alignment, genetic-variant callers, and software workflow frameworks.

Chapter 3 in this thesis will focus on my work on the development of a streamlined and interoperable program that implements a genetic-variant-detecting workflow based on *pangenome references* for the application of single-samples. The chapter covers how the program was applied to real and simulated data and how it demonstrates improvements in genetic-

variant-detection accuracy over other commonly used tools. I will also cover the contributions I've made towards construction of a *pangenome reference*.

Chapter 4 covers the development of a program that extends the methods developed in chapter 3 to leverage nuclear family pedigrees. I demonstrate how this program further enhances the genetic-variant-detecting abilities of the program developed in chapter 3 by evaluating performance over simulated and real data. I will also cover the development and evaluation of an enhancement to a variant discovery program developed in the NIHUDP that automatically detects a short-list of genetic-variants that are potentially deleterious to the individual.

Chapters 5 and 6 of this thesis covers projects that I have been involved with and helped to develop that cover the interpretation of variants that can be discovered by the programs developed in chapters 3 and 4.

Chapter 2

Background

2.1 Sequencing Technology

The availability of next-generation sequencing (NGS) technologies has given researchers the ability to study human genetic variation at the population level [9, 6, 67, 63]. NGS methods have become widely adopted in the fields of both medical and population genetics. Much of NGS is used in genetic screens for genes and variants of known disease association. It is also being used to find rare variants to diagnose new diseases or diseases of unknown cause. For example, the National Institutes of Health Undiagnosed Diseases Program (UDP), whose purpose is to diagnose individuals who have already had exhaustive workup yet remain undiagnosed, uses NGS along with gene filtering approaches to find causal mutations for undiagnosed disorders [36]. In another group, Mallott et al. employed similar methods using NGS to find variants that are potential causes for Severe Combined Immunodeficiency (SCID) in children [68].

Both of these groups used NGS to filter for the minimum number of potential candidates of causal variants to unknown diseases in order to drive down the time and costs of the study. Specifically, the groups used the exhaustive information contained in NGS data to examine a patient's entire set of variants for features that fulfill their search criteria. Some of these criteria include searching for variants that exist in other patients with similar diseases, variants that follow inheritance patterns with family members that also share the same signs of the disorder, variants that are not contained in those who are healthy, and variants that are located in genetic regions that contains other variants that share known disease signs with those of the patient [36, 68]. The lower the number of candidate variants there are, the fewer expensive in-vitro cellular biology experiments that are performed and model biological systems that are created to verify the genetic cause for the disorder, thus reducing the cost of the study [36, 68, 89].

The practice of variant filtration to reduce the cost of a study incentivizes the use of all available genetic information in sequencing technologies[43, 94]. The current state of NGS relies on methods and algorithms which perform well in identifying genetic variation for a majority of the human genome, but still have difficulty in reliably sequencing highly variant or repetitive regions of the genome. A primary source for this difficulty is that the reference genome used in NGS insufficiently represents variation in these complex regions [18]. In NGS, most of the variation that is identified within an individual is determined by the differences between the sequences of the individual and the reference genome. If the reference genome does not accurately resemble an individual or the population that individual comes from then parts of sequence of the individual will be poorly assembled. Within poorly assembled sequences a portion of the variants will either be incorrect or will not be identified at all. What I propose in

this thesis is a method for improving the discovery of variants from NGS data within regions of high and low variant composition by using the genomic inheritance information contained within pedigrees.

The current practice of NGS relies on a single human reference sequence which is updated on a recurring basis [18]. NGS follows three main steps: sequence generation, sequence alignment, and data analysis.

Sequence generation is done after sample preparation and library generation is done to obtain a high-quality set of fragmented DNA sequences. Common NGS sequencing methods use what is known as sequencing by synthesis [96]. A common sequencing by synthesis process is done by first binding prepped DNA fragments onto a dense glass slide of universal probes known as a flow cell [75]. Once the fragments are bound to unique positions on the slide, they are then amplified so that neighboring positions to these fragments contain copies of the original sequence fragments [35]. The result is a series of spots on a glass slide, each representing a unique clonal cluster of a DNA fragment. Finally, the sequencing process begins by binding universal primers to the set of clusters. DNA polymerase attaches one of four complementary nucleotide terminators, each containing a fluorescent dye, to the primers a single base at a time. After the terminators are bound, the clusters are photographed, the terminators are removed, and the next nucleotide terminator is bound. This cycle is repeated until each strand is completely bound by a series of complementary nucleotide terminators. The result of this process is a digitized record of short nucleotide sequences known as reads.

Other sequencing techniques include long-read sequencers like PacBio sequencing as developed by Pacific Biosciences and the MinION Nanopore sequencer by Oxford Nanopore

Technologies. The PacBio sequencer is a single-molecule real-time method which synthesizes and observes the sequence of each strand of DNA without pausing between bases [93]. In Nanopore sequencing, each strand of DNA is fed through a small pore where a voltage gradient is passed through the pore [12]. Changes in the voltage that are detected would represent a single DNA base in the sequence. Both sequencers share the benefit of producing large reads in a short amount of time. They also require small amounts of initial sample DNA which help to reduce errors resulting from amplification bias. However, they both produce base error rates that lower the accuracy of variant detection. Sequenced base error rates can contribute to mismatching errors within the sequence alignment phase of an NGS pipeline [28].

During sequence alignment, the set of sequenced reads are run on various alignment algorithms that commonly work by a seed and extend method [73]. First a data structure known as a hash-table is used to store all short sets of strings of length k, called k-mers, of a read sequence. The hash-table has the computational property of being fast in the time it takes to look up where a particular k-mer belongs in a read sequence. The seed operation occurs by mapping the set of k-mer substrings from a read to the positions they match to on a reference sequence. Once a position for the seed is found, the seed is extended by adding more of the read sequence to the right and left of the seed. To take into account sequence differences between the read and the reference at this position, the seed extension is refined by running more robust local sequence aligners like the Smith-Waterman [105] algorithm which is a local-alignment implementation of the general Needleman-Wunsch algorithm [78].

More modern alignment algorithms follow this seed-and-extension framework with improved indexing techniques introduced by the Burrows-Wheeler transform (BWT) [13]. The

BWT is a sorting and compression algorithm where a string is permuted into a matrix of all possible single character rotations. Each row in the matrix is then lexicographically sorted and the last column in the matrix is taken as the BWT of the string. The BWT data structure has the benefit of being more compressible and memory efficient. It also enables fast searching for exact matches of specific substrings in linear-time by only using the first and last columns of the Burrows-Wheeler matrix [32].

Various read mappers like BOWTIE and BWA-MEM both use the BWT to do initial seeding and query operations of a read onto a reference genome [61, 64]. BOWTIE alignment works by using the BWT to index a reference sequence [61]. It then goes through a back-tracking process for inexact read alignment. For each read, BOWTIE queries the reference for successively longer suffixes that contain a subset of the read. If the growing query sequence is not found in the reference, then the leading bases of the previously queried positions will be changed following an alignment policy and the algorithm will continue growing the query. The BWA-MEM algorithm also works by using the BWT to index both the reference and query sequence [64]. BWA-MEM alignment looks for the positions on the reference where the longest substring of a read exactly matches a location. These are referred to as supermaximal exact matches (SMEMs). If the SMEMs are too large, then the algorithm will re-seed for SMEMs that cover the middle base where the previous SMEMs were found. It then uses these SMEMs as the seeding locations where it does a dynamic-programming alignment procedure like Smith-Waterman along with an extension-limiting scoring scheme to do the extension step.

What follows sequence alignment is the downstream data analysis of NGS data where one can computationally process and discover patterns in the aligned DNA sequences. One of

the most common procedures of this step is the process of variant calling [73]. Variant calling is the detection of variants at the DNA level in the sequence when compared to a reference sequence [72]. In general, the variant caller uses the proportion of reads at a particular position, known as the depth of coverage, along with read base qualities and read mapping qualities to determine the most likely genotype of the sample at that location.

Common tools that are used in variant calling are the Unified Genotyper and HaplotypeCaller as developed by the Broad Institute for the GATK project [113]. The Unified Genotyper works by computing the likelihoods that a base is observed given what the genotype might be by taking into account only the set of reads that have mapped at a position with a high enough mapping quality and base quality. It then takes the product of these likelihoods to calculate the likelihood of a genotype given the set of reads at a position. The highest likelihood determines the genotype that's called for that site.

With the HaplotypeCaller, its algorithm works through four main steps [113]. First, HaplotypeCaller identifies regions of the genome that show evidence for variation based on the reads relative to a reference. Then for each region it does a local reassembly of the reads to estimate the possible set of haplotypes – associated sequence of variants – and realigns those haplotypes to their respective region on the reference. Third, it determines the likelihoods for each haplotype given the set of read data by running pairwise alignment of each read against every haplotype. Then those haplotype likelihoods are marginalized to obtain the likelihoods of observed alleles at a position. Finally the program uses Bayes' rule to determine the posterior probability of a genotype at that position given the set of reads. The most likely genotype is used to define the variant at that site.

Consequently, the variant calling accuracy mainly depends on the initial read coverage at a particular coordinate, the base quality scores and mapping quality of those reads at a given site [28]. If the base qualities or the depth of read coverage or the mapping quality of reads at a particular position is too low, a small set of incorrectly mapped reads can throw off the variant caller and detect variants that otherwise don't exist in that individual's real genome [28, 113]. In this proposal, the term false positives (FP) will refer to these incorrectly called variants while false negatives (FN) will refer to the set of variants that exist in an individual that were not called by a variant caller. Since humans are diploid organisms, the genotypes of an individual are determined by the proportion of multiple variants that are called at each site. In this way, genotypes can be incorrectly called even if one of the two possible alleles is called correctly at a site. FP's and FN's will also refer to genotypes throughout this thesis.

Although NGS alignment performs well enough for studying genetic variation for common alleles in genome-wide association studies, it produces FPs at a rate that is too high for conducting adequate genome-wide studies of rare variants in regions of the genome that are difficult to sequence [40]. Here, hard-to-sequence areas of the genome are defined as areas that produce low read mapping confidence. These regions are commonly either high or low in sequence complexity. Regions that contain a high concentration of common variants — single nucleotide polymorphisms (SNPs) or insertions and deletions (INDELs) — are defined as being high in sequence complexity while regions that contain large numbers of repeating sequence are defined as being low in sequence complexity. When a particular region shares these traits, the diminished context reduces the ability for a sequence aligner to map a read that originated from these regions [29]. This is due to either the read not containing enough information for

the aligner to confidently map to that position, or the reference sequence doesn't resemble the individual's actual genomic variation well enough in these regions. Since this situation results in a portion of reads that either don't map or incorrectly map to these positions, the pileups of reads for these regions would theoretically be reduced or contain incorrectly-mapped reads. As a result a variant caller that is dependent on these read pileups to determine variants will run into a situation where FP or FN variant calls are made. This reduces the ability to accurately call variants within these regions which can lead to misdiagnosis or underdiagnosis of genetic disorders in clinical practice.

2.2 Alternative References

One of the main problems facing bioinformatics and especially the problem of next generation genome sequencing is the task of sequence mapping and variant calling. Over the last decade the field has found that linear based references are inadequate to discovering variants that are unique to an individual. This is especially true for variants that lie within highly complex or repetitive regions of the genome since they require more information for deciding where a sequence read has come from. This is commonly referred to as a reference bias and has been recently observed to be a major problem when it comes to detecting genes from individuals from admixed populations. Rachael Sherman and colleagues in 2018 found that when compiling and studying a collection of genomes from those with African ancestry that they contained about 10% more DNA sequence than what was present in the current build of the human reference genome, GRCh38.

2.2.1 Diploid Aligner

One idea that had been proposed and developed by Timothy Gall within the NIHUDP is to use parental, subpopulation and personal information during sequence alignment to improve overall variant calling which would subsequently improve the detection of rare genetic variants [40]. The method was born out of the methods developed for the computational pipeline `AlleleSeq` for studying sample-specific allele expression in transcriptomic data [2]. This project was known as the `Diploid Aligner` and its core idea was to use an individual's parents sequence information to modify the linear reference that is used in NGS. The parental variant information can come in the form of SNP-chip or whole-genome NGS data. This parental variant information is used to create three different reference sequences, one representing the mother, one representing the father, and the third being a concatenation of the two. Some regions, like the sex and mitochondria chromosomes, are not diploid and therefore require special treatment. For those special cases, the paternal set of variants on chromosome Y would be constructed as part of the paternal reference and the mitochondrial chromosome would be comprised of maternally-derived variants and would go into the construction of the maternal reference.

The child's reads are then aligned to each of the three new references. The parental reference that the read best maps to will be used as the actual mapping coordinate for the read. A summary of the parental and concatenated alignment process is illustrated in Figure 2.1. This modification of the reference would result in references that better match the sequence of the individual which in turn would improve downstream variant calling. Its motivation was

driven by the NIHUDPs need to achieve the highest possible accuracy in order to minimize FPs within individuals with rare genetic disorders. Currently, the NIHUDP's main role is to take in patients with undiagnosed rare diseases that have already had exhaustive workup from many other institutions and hospitals, so conventional methods will likely not be enough to figure out their disorders. The best practices of reducing FPs within the variants of an individual are well developed by various filtering techniques and linkage analysis, but these methods are downstream from the sequencing and alignment processes of NGS meaning they are dependent on the quality of alignment [92, 70].

The Diploid Aligner attempted to improve the accuracy and recall of aligning NGS sequence reads to a canonical reference. In theory, this would improve the prior probability of having a template that increases the likelihood of mapping reads to where they actually came from in the individual's genome [40]. The higher likelihood of correct mapping produces more read coverage of greater mapping quality in high or low complexity regions. As a result, this would reduce the presence of FPs and FNs within the variant calling process as there is more high quality information available to resolve variant discrepancies within these positions. Since 2016, the diploid alignment project had been discontinued, but it's principles and concepts are carried on in the work presented in this dissertation. Though the project was incomplete, there had been a proof-of-principle study of work showing that it does improve the overall mapping quality of the alignment (0.111% increase of unique read mapping) relative to alignment to a traditional linear reference genome. A brief summary of the results can be seen in Table 2.1 below, where 10 NIHUDP family cohorts were run using the Diploid Alignment pipeline and alignment quantity and gapped alignments were compared with those of traditional NGS

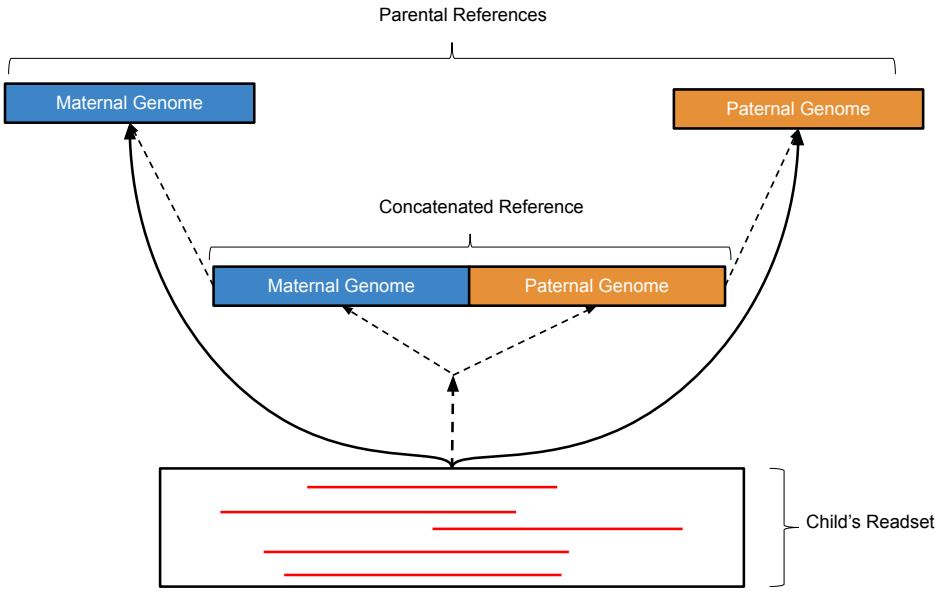


Figure 2.1: Main schematic of the alignment step in the diploid alignment pipeline. For each read sequenced from the child, the read is aligned to each of the two parental references and once to the concatenated reference. The side that the read best aligns to in the concatenated reference (dashed arrows), either the maternal or the paternal side, decides which of the parental reference alignments (solid arrows) is chosen to be the actual alignment for that read.

	Total %	stdev	min	max	#Families
% Increased aligned reads	0.036	0.007	0.021	0.042	10
% Increased aligned read pairs	0.041	0.007	0.025	0.047	10
% Decreased gapped alignments	90.8	0.3	90.5	91.2	10

Table 2.1: Preliminary alignment results from using the diploid alignment pipeline on 10 NIHUDP family cohorts. These read alignment results are relative to single-reference based next-generation sequencing using bwa and GATK best-practices pipeline on an HG19 human genome reference sequence.

pipelines.

2.2.2 Genome Graphs

Another project that is similar to the Diploid Aligner which also attempts to solve the reference bias problem is the Variation Graph (VG) toolkit. VG was developed within the Computational Genomics Lab at UC Santa Cruz where the goal is to develop a different way of representing the reference sequence by using a graph-based data structure to better capture the diversity of human genetic variation and to develop an aligner which can align NGS reads to this graph [44]. The graph-reference, also known as a genome graph, is a series of nodes representing bases in a sequence that are connected to each other by edges. A sequence of bases then can be represented as a path of connected nodes in a graph. In this way, the genome graph can store variant and structural information as 'forks' and 'merges' in the sequence path, where each path can represent a unique haplotype. This is an extension of the ideas established by partial-order alignment (POA) sequence graphs [62]. POAs were established to represent multiple sequence alignments (MSA) in a more robust way. The directed edges of a POA establish

the ordering of sequences much in the same way multiple paths in a VG graph represent multiple sequences. Figure 2.2 below shows an example of how sequences and structural variation can be represented by the VG graph scheme.

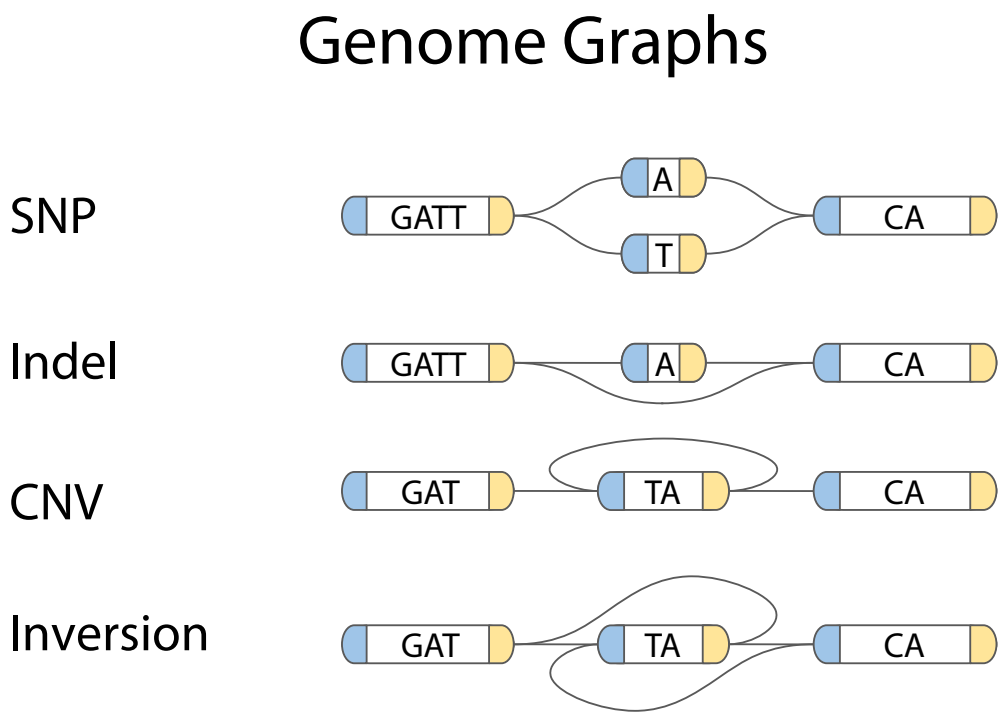


Figure 2.2: Example genome graphs. Each segment holds some number of bases. A join can connect, at each of its ends,to a base on either the left (5', blue) or the right (3', yellow) side of the base. When reading through a thread to form a DNA sequence, you leave each base on the opposite side from which you entered, and reverse complement it if you enter on the 3' side and leave on the 5' side.

The example graph above shows the capabilities of the system. One thread of this

graph can represent the sequence “GATTACA” (reading from the left side of the leftmost sequence to the right side of the rightmost sequence, along the nodes drawn in the middle). Other alternative paths include three variants: a substitution of “A” for “T”, a deletion of an “A”, an arbitrary number of copies of the ”TA” sequence and an inversion of a “TA” sequence [44].

The main advantage of this data structure is that it can store more sequence variant information to better capture the common structural and sequence variation found in human populations. Another advantage is that the set of haplotypes contained within a genome graph can be compressed and indexed. The VG graph compresses haplotypes by using the Graph Burrows Wheeler Transform (GBWT) [101] which is a scalable extension of the Graph Positional Burrows Wheeler Transform (gPBWT) [79, 30]. The GBWT enables fast and memory efficient storage and subhaplotype queries for quick searching and counting of haplotypes. Efficient haplotype queries are useful for cases of checking for haplotype consistency when mapping reads to a graph [79]. In the case of read mapping onto a genome graph, the VG project uses GCSA2 indexing which is a k-mer based index that is stored within a de Bruijn graph datastructure [104, 100, 101]. When aligning NGS reads to this graph, the alignment process will take into account variants that are theoretically more likely to contain a haplotype that better matches an individual’s sequence relative to a linear reference [29]. The VG alignment process is similar to the Diploid Aligner in that its goals are to improve sequencing by altering the prior probabilities that a read comes from certain regions of the genome, given the reads observed mapping position.

Using VG graphs one can design graph references and sequence mapping algorithms that take advantage of the greater information provided by these new references in order to bet-

ter decipher the location of a sequence read. The primary algorithm for which sequence reads are precisely aligned to a graph is done using a seed-and-extend approach where read SMEMs are seeded using the GCSA2 and GBWT index and the extension and alignment is done using Partial Order Alignment (POA). POA is an extension of the Smith Waterman alignment algorithm where a dynamic programming matrix is filled in with the alignment scores of all ways a read aligns with a reference [62]. In addition to this alignment algorithm, POAs fill a multi-layered DP matrix where each branching layer represents a corresponding branching path in the graph. Each cell in this matrix is filled based on a gap score which reflects greater penalty for alignments that permit more insertions and deletions than one with more exact matches. The specific score given to a cell is based on the minimum cumulative score of its previous neighboring score. Once each cell is filled out, a traceback algorithm is done to find the path through each preceding cell in the matrix that gives the current cell its score. With POA DP matrices, the traceback algorithm simply chooses which branching cells produced the minimum score of the current cell. There are cases where multiple alignment solutions are equally optimal in terms of gap penalties. In such cases an arbitrary decision is made which path to take. In practice, this can be tricky since it introduces non-parsimonious solutions that can differ between each read sequence aligned to a site. To resolve this, INDEL-realignment is implemented on a series of alignments at each potential INDEL in a read. INDEL-realignment looks at reads at these INDEL sites and uses consensus to determine if an INDEL is present, and if one is then how that INDEL is represented amongst the reads. This produces a consistent alignment representation that helps cleanup the alignment and gives variant callers more confidence when calling INDEL variants.

With better sequence mapping comes greater information available to genotypers to detect variants in these regions and thus provides greater variant recall for variants that are unique to an individual. Now an additional problem presents itself, what makes a good graph reference? There are a number of techniques each designed around optimizing the amount and type of variation present in a graph. Through experiments, we've found that including more and more variation to a graph reference can have a counterproductive effect on performance. For example, if one were to introduce variants at a location in a genome that contained only one difference from another locus in the same genome then ambiguity has been introduced that results in less confident mapping. This potential for introducing variation that increases the entropy of the reference is mainly focused in the areas of the genome that are highly repetitive or recently duplicated from an ancestral genomic sequence. This phenomenon can be seen below for copy number variants (CNVs) in Figure 2.3 and for SNPs and INDELs in Figure 2.4.

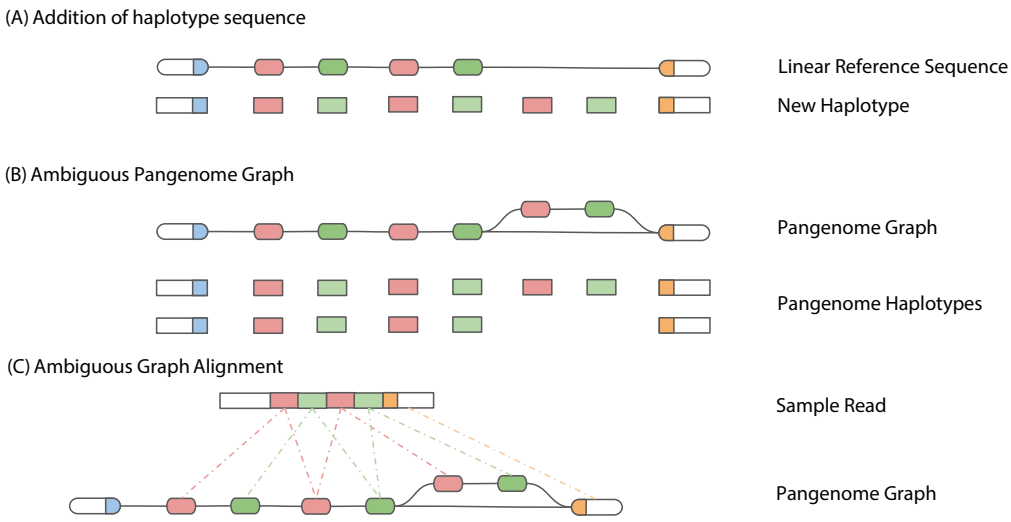


Figure 2.3: Simplified example of how introducing haplotypes can introduce ambiguity in graph alignment to a graph with CNVs. (A) A linear reference sequence with a single sample haplotype that will be added to it. (B) The result of constructing a graph using the reference sequence and the added haplotype. The graph contains a variable number of copies of the CNV sequence in red and green. (C) Alignment of a read where seed match support for the CNV sequence ambiguously maps the read to three potential places on the graph.

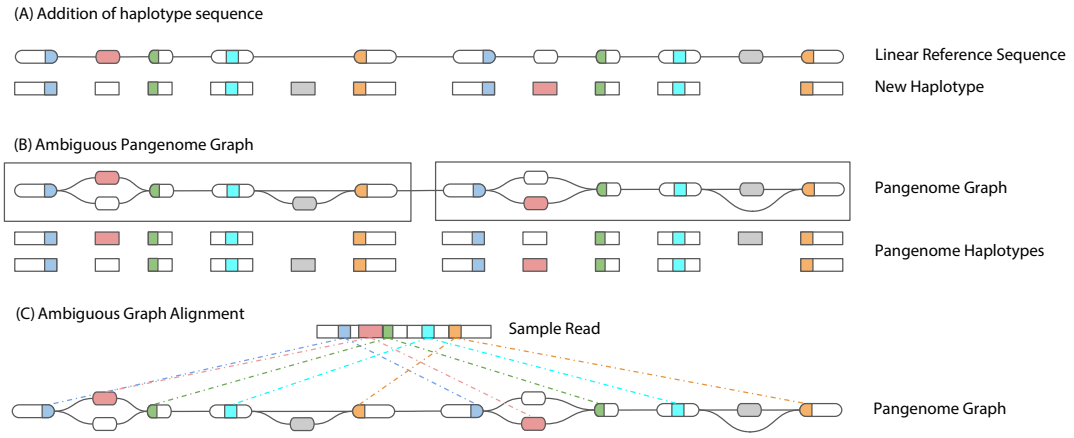


Figure 2.4: Simplified example of how introducing haplotypes can introduce ambiguity in graph alignment to a graph with SNPs and INDELs. (A) A linear reference sequence with a single sample haplotype that will be added to it. (B) The result of constructing a graph using the reference sequence and the added haplotype. The resulting pangenome graph contains consecutive graph subsequences (black boxes) that exactly match while representing two different haplotypes. (C) Alignment of a read where seed match support is equal between the two matching graph regions.

2.3 Pedigree Graph Alignment

What I present in this dissertation is a combination of these two ideas in what I refer to as VG-Pedigree. This aligner would merge the population-level data inherent in the Pangenome graph with the familial genetic awareness of the diploid aligner to produce a new graph-based canonical reference that's tailored to an individual's pedigree. The basic premise of the project

is to enhance the VG alignment and variant calling processes using the parental variant information obtained either through SNP-chips or whole-genome or whole-exome sequencing. The parental variants will be used as additional prior evidence for picking the most probable set of haplotypes that define an individual. Figure 2.5 illustrates an outline of the VG-Pedigree paradigm.

Alignment of the individuals sequence to this altered Pangenome reference will reveal a series of variants. Due to the preliminary work of the Diploid Aligner, it is possible that that a large number of these variants will either be Mendelian inconsistent due to sequence allele-bias in NGS data [40]. Allele bias arises from uneven amplification of DNA strands during the library preparation process of NGS [46, 83]. This results in the frequency of certain alleles to deviate from their expected binomial-distribution. Mendelian inconsistency occurs when the genotypes of the individual and their parents do not follow a Mendelian model of inheritance [14]. For instance, if an individual is heterozygous at a position on its genome while its parents are both homozygous at that same site, then that pattern would indicate a Mendelian inconsistancy. There are cases where not every Mendelian inconsistancy can be corrected as a small fraction of these variants are true new-mutations in the child, termed `de-novo` variants. Though there are an expected number of approximately 50-100 `de-novo` variants per individual, there are 1,000's to 10,000's of `de-novo` variants produced by whole-genome NGS datasets [57, 34, 5]. So correcting or filtering out these errors will still produce more accurate results.

These problematic regions will need to be resolved by restricting the structure of the Pangenome graph reference to the variants and haplotypes that are consistant with the variants present in the pedigree. This can be done by doing a first iteration of alignment of sequence

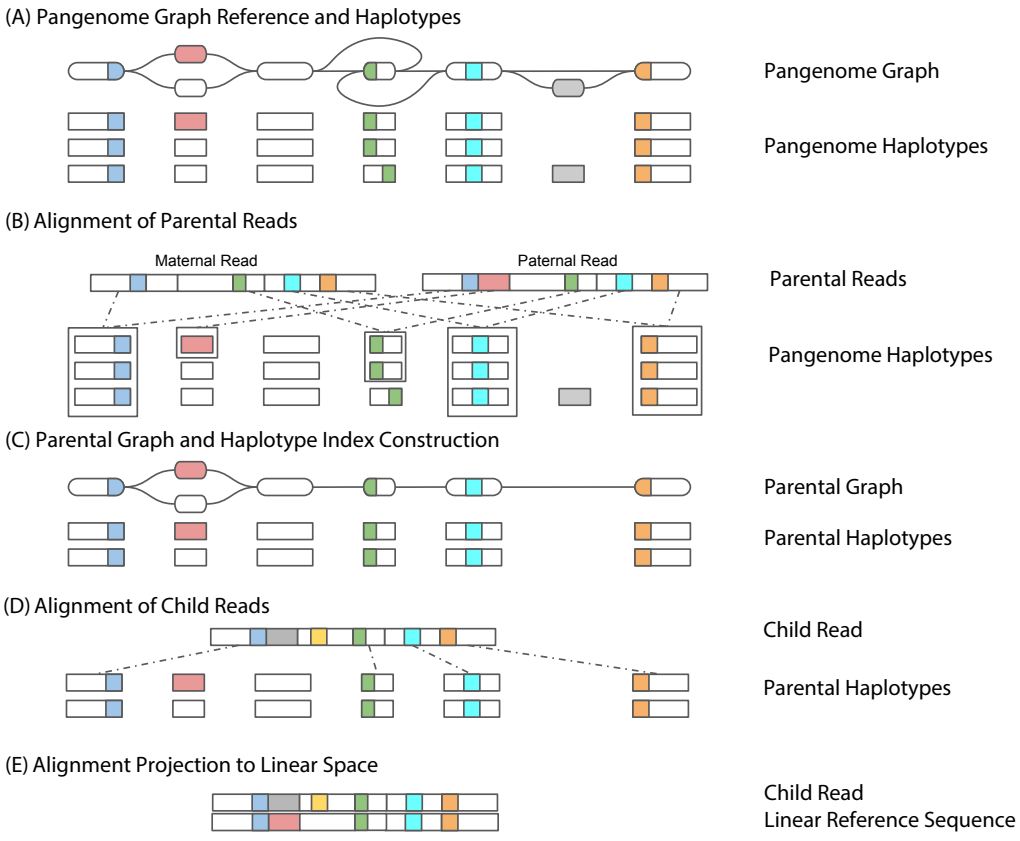


Figure 2.5: Main schematic of the alignment workflow in VG-Pedigree. (A) An example of a pangenome graph reference and the haplotype set that coincides with that reference. (B) Alignment of parental reads to the haplotypes of the pangenome reference. (C) Construction of the parental graph reference and haplotype set. (D) Alignment of the child reads to the haplotypes of the parental reference. (E) Projection of the alignments in (D) to the linear reference space.

data from pedigree samples to a pangenome graph. Then variants are jointly-called between the parents and the child to follow genotyping consistency with the alignment data. Then the pangenome reference is modified using the genotypes called for the parents in the joint-called variant set. The haplotype set can be defined by first phasing the genotypes of the parents with respect to the genotypes of the pedigree and the read alignments that span 2 or more variants. The phased sets of parental genotypes can then be incorporated into the modified pangenome or parental graph reference.

Once this editing is done, the individuals reads that overlap these problematic regions will need to be realigned. This cycle of editing the pangenome reference and realignment can be iterated until the alignment errors converge on a minimum. Afterwards, variant calling will be made on the pileups of the reads after projecting the pangenome alignments to linear reference space. Projecting alignments to linear sequence space opens up the ability to use well-established linear-based variant callers.

2.4 DeepVariant Variant Calling

Another area of optimization is in the problem of variant calling. Recently, new developments in machine learning have helped to improve the accuracy of variant calling by using Convolutional Neural Networks (CNNs). CNNs are neural networks that are trained to decide and successfully classify various features of spatial information. Because CNNs are neural networks, they are trainable and can therefore be improved by the amount of new and diverse datasets that are available. This class of neural networks work by decomposing the

problem of image recognition by breaking down an image into a number of subsegments. A matrix which is designed to recognize a certain aspect of images such as a line or a circle, known as a filter or kernel, passes through and performs a dot product against each subsegment of the image and returns a numerical value that represents the strength of the particular filter that recognizes the subsegment of that image. Collecting these values results in a set of matrices of recognized shapes, one for each matrix, known as a feature map. This process of passing a filter through an image is known as a convolution. After feature maps are produced, each chunk within each feature map is pooled together to produce a simplified feature map. This process, known as pooling helps to reduce the computational complexity of processing each layer while also maintaining enough detail for feature recognition and reducing the likelihood of overfitting. This process of convolution and pooling is repeated for a number of image feature matrices through a number of layers resulting in a vector representing a list of numbers, one for each class that's to be interpreted.

In the case of variant calling, the problem of identifying the most likely genotype based on a set of overlapping reads can be reconstituted as a spatial problem that CNNs can solve. What CNNs offer in this case is a more systematic approach to detecting neighboring noise in the read pileups when determining the likelihood or quality of a genotype at a specific location. If a pileup contains too much inconsistent or noisy sequence information directly around a position that is being considered for a genotype, then the probability of guessing a correct genotype would decrease as a function of that noise. Different topologies of neighboring sequences like a tandem repeat that neighbors an INDEL variant would cause a genotyper to produce inconsistent results. But if a CNN were to successfully recognize this feature and its

filters and layers were designed to recognize and know of validated examples of what should be there, then the CNN would be able to more accurately determine the uncertainty of calling a genotype in those locations.

DeepVariant is a software workflow developed by the Genomics team in Google Health at Google Inc. that applies the techniques of CNNs to solve the problem of variant calling based on pileup images [86]. In the first section, the workflow works by taking the set of aligned reads and scans for regions that are different from a canonical reference genome. Candidate regions are determined in a similar way as is done in the HaplotypeCaller algorithm. First active regions are determined from the read pileup differences from the reference at their respective mapping position. Next, a De Bruijn graph is constructed from the set of k-mers that are derived from the reads aligned at the pileup in an active region. Common paths in the graph with edges weighted by frequency of connecting k-mers determine the set of likely haplotype sequences. Then reads are realigned against this set of candidate haplotypes, picking the best scoring alignment as the best alignment for that read. Finally candidate regions are determined based off of the realigned reads in the active region. From this set of candidate regions, the next phase of the DeepVariant workflow prepares the data for CNN classification by converting them to a set of feature images that are subsequently merged into a combined image. The following figure illustrates the details of this image encoding scheme. Once a set of images are compiled from the candidate regions, the pre-trained CNN processes each image to determine a most likely genotype for that position.

Internally, the CNN is trained by first taking an initial randomly parameterized or previously trained input model along with a set of pileup images labeled with the genotypes

that are known to exist in those images and runs them both through a training cycle. The training cycle processes the pileup images through the initial CNN model, checks the output to determine the amount of error with respect to the expected genotypes, uses the stochastic gradient descent algorithm to suggest a more optimal change in the CNN model, and runs that model through the cycle again. Stochastic gradient descent works by taking the error as calculated by an error or objective function that estimates the amount of error a particular model configuration makes with a random shuffle of the training dataset and returns a set of parameter values to change in the model. Subsequent iterations will result in a lower value of the objective function with respect to the parameter set used for the model. This training cycle is repeated until a certain predetermined number is reached or the estimated error converges towards a minimum that is past a certain threshold. The advantage of this system is that it doesn't require prior knowledge of genomic topology or structure in order to establish a working model. The authors demonstrate in their study that this framework can generalize such that variant calling performance is robust to using different versions of a genome reference for human data or even when applying DeepVariant trained on human data in order to call data on different mammal species.

More recently, a pedigree-based extension of DeepVariant was implemented known as DeepTrio [56]. DeepTrio works in the same way as DeepVariant except that the candidate regions and images are determined at once where if a candidate region is present in one sample then the same region is considered for the other two individuals in the trio. This enforces consistency in joint calling where a genotype call is made in the same spots for every individual in the trio. The CNN model for DeepTrio is also trained differently where a child and parental

models are trained separately. Child models are based on images compiled from pileups from both the parents and the child with the labels being based on child truth-set data. The parental models are based on the same merged pileup image of the trios reads at a candidate region except the individual parent's pileup is used with some support of the child's pileup along with the labels being based on the individual parent's truth-set. Since Mendelian information can assist in determining a more accurate genotype likelihood for the offspring, parental pileup information is used to enhance the training of the child model. However, total enforcement of Mendelian consistency can filter out real `de-novo` variants, so some leniency in the model training needs to be implemented.

2.5 Software Workflow Frameworks

Basic workflows are fundamental to the field of bioinformatics since most bioinformatic problems require many steps in order to process complex data. They require the right computational tools and frameworks for processing large whole genome data on different compute environments. This is important because the scale of the data from which the programs developed in this thesis aims to process is large and will require the resources of a compute cluster. There are many services that exist to provide scalable compute environments for scientific computation [115, 121, 49]. Due to the number of available services, some tools are needed to abstract construction of this application such that it can be compatible with as many systems as possible.

There are three main categories of software frameworks that together help one build

large, scalable, and portable bioinformatic applications designed for cloud compute environments. These are containerization systems, workflow languages, and batch systems/workflow-execution engines. The 1st of these, containers, includes the commonly-used Docker which provide a framework for packaging up software packages such that the user no longer needs to worry about not just application-specific dependencies but also operating-system level dependencies [53]. This makes writing and debugging more complex applications on larger compute systems easier for the user. The user can also chain different containers of software together such that they can build and mix the order of containers that fit specific use-cases in an application. Singularity is another containerization system which, much like Docker, provides ways for a user to package software into containers. However, the difference with Singularity is that it fixes a fundamental issue with Docker that makes it harder to work with controlled-access data or compute resources. Singularity gives the user complete control of their container environment without the requirement of being a root user [59]. Since most controlled-access data sources or compute resources are designed to be highly secure, one of those securing measures is restricting access to data on a need-to-use basis. This means that a user can not by default have access to an entire file system or software environment that a user of Docker would require. Since most of the data used in this thesis is controlled access human data, we will primarily focus on systems that enable this functionality, and thus will be using Singularity as the primary containerization engine for this project.

The other component to writing sustainable applications are workflow languages. Workflow languages provide a higher-level, simpler interface from which a user can write complex pipelines with. The most common of these are the Workflow Description Language (WDL)

and the Common Workflow Language (CWL) [115, 25]. Both of these languages provide a number of features that are abstract enough and simple enough in syntax that workflow engines are able to more easily support than for lower-level scripting languages like python or java. The downsides are that they are typically much less powerful in terms of the number of different things one can do with them than for lower-level languages. Critical features of WDL and CWL provide definitions and configurations for running tasks that each define what programs are used, what inputs are processed and what is output from that task. They typically provide a way to define branches and merges between these tasks as well as switches for user-desired execution paths of a workflow.

The last component, workflow execution engines, are the main system that are responsible for coordinating the execution of a workflow on a cloud or cluster compute environment. The most common workflow execution engines for WDL and CWL-based applications are Toil and Cromwell [114, 115].

Toil enables a user to write and run workflows either using workflow languages like WDL or its own in-house python language for defining task component jobs and how those jobs connect to form a workflow. Toil supports various cluster systems including local HPC systems or cloud environments like Amazons AWS or google's cloud platform (GCP). It also supports various job schedulers and batch systems which helps make workflows more compatible with more systems. It also supports call caching where if a previous result has already been computed and stored in its hash then Toil simply looks that value up instead of wasting valuable compute resources recalculating that value. Toil also supports various other quality-of-life features that help make applications more robust to transient errors. These include fault-tolerance,

restartability, and elastic scalability that runs more compute when there's more input data to process as well as spot-market pricing which uses cheaper but less consistently available compute resources to be used to reduce costs.

Cromwell is another workflow engine that features many of the same things that Toil supports. It can work with various workflow languages like WDL and CWL. It also supports various compute scheduling systems and cloud systems. But unlike Toil, Cromwell is not as featured especially with respect to traits like call caching and fault-tolerance. That's mainly because Cromwell was designed for simplicity and as a result it is much more reliable than Toil when it comes to running workflows on the kinds of systems that it supports.

Part II

Basic Graph Genome Pipelines

Chapter 3

A Pipeline for Pangenome Analysis

3.1 Preamble

In this chapter I present a couple of projects that I have contributed to which tackle the problem of reference bias at the scale of a single sample of sequence data. I describe two papers in which I will detail a summary of their algorithms which I was not directly involved with developing, but I was involved in identifying critical issues, constructing genome graphs that improved performance of these algorithms and developing software workflows that have improved usability and performance for other users. The first paper, VG Toolkit, describes the core concept of graph genomes and a toolkit of programs that enable sequence alignment, graph construction and manipulation, and variant calling of graph-based alignments [44]. For VG, I contributed to identifying critical bugs and organized benchmarks using real sequence data to assess performance and quality. The second paper, Giraffe, I present my contributions to construction of a pangenome reference that maximized mapping and variant-calling performance

for variants that are unique to a sample. I will also cover my development of the software suite of workflows that streamline the process of mapping and calling variants from pangenome graphs.

3.2 Introduction

Advances in genome sequencing technology have provided researchers with a wealth of data. Much of this data is used in investigating the large sets of genetic variation that is present in individuals, however much of the variation is poorly characterized within difficult to sequence regions of the genome. These difficult regions span segmental duplications as well as regions of high mutation density. Mapping to a linear reference has been found to contribute to poor performance in detecting variants located in these difficult regions. This is mainly due to the linear reference lacking information about orthologous haplotypic variants which results in mismapping of reads to locations that they did not originate from.

3.2.1 Graph Genomes

There are a number of techniques for overcoming this potential performance problem and they all are concerned with what variants are present in a reference. The main idea here is to reduce the variant set present in the graph reference down to just the variants that are most relevant to the individual. One strategy is to only include variants that are known to be common based on population allele frequency data present in various population databases. Some common consensus databases include the 1000 genomes project, the GnomAD project,

the UK10K database, and the all-of-us project [9, 54, 118]. The 1000 genomes project is one of the earliest projects to produce a diverse and well-curated dataset of phased genotypes. The latest release provides 2,504 samples amongst 26 populations with a total of about 84.4 million variants [9]. GnomAD is a more recent project which gives a much larger dataset of 71,702 diverse non-related samples totalling 602M SNPs and 105M INDELs [54]. However, the main downsides to this dataset is that a portion of it includes samples from specific genetic studies that are prone to sampling bias and accumulation of patient sample data. The allele frequencies of this dataset are likely to not be representative of larger populations. So use of this data requires some preprocessing work in order to tease out potentially biased samples. It also does not contain phased information which is critical to optimized pangenome alignment. For the UK10K dataset, approximately 10,000 samples from various disease and general cohorts around the UK were sequenced to provide a resource for identifying rare variants. Similar to GnomAD, the downsides of this dataset are that they include not only samples from specific disease cohorts, but also rare variants which are not of particular use for general pangenome graph alignment [118]. In addition to sequence variation, haplotype information can also be embedded in the pangenome graph. This is typically done with the incorporation of phased variant data which ultimately labels paths in the graph that indicate which variants belong to the same haploblock.

Using these principles we can determine that a likely optimal workflow will require a population-based graph reference that includes not only common variants but also haplotype information that helps determine valid mapping proposals during the seeding process of graph alignment. Another avenue of improvement to reference sequences is the use of sequences that

steer away the mapping of read sequences that originate from error-prone regions of the genome. These are known as decoy sequences and they are primarily comprised of the Epstein-Barr virus (EBV) virus which is very commonly found in most individuals genomes due to the ubiquitous spread of the virus. However, a majority of the decoy sequence mainly includes common human sequences that have not found an adequate place in the reference genome. The decoy sequence has a number of benefits. It allows for reads that an alignment algorithm would normally spend a large amount of time attempting to align to the reference to instead more effectively align to sequences that are a better match. It also guides reads away from critical regions of the genome and reduce erroneous sequence information from corrupting the task of calling variants in these regions. So in summary the decoy sequence aids in reducing false positive variant calls while also speeding up the task of read alignment.

3.2.2 Faster Mapping with VG Giraffe

Potential optimizations in the workflow can been made in a couple of areas. One is in the mapping stage with the use of the VG Giraffe mapping algorithm. The Giraffe algorithm takes advantage of the haplotype structure of a graph reference and attempts to produce gapless alignments of the reads prior to resolving mismatches through dynamic programming based on two sets of assumptions: that common INDELs are already present in the graph and that most errors produced by Illumina short-read sequencers are in the form of SNPs.

Giraffe uses a GBWT based on sampled haplotypes and a distance index to more accurately and efficiently map and cluster paired reads to the graph. It does this in four stages. The first stage is in finding seeds for each read which critically relies on the use of minimizers.

Minimizers are a subclass of k-mers which are a set of short subsequences that are of fixed length k. Historically, encoding and indexing all subsequences of length k of a reference or a set of reads is computationally or memory intensive and so a more efficient implementation is to use a common subset of k-mers in a sequence to reduce this resource requirement. Minimizers are chosen such that a subset of k-mers which represent common k-mers within overlapping windows best represent the overlapping windows according to some hashing function. Due to the nature of the graph reference containing more information than that of a linear reference, we assume that there exists a haplotype path that more completely matches at least some the read set than that of a linear reference. Because of this assumption, we can use longer minimizers which will cause the mapping algorithm to examine more unique sequences for read anchoring.

The second stage of the mapper is the use of a distance index for clustering anchors. This index is based off of a tree data structure which represents a hierarchical decomposition of the graph. The index allows for fast computation of the shortest distance between any two positions in the graph reference. In order to compute this index, snarls need to be defined in the graph which are subsequently computed through another tree datastructure. The snarl tree structure is used to succinctly describe this nesting topology and serves as the basis for calculating the minimum distance between any two pairs of nodes in a graph.

To calculate the minimum distance between any two nodes, paths between nodes need to be established. The snarl decomposition guides this path-finding algorithm by using the principle that any path between boundary nodes of snarls in a chain must pass through the boundary nodes of every snarl that lies between those nodes in the chain.

The minimum distance index is constructed in the node order of the post-order traver-

sal of the snarl tree. To compute distances for each snarl, a Dijkstra algorithm is applied to find the shortest path from each child of the snarl to the snarl itself. For each chain the child snarls and chains of that chain are looked up in the snarl index to compute the distances for that chain. With the minimum distance index established, the minimum distance for an arbitrary pair of nodes can be computed.

The other component required by the VG Giraffe mapper is the clustering algorithm. In seed and extend-based algorithms like short read mappers, the substring seeds of each short read is mapped exactly to a specific position on the reference. The extension step requires an clustering of these seeds. In the Giraffe space, the minimum distance index is used to cluster these seeds.

Once the seeds are clustered, the Giraffe algorithm moves on to extending the seeds. During extension, the alignment boundaries of each seed within their respective clusters are grown until a pre-set number of mismatches are present. Dynamic programming is then done on the extended seeds to form a gapped alignment.

One critical component of Giraffe is the path covering and downsampling of the GBWT index. The reason for this is because there can exist haplotypes in the GBWT where an error in a read will result in the read appearing to map more correctly to one haplotype over another, resulting in a false positive mapping. There are also contigs with no known location in any haplotype that can exist and are useful in the activity of mapping. These can include the Epstein–Barr virus, unlocalized contigs and decoy contigs. So prior to using the GBWT index for Giraffe mapping, we first do two things. First is to create a path cover that includes these miscellaneously-categorized contigs and include this path in the GBWT. Then we downsample

the haplotypes embedded in the GBWT for the contigs that contain haplotypes.

For paired-end reads, each read can inform the likely position of their mate since the distance between the two are known a-priori. This is known as the fragment length and the Giraffe algorithm looks at a paired reads fragment as a function of their fragment-length distribution in order to estimate where each read is expected to lie in the reference. Giraffe then maps each read in each pair separately to obtain a set of seeds. The algorithm then attempts to cluster these seeds by which read they came from and looks to see how far apart the two sets of clusters are. If the clusters are close enough as defined by the fragment length distribution, then they are clustered together. Otherwise, one of the two read pairs will be rescued through realignment using the distance and minimizer indexes.

3.3 Graph Construction Workflow

Construction of a reference genome graph comprises a number of indexes. These include the VG graph which is a directed acyclic sequence graph (DAG) constructed from a linear reference sequence in FASTA format and a set of variants represented in VCF format. DAGs are useful graphs in that they don't contain cycles which enables linear ordering of nodes. DAGs also enable topological sorting of nodes which allow for more efficient representations of sequence data and permits the use of efficient algorithms for searching and indexing this representation. The XG index is a space-efficient representation of the graph which is mainly used for tasks that don't require modification of the graph. The GBWT index is the haplotype index. It stores similar threads that span nodes in the VG graph in a space-efficient manner.

The GCSA index is a kmer-based index which enables the graph mapping algorithm the ability to quickly lookup where subsequences of reads lie in the graph reference. One important step in graph construction, particularly for GCSA indexing, is sequence pruning. This is needed in order to reduce the nodes present in complex regions of the graph which helps to dramatically reduce the number of kmers that need to be indexed during GCSA construction.

Additional indexes are required for the faster VG Giraffe alignment algorithm. These include the snarls, distance, graph GBWT, and minimizer indexes. The snarl index is constructed from the concatenated VG graph file using the command `vg snarls`. The distance index is constructed using both the XG and snarls index with the command `vg index -j`. The graph GBWT and down-sampled GBWT index is constructed from the XG and GBWT index with the command `vg gbwt -o -g`. Finally, the minimizer index is constructed with the distance index and down-sampled GBWT and graph GBWT using the command `vg minimizer`. Figure 3.1 illustrates the complete workflow diagram.

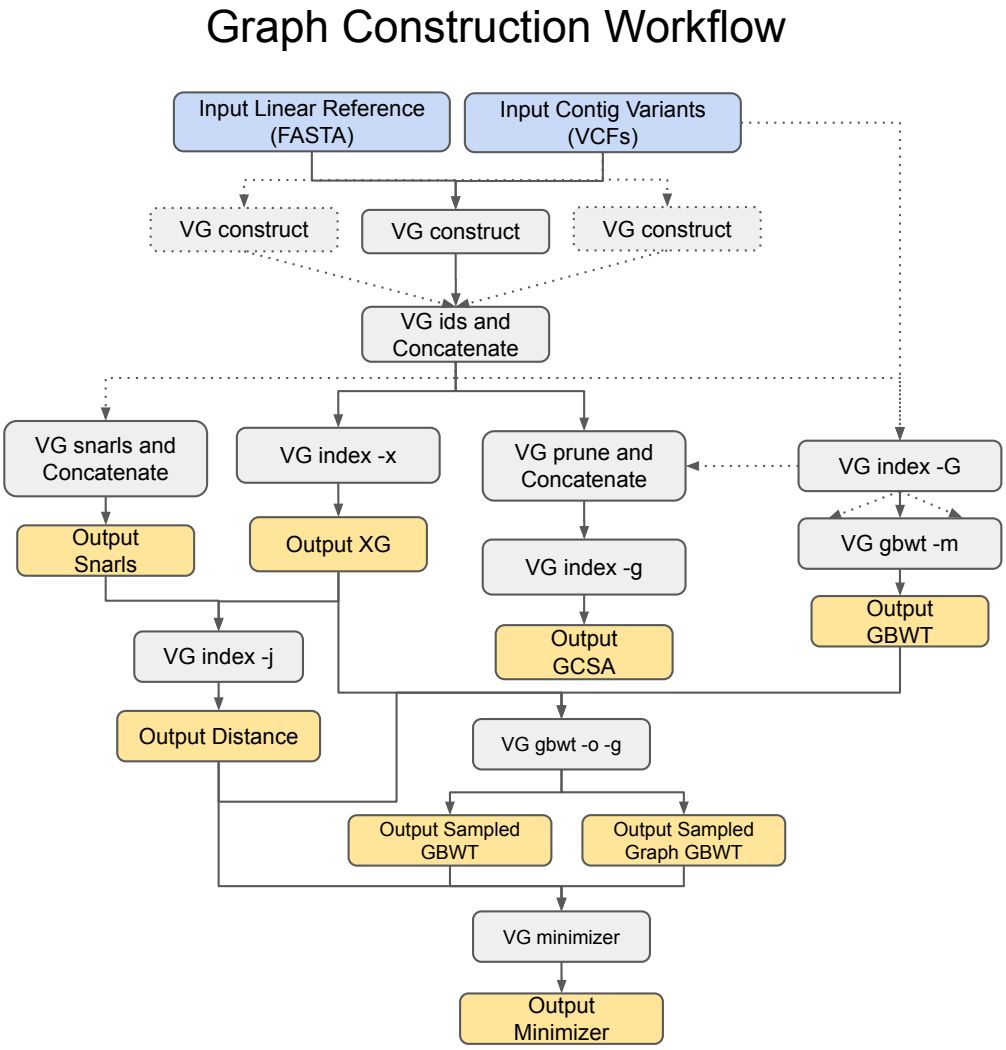


Figure 3.1: Main schematic of the graph construction workflow. Dotted lines represent optional paths or multiplexes of command execution.

3.4 Complete Single Sample Mapping and Calling Workflow

The following workflow implements the necessary steps from graph construction to sequence alignment to variant calling for a single sample. First a graph reference is constructed using a population of variants represented in VCF format and a linear reference sequence represented in FASTA format. The graph construction workflow takes in these two inputs and at minimum generates a concatenated VG graph file along with an XG and a GCSA index. Optionally, if the population of genotypes contained within the input VCF contains phased variants, then a GBWT index can be built which can enhance the graph alignment algorithm. Following graph construction the basic mapping and alignment workflow takes in sequence reads either in FASTQ, BAM or GAM format and at minimum the XG, GCSA indexes of the graph reference. Currently there are a number of aligners in development that do this at an adequate scale for whole genome data. The most developed are `vg map` and `vg mpmap` or the multi-path mapper.

Each mapper runs a seed and extenstion-based mapping algorithm with POA for more precise local assembly. First the algorithm finds the Super Maximal Exact Matches (SMEMs) mapping of the read to the reference by searching a suffix array representation of the GCSA index. SMEMs are subsets of read sequences that match a reference while not overlapping other MEMs in the same read. We then chain these SMEMs together by making a Markov model and using the Viterbi algorithm to find the most likely set of candidate chains of SMEMs. For the Markov model, nodes represent positions in a particular path in the graph reference where a SMEM matches and the transitions between nodes represent probabilities proportional to the implied INDEL length between SMEMs based on distances that are determined by the

corresponding paths in the reference graph. Finally the sequences that represent the chains of these SMEMs are aligned using the dynamic programming POA described earlier. During this process the graph reference needs to be modified in a couple of ways. First is that the graph inherently represents cycles for inversions and copy number variants. In order to run POA against these regions, the graph needs to be acyclic. If inversion sequences are present in a portion of the graph that a chain is aligning to, then a reverse complement of the sequences will be made and represented as an additional node in the new locally-modified graph reference. If copy number variant sequences are present in the graph, then a pre-determined number of copies of that sequence are made and alternative paths are generated in the altered reference graph each representing a different number of copies of that sequence.

Expanding on this concept of graph alignment is the problem of aligning to multiple legitimate paths in the graph reference. The multipath mapper `vg mpmap` simultaneously takes into account each possible path that a read aligns to and preserves that information for subsequent variant calling. `vg map` on the other hand only finds the most likely path that a read aligns to and outputs just that information for the variant caller.

A number of options can be done at this point. Either the graph mapper outputs read alignment records to a graph based in Graph Alignment Map (GAM) format or it can project the alignments to the linear path as defined by the linear reference used in the initial graph construction to produce an alignment file in Binary Alignment Map (BAM) format. From this result, one can call variants a number of ways, either by calling graph-based alignments from the GAM using `vg call` or the more commonly-used linear-based alignment callers for BAM files like GATK HaplotypeCaller or Google's DeepVariant. Once each contig alignment is called

with the chosen variant caller, the results are concatenated and sorted into a final output VCF or genomic VCF (GVCF) file. Figure 3.2 illustrates the complete single-sample mapping and variant calling workflow schematic.

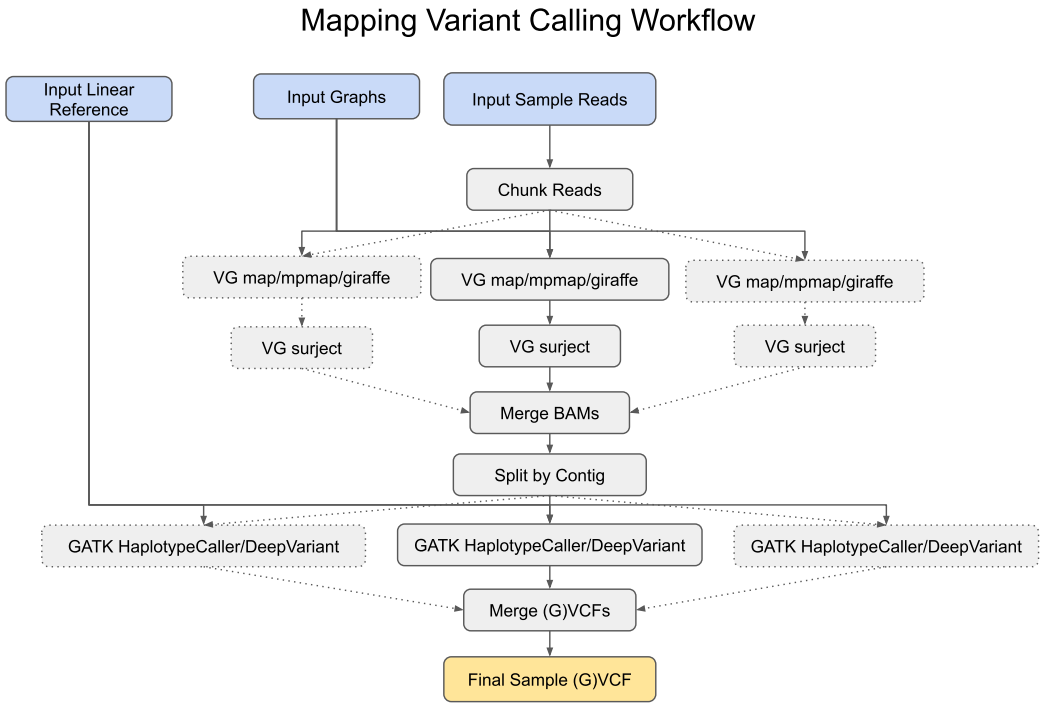


Figure 3.2: Main schematic of the vg-based alignment and variant-calling workflow. Dotted lines represent optional paths or multiplexes of command execution.

3.5 Variant Calling Evaluation

Variant calling is another domain of evaluation which focuses on the downstream effects of mapping and the quality of a variant caller. This includes looking at variant calling qualities and concordance with truth-sets.

Variant calling qualities concerns the quality of the mapping of the pileup of the sequences at a site, the quality of the bases sequenced for each read at a site, and the allele skew of the coverage of sequences at a site. In general, greater variant calling quality scores indicate higher quality in the calling algorithm and the quality of the mapping of reads at a site.

Truth-sets are used to analyze how well the called variants from a test set match the variants in a well-sequenced and highly-validated truth set. The current best practice for this analysis is based on comparison to varian sets as determined by the Genome In A Bottle Consortium (GIAB) who not only provide a continuously updated truth set but also a set of pre-determined genomic regions for investigation. These regions range from high confidence regions to the harder regions of segmental duplications, polynucleotide repeats, short tandem repeats, and difficult polymerase-chain-reaction primer sequences. Their truth set is curated using various sequencing platforms ranging from the common short-read sequencers of Illumina to the long read sequencers of PacBio, 10X Genomics, NanoPore and the high-fidelity Circular Consensus Sequence reads (CCS/HI-FI) [50, 52, 120, 116]. The PacBio-developed CCS technology is of particular interest for sequencing the difficult regions of the genome as they are one of the long-read technologies that have the lowest base error rate. This is mainly due to their method of using highly redundant sequencing of circularized reads [120].

Some of these regions came out of recent gene duplication events in genes that accumulated mutations and were responsible for many of the segregating traits of the human species from their most recent common ancestor. These are genes that have high similarity or homology and are a focus in a lot of recent NGS pipeline performance analysis due to the difficulty of mapping to these regions [69].

3.6 VG Toolkit Results

In 2018 our group published in Nature Biotechnology the initial release of the Variation Graph Toolkit and found that for variants that are unique to the individual that the graph alignment process produced more accurate alignments than the leading linear-based approaches [44]. However, though mapping performance was of high quality, there were a few major issues that were left unfixed. One critical bug was related to how graph-based alignments were projected from graph-space to linear-space which is a requirement of linear-based variant callers like GATK HaplotypeCaller or Google's DeepVariant. The other critical issue concerned the construction of a sufficiently-performing graph-reference that was based on the latest GRCh38 linear reference and population variant dataset.

The major bug that went undiscovered since the publication of the VG Toolkit was that of how reads were projected into linear-space. It was discovered that the base-quality strings for the reads that were mapped to the opposite strand were not correctly reversed in the BAM file. Fixing this error resolved the false-positive issues that impacted the overall variant calling accuracy when genotyping projected pangenome alignments. Figure 3.3 below illustrates the

performance gain when correcting this error.

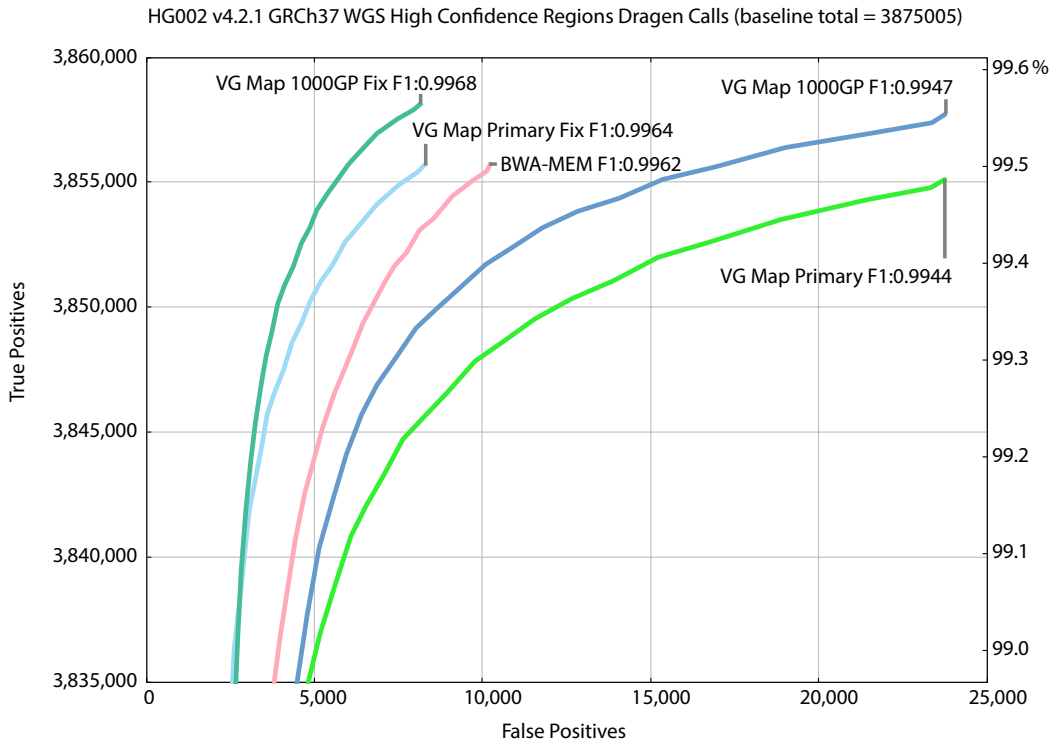


Figure 3.3: ROC curves of variant calling performance of various VG and BWA-MEM alignments with respect to HG002 GIAB v4.2.1 GRCh37 truth variant call sets. `BWA-MEM` represents reads aligned to the hs37d5 cannonical reference with BWA-MEM version 0.7.17-r1188. `VG Map Primary` represents reads aligned to a VG graph reference that is only constructed using the linear hs37d5 cannonical reference sequence and projected to linear reference space using VG version `1.24.0`. `VG Map 1000GP` represents reads aligned to a VG graph reference that is constructed using hs37d5 reference sequence and the 1000GP Phase 3 GRCh37 haplotype set and projected to linear reference space using VG version `1.24.0`. `VG Map Primary Fix` and `VG Map 1000GP Fix` represents reads aligned using the same method as `VG Map Primary` and `VG Map 1000GP`, respectively, except the projection step used VG version `1.28.0` which implements the reverse base-quality correction. All mapped reads were called using Illuminas Dragen version 3.7.5 genotyper.

In the VG paper we analyzed how various configurations of alleles that make up the pangenome reference affect mapping performance. We found that not only was the mapping

performance better for variants not found in the linear reference, which are the variants of analytical interest, but that mapping bias was reduced for larger structural variants in these regions. We also looked at the performance of graph references that contain varying degrees of allele frequencies and found that when filtering out rare variants that overall mapping accuracy is improved across-the-board in all regions examined. Though we found good performance with the pangenome graph that we used, the sequence and variant sets were based off of GRCh37 coordinates and not the improved representation of GRCh38 reference contigs.

During development of a new and faster mapper that would soon be named VG Giraffe, we were working on a GRCh38-based pangenome reference that would update the output of our new mapper to the more modern sequence-coordinate system that would go on to dominate much of what the bioinformatics and genomics field would rely on. For this task there were a few things we looked into. The 1000 Genomes Project had produced a few versions of phased population variant datasets that each had their performance issues. The first is an older variant dataset that was converted directly from their old GRCh37 variant set, termed liftover-GRCh38, as sourced from https://cgl.gi.ucsc.edu/data/giraffe/construction/. A more recent dataset from 2019 which is produced from directly mapping and calling variants against GRCh38 and restricted to biallelic SNVs and INDELs, termed snpindel-GRCh38, as sourced from http://ftp.1000genomes.ebi.ac.uk/vol1/ftp/data_collections/1000_genomes_project/release/20190312_biallelic_SNV_and_INDEL/. And the most recent re-sequencing project from the New York Genome Center which is using higher coverage Illumina NoaSeq data, termed nygc-GRCH38, as sourced from http://ftp.1000genomes.ebi.ac.uk/vol1/ftp/data_collections/1000G_2504_high_coverage/working/20201028_3

202_phased/.

Each graph constructed by a different variant dataset produced varying results when calling variants using different genotyping platforms. Tables 3.1 and 3.2 show the performance of mapping GIAB HG002 and HG003 sample sequences against various pangenome references while using the same mapper and variant-caller. Illumina's Dragen platform version 3.7.5 is used as a baseline comparison of a linear-based aligner and variant-caller, termed DRAGEN-GRCh38[51]. Ultimately, the liftover-GRCh38 pangenome reference was chosen based on a combination of sensitivity and overall F1 accuracy statistics. The pangenome reference had the best average performance in sensitivity across the examined benchmark samples without sacrificing too much precision relative to the other mapping methods. liftover-GRCh38 was also the most conservative choice as it was based off of the well sequenced and phased 1000 Genomes Phase 3 project.

Pipeline	Var Type	TP	FN	FP	Recall	Precision	F1
DRAGEN-GRCh38	INDELS	522,615	2,854	**2,475**	0.994569	**0.995481**	0.995024
	SNPS	3,344,965	20,162	**14,870**	0.994009	**0.995575**	0.994791
liftover-GRCh38	INDELS	522,882	2,587	2,718	0.995077	0.995041	**0.995059**
	SNPS	**3,349,000**	**16,127**	17,520	**0.995208**	0.994797	0.995002
snpindel-GRCh38	INDELS	522,730	2,739	2,649	0.994788	0.995165	0.994976
	SNPS	3,347,781	17,346	16,121	0.994845	0.995209	**0.995027**
nygc-GRCh38	INDELS	**522,917**	**2,552**	2,758	**0.995143**	0.994969	0.995056
	SNPS	3,348,523	16,604	16,884	0.995066	0.994984	0.995025

Table 3.1: Hap.py performance of linear and graph-based pipelines against grch38-based references using 150bp paired-end reads with respect to HG002 GIAB v4.2.1 truth variant call sets in high confidence regions. All mapped reads were called using Illuminas Dragen v3.7.5 genotyper. Best values in each column are highlighted in bold text.

Pipeline	Var Type	TP	FN	FP	Recall	Precision	F1
DRAGEN-GRCh38	INDELS	501,769	2,732	**2,646**	0.994585	**0.994969**	0.994777
	SNPS	3,307,236	20,260	**16,084**	0.993911	**0.995161**	**0.994536**
liftover-GRCh38	INDELS	**502,055**	**2,446**	2,922	**0.995152**	0.994450	**0.994801**
	SNPS	**3,311,458**	**16,038**	22,400	**0.995180**	0.993283	0.994231
snpindel-GRCh38	INDELS	501,881	2,620	2,903	0.994807	0.994484	0.994645
	SNPS	3,310,058	17,438	21,429	0.994759	0.993569	0.994164
nygc-GRCh38	INDELS	502,026	2,475	2,943	0.995094	0.994410	0.994752
	SNPS	3,311,110	16,386	21,621	0.995076	0.993514	0.994294

Table 3.2: Hap.py performance of linear and graph-based pipelines against grch38-based references using 150bp paired-end reads with respect to HG003 GIAB v4.2.1 truth variant call sets in high confidence regions. All mapped reads were called using Illuminas Dragen v3.7.5 genotyper. Best values in each column are highlighted in bold text.

Although liftover-GRCh38 had good performing results, there were still some optimizations that were made to that pangenome reference. We further examined various stratifications of genomic regions, as defined by the GIAB [58]. These regions included the Major Histocompatibility Complex (MHC) which is known for maintaining a high density of variation, 1000GP variant regions excluded from the GIAB sample (1000GP-excluded), low mappability regions that are made up of duplicated and paralogous sequence, and difficult to sequence regions that are made up of low sequence variability. We found that the largest source of precision error for liftover-GRCh38 was contributed by segmental duplications that were larger than 10 kilobases in length. Table 3.3 shows that filtering out these variants from the liftover dataset prior to pangenome construction, termed liftover-GRCh38-nonsegdup, produced much better performing SNP accuracy across nearly all regions. The linear graph reference constructed using just the GRCh38 sequence was used as the point of comparison in this analysis, termed

primary-GRCh38.

Pipeline	Var Type	TP	FN	FP	Recall	Precision	F1
primary-GRCh38	INDELS	522,657	2,812	2,608	0.994649	0.995239	0.994944
	SNPS	3,347,097	18,030	15,672	0.994642	0.995341	0.994991
liftover-GRCh38	INDELS	**522,887**	**2,582**	2,774	**0.995086**	0.994940	**0.995013**
	SNPS	**3,349,241**	**15,886**	18,073	**0.995279**	0.994634	0.994957
liftover-GRCh38-nonsegdup	INDELS	522,439	3,030	**2,582**	0.994234	**0.995284**	0.994759
	SNPS	3,349,015	16,112	**15,214**	0.995212	**0.995479**	**0.995345**

(a) All Regions

Pipeline	Var Type	TP	FN	FP	Recall	Precision	F1
primary-GRCh38	INDELS	1,624	53	39	0.968396	0.978793	0.973567
	SNPS	19,745	432	262	0.978589	0.986798	0.982677
liftover-GRCh38	INDELS	**1,649**	**28**	**23**	**0.983304**	**0.987674**	**0.985484**
	SNPS	**20,015**	**162**	**80**	**0.991971**	**0.995981**	**0.993972**
liftover-GRCh38-nonsegdup	INDELS	1,646	31	29	0.981515	0.984509	0.983009
	SNPS	19,998	179	164	0.991129	0.991787	0.991458

(b) MHC Regions

Pipeline	Var Type	TP	FN	FP	Recall	Precision	F1
primary-GRCh38	INDELS	338,507	2,244	2,101	0.993415	0.994085	**0.993750**
	SNPS	2,163,130	17,419	14,905	0.992012	0.993159	0.992585
liftover-GRCh38	INDELS	**338,682**	**2,069**	2,294	**0.993928**	0.993549	0.993738
	SNPS	**2,165,083**	**15,466**	17,418	**0.992907**	0.992022	0.992464
liftover-GRCh38-nonsegdup	INDELS	338,362	2,389	**2,084**	0.992989	**0.994129**	0.993559
	SNPS	2,164,822	15,727	**14,513**	0.992788	**0.993343**	**0.993065**

(c) Low Mappability and Segdup Regions

Pipeline	Var Type	TP	FN	FP	Recall	Precision	F1
primary-GRCh38	INDELS	366,438	2,704	2,361	0.992675	0.993966	0.993320
	SNPS	610,666	17,297	14,133	0.972455	0.977414	0.974928
liftover-GRCh38	INDELS	**366,636**	**2,506**	2,520	**0.993211**	0.993566	**0.993389**
	SNPS	**612,700**	**15,263**	16,592	**0.975694**	0.973672	0.974682
liftover-GRCh38-nonsegdup	INDELS	366,178	2,964	**2,339**	0.991971	**0.994018**	0.992993
	SNPS	612,464	15,499	**13,692**	0.975319	**0.978165**	**0.976740**

(d) Difficult Regions

Table 3.3: **Genotyping evaluation with Hap.py in HG002 against different stratifications of the high-confident regions of the liftover-GRCh38 graph** (A) SNP and INDEL accuracy within all confident regions. (B) Accuracy in MHC regions. (C) Accuracy in repetitive and segmental duplication regions. (D) Accuracy in regions with low sequence variability. All mapped reads were called using Illuminas Dragen v3.7.5 genotyper. Best values in each column are highlighted in bold text.

3.7 VG Giraffe Results

3.7.1 Giraffe genotyping outperforms best practices

In 2021 our group published our work on developing and evaluating the performance of a faster pangenome mapper `VG Giraffe` [102]. In this study we applied the pangenome graph reference based on the liftover-GRCh38 dataset with large segmental duplications removed as our basis for performance evaluation.

We used Illumina's Dragen platform[51] to genotype SNV and short indels using Giraffe mappings projected onto the linear reference assembly. We compared them to results using competing graph and linear reference mappers (see 3.7.2.1). No training or calibration was performed for any of the mappings other than those performed by default by Dragen itself. We evaluated the calls using the Genome In a Bottle (GIAB) v4.2.1 HG002 high confidence variant calling benchmark[117].

Out of the examined pipelines, Giraffe mappings to the 1000GP graph produce the highest overall F1 score (harmonic mean of precision and recall) at 0.9953 (Figure 3.4B and Tables 3.4 and 3.5). Structural variants were also more accurately detected when using the `VG Giraffe mapper` with respect to using `VG Map` (Figure 3.4C).

Similar but uniformly higher results were found with higher coverage, 250bp reads (Figure A.1 and Tables A.1 and A.2). Although one would expect longer reads and higher coverage to produce better variant calls, all else being equal, Giraffe with the 150bp read set has a slightly higher F1 score (0.9953) than BWA-MEM with the higher coverage 250bp read set (0.9952). Further restricting comparison only to confident regions that overlap variant calls

from the 1000GP variants used in graph construction, Giraffe has the highest F1 score at 0.9995 relative to the other methods (Figure A.2 and Table A.3). Perhaps surprisingly, Giraffe also maintains the highest F1 score (0.9528) when performing the converse analysis, restricting the comparison to only confident regions that do not overlap 1000GP variant calls (Figure A.3 and Table A.4).

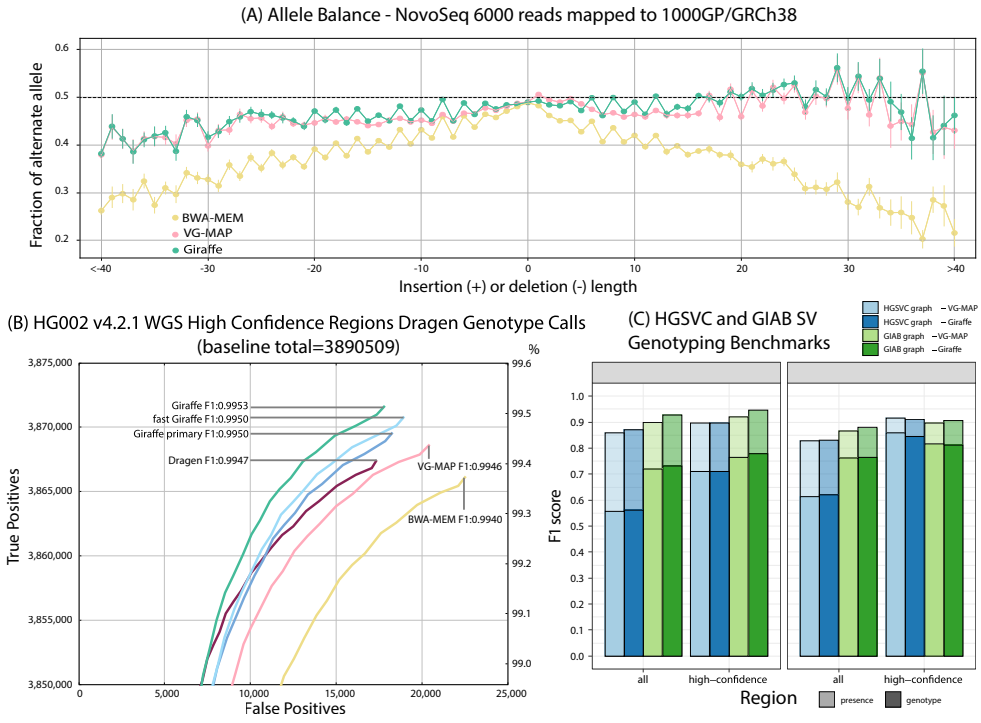


Figure 3.4: **Evaluating Giraffe for genotyping**. (A) The fraction of alternate alleles in reads detected for heterozygous variants in NA19239. Reads were mapped to the 1000GP graph with Giraffe and VG-MAP and to GRCh38 with BWA-MEM, and the fraction of reads supporting reference or alternate alleles was found for each indel length. (B) Assessing true positive and false positive genotypes made using the Dragen genotyper with mappings from Giraffe and other mappers. The line labeled Dragen represents the mapper included with the Dragen system itself. (C) Comparing Giraffe to VG-MAP for typing large insertions and deletions. "Presence" (lighter bars) evaluates the detection of SVs without regard to genotype; "genotype" (darker bars) requires the SV to be detected and its genotype to agree with the truth genotype. The y-axis shows the F1 score. For the HGSVC benchmark, we define high-confidence regions as regions not overlapping simple-repeats and segmental duplications.

Pipeline	TP	FP	FN	Precision	Sensitivity	F-measure
BWA-MEM	3,866,030	22,519	24,392	0.9942	0.9937	0.9940
DRAGEN	3,867,353	**17,369**	23,071	**0.9955**	0.9941	0.9948
VG-MAP	3,868,494	20,393	21,929	0.9948	0.9944	0.9946
Giraffe primary	3,869,525	18,308	20,909	0.9953	0.9946	0.9950
Giraffe	**3,871,501**	17,787	**18,917**	0.9954	**0.9951**	**0.9953**
fast Giraffe	3,870,737	18,964	19,681	0.9951	0.9949	0.9950

Table 3.4: VCFeval performance of linear and graph-based pipelines against grch38-based references using 150bp paired-end reads with respect to HG002 GIAB v4.2.1 truth variant call sets in high confidence regions. Best values in each column are highlighted in bold text.

Pipeline	Var Type	TP	FN	FP	Recall	Precision	F1
BWA-MEM	INDELS	522,406	3,063	2,723	0.994171	0.995028	0.994599
	SNPS	3,343,857	21,270	19,780	0.993679	0.994121	0.993900
DRAGEN	INDELS	522,618	2,851	**2,478**	0.994574	**0.995475**	0.995025
	SNPS	3,344,971	20,156	**14,868**	0.994010	**0.995576**	0.994793
VG-MAP	INDELS	522,643	2,826	2,689	0.994622	0.995092	0.994857
	SNPS	3,346,085	19,042	17,677	0.994341	0.994746	0.994544
Giraffe primary	INDELS	522,657	2,812	2,608	0.994649	0.995239	0.994944
	SNPS	3,347,097	18,030	15,672	0.994642	0.995341	0.994991
Giraffe	INDELS	**522,864**	**2,605**	2,550	**0.995043**	0.995346	**0.995194**
	SNPS	**3,348,882**	**16,245**	15,213	**0.995173**	0.995479	**0.995326**
fast Giraffe	INDELS	522,825	2,644	2,583	0.994968	0.995286	0.995127
	SNPS	3,348,156	16,971	16,358	0.994957	0.995139	0.995048

Table 3.5: Hap.py performance of linear and graph-based pipelines against grch38-based references using 150bp paired-end reads with respect to HG002 GIAB v4.2.1 truth variant call sets in high confidence regions. Best values in each column are highlighted in bold text.

DeepVariant is a highly accurate genotyping tool that requires training [86]. We trained DeepVariant version 1.1.0 to use Giraffe mappings and evaluated it on the held-out sample HG003. We compared it to the Dragen pipelines tested and DeepVariant using BWA-MEM with the BWA-MEM trained model they provide. The Giraffe-DeepVariant pipeline (F1: 0.9965) outperforms all other tested pipelines (Tables 3.6, 3.7 and Figure 3.5).

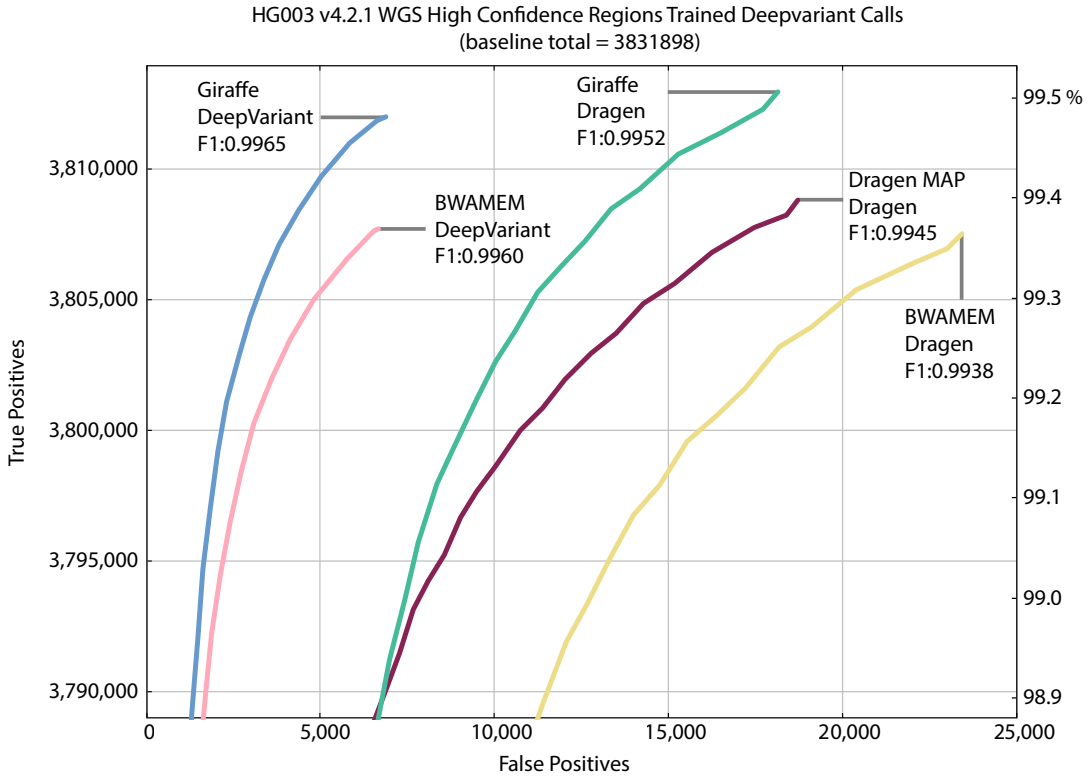


Figure 3.5: True positive and false positive genotypes made using the DeepVariant genotyper trained on alignments of 150bp paired-end reads from HG002 and HG004 GIAB samples and tested on 150bp paired-end reads from the HG003 GIAB sample and evaluated against the HG003 GIAB v4.2.1 truth variant call sets in high confidence regions. Dragen genotyper results are included for performance comparison. The ROC curve discrimination threshold is based on variant call quality.

Pipeline	TP	FP	FN	Precision	Sensitivity	F-measure
BWA-MEM + Dragen	3,807,536	23,443	24,345	0.9939	0.9936	0.9938
Giraffe + Dragen	**3,812,963**	18,143	**18,922**	0.9953	**0.9951**	0.9952
Dragen MAP + Dragen	3,808,794	18,738	23,081	0.9951	0.9940	0.9945
BWA-MEM + DeepVariant	3,808,209	**6,680**	24,173	**0.9982**	0.9937	0.9960
Giraffe + DeepVariant	3,812,716	6,889	19,898	**0.9982**	0.9948	**0.9965**

Table 3.6: RTG VCFeval performance between DeepVariant and Dragen variant callers on grch38-based linear and giraffe 1000GP mappers using 150bp paired-end reads with respect to HG003 GIAB v4.2.1 truth variant call sets in high confidence regions. Best values in each column are highlighted in bold text.

Pipeline	Var Type	TP	FN	FP	Recall	Precision	F1
BWA-MEM + Dragen	INDELS	501,574	2,927	2,789	0.994198	0.994696	0.994447
	SNPS	3,306,162	21,331	20,642	0.993589	0.993797	0.993693
Giraffe + Dragen	INDELS	**501,993**	**2,508**	2,566	**0.995029**	0.995122	0.995075
	SNPS	**3,311,179**	**16,317**	15,563	**0.995096**	0.995323	0.995210
Dragen MAP + Dragen	INDELS	501,770	2,731	2,644	0.994587	0.994972	0.994780
	SNPS	3,307,236	20,260	16,078	0.993911	0.995163	0.994537
BWA-MEM + DeepVariant	INDELS	501,433	3,068	**1,393**	0.993919	**0.997342**	0.995627
	SNPS	3,306,484	21,012	5,268	0.993685	0.998410	0.996042
Giraffe + DeepVariant	INDELS	501,919	2,582	1,851	0.994882	0.996476	**0.995678**
	SNPS	3,310,338	17,158	**5,015**	0.994844	**0.998488**	**0.996662**

Table 3.7: Hap.py performance between DeepVariant and Dragen variant callers on grch38-based linear and giraffe 1000GP mappers using 150bp paired-end reads with respect to HG003 GIAB v4.2.1 truth variant call sets in high confidence regions. Best values in each column are highlighted in bold text.

We compared the performance of Giraffe, VG-MAP, Illumina's Dragen platform, and BWA-MEM for genotyping SNVs and short indels. The design of each calling pipeline is described in 3.7.2.1 and the parameters and indexes for each experiment is described in A.5. The variants produced by each pipeline were compared against the Genome In a Bottle (GIAB) v4.2.1 HG002 high confidence variant calling benchmark[117] using the RealTimeGenomics `vcfeval` tool[22] and Illumina's `hap.py` tool[58]. This benchmark set covers 92.2% of the GRCh38 sequence.

We also evaluated a DeepVariant [86] pipeline that uses Giraffe mappings (see 3.7.2.5). Using the default DeepVariant 1.1.0 trained model, we tested genotyping the HG003 sample across the entire genome. This sample was not used in training the model.

3.7.2 Methods

3.7.2.1 Genotyping and Evaluation Methods

We mapped ∼830 million paired-end, 150-bp-long reads (Precision FDA Challenge V2 Illumina ∼35x coverage) from the HG002 sample to the 1000GP graph. We also separately evaluated mapping ∼1.08 billion 250-bp-long reads (∼40-50x coverage of Illumina HiSeq 2500) from HG002 to see if using longer reads and higher coverage would affect the results. For the genome graph mappers, we evaluated variant calling performance by using `vg surject` to produce BAM representations of our graph alignments projected onto the linear reference assembly, and then using Dragen version 3.7.5 to call variants against the `hs38d1` reference for each set of alignments. Dragen was used as the primary variant caller because Illumina, who sells it, has found it to produce robust results[51, 58, 82]. Its speed was also useful

for the purposes of rapid evaluation of whole genome alignments of real read data. No training or calibration was performed for any of the generated mappings other than those performed by default by Dragen itself.

All pipelines evaluated for short variant calling performance had the same structure. First, we mapped reads to the appropriate GRCh38-based linear or graph reference using the pipeline's mapper. Then, we genotyped the resulting alignments using Illumina's "Dragen Bio-IT Platform" product, version 3.7.5, against an index generated from the `hs38d1` human genome reference. The mappers for each pipeline evaluated were Illumina's Dragen internal aligner, BWA-MEM, VG-MAP, Giraffe, and Giraffe in fast mode. The Genome In a Bottle HG002 version 4.2.1 high-confidence VCFs were used as the truth sets for evaluating performance of variant calling[17]. The VCF was obtained from https://ftp-trace.ncbi.nlm.nih.gov/giab/ftp/release/AshkenazimTrio/HG002_NA24385_son/NISTv4.2.1/GRCh38/HG002_GRCh38_1_22_v4.2.1_benchmark.vcf.gz and the high confidence regions evaluated were based on the BED files obtained from https://ftp-trace.ncbi.nlm.nih.gov/giab/ftp/release/AshkenazimTrio/HG002_NA24385_son/NISTv4.2.1/GRCh38/HG002_GRCh38_1_22_v4.2.1_benchmark_noinconsistent.bed. We used RTG's `vcfeval` [22] and Illuminas `hap.py` https://github.com/Illumina/hap.py to evaluate variant calling concordance with truth sets. All pipelines used the computational resources of the NIH HPC Biowulf cluster (http://hpc.nih.gov).

3.7.2.2 Graph construction

For the variant calling experiments, graph references, and their indexes for running vg mappers, were constructed using different versions of vg ranging from `1.20.0` to `1.31.0`, orchestrated by `toil-vg construct`.

The graph references for GRCh38-based experiments were constructed using vg version `v1.31.0` as packaged in container `quay.io/vgteam/vg:v1.31.0`, using the cannonical `hs38d1` FASTA reference file as sourced from https://storage.googleapis.com/cmarkell-vg-wdl-dev/giraffe_manuscript_data/genome_references/linear_references/GCA_000001405.15_GRCh38_no_alt_analysis_set_plus_GCA_000786075.2_hs38d1_genomic.fna and the 1000 Genomes Project Phase 3 phased VCFs that were lifted over from old GRCh37-based joint-called datasets as sourced from https://storage.googleapis.com/cmarkell-vg-wdl-dev/giraffe_manuscript_data/genome_references/graph_references/1000gp_data/ALL.chr*_GRCh38.genotypes.20170504.rename.vcf.gz. *Primary* graphs and indexes were constructed using just the linear reference FASTA with vg version `v1.26.0-180-gdc119fa04` as packaged in container `quay.io/vgteam/vg:ci-2284-dc119fa046aa7131a1a8e026be36da2d79bc2f22`. *1000GP* graphs and indexes were constructing using both the FASTA and the VCFs after filtering out variants belonging to segmental duplication regions of greater than 10 kilobases in length as defined in the bed file https://ftp-trace.ncbi.nlm.nih.gov/giab/ftp/release/genome-stratifications/v2.0/GRCh38/SegmentalDuplications/GRCh38_segdups_gt10kb.bed.gz with vg version `v1.31.0` as packaged in container

quay.io/vgteam/vg:ci-2890-655a9622c3d60e87f14b88d943fbd8554214a975.

3.7.2.3 Read sets

Two different read sets were used to evaluate performance. One was the 150 bp paired-end FASTQ set from sample HG002 as obtained from the FDA Precision Challenge dataset. These reads are available as paired FASTQs at https://storage.googleapis.com/cmarkell-vg-wdl-dev/test_input_reads/HG002.NovaSeq.pcr-free.35x.R1.fastq.gz and https://storage.googleapis.com/cmarkell-vg-wdl-dev/test_input_reads/HG002.NovaSeq.pcr-free.35x.R2.fastq.gz. The other read set consisted of 250 bp paired-end FASTQ reads from sample HG002, obtained from the GIAB NovoAligned BAM at ftp://ftp-trace.ncbi.nlm.nih.gov/giab/ftp/data/AshkenazimTrio/HG002_NA24385_son/NIST_Illumina_2x250~bps/novoalign_bams/HG002.hs37d5.2x250.bam. These reads in paired FASTQ format are available at https://storage.googleapis.com/cmarkell-vg-wdl-dev/test_input_reads/HG002_read_pair_1.fq.gz and https://storage.googleapis.com/cmarkell-vg-wdl-dev/test_input_reads/HG002_read_pair_2.fq.gz.

3.7.2.4 Read mapping

The HISAT2 mapping runs were done using version `2.1.0`. Only default settings, with the addition of the `--no-spliced-alignment` flag, were used during HISAT2 execution.

One pipeline used Illumina's Dragen module for both mapping and alignment; both steps were run against the same index derived from the `hs38d1` human genome reference.

The BWA-MEM mapping runs were done using version `0.7.17-r1188` against the

`hs38d1` human genome reference.

The vg alignment runs used different graph indexes depending on the mapper used (VG-MAP or Giraffe). GRCh38-based experiment run details are given in Supplementary Table A.5. All runs used the `toil-vg map` workflow for the alignment procedure and ABRA2[76] for indel realignment after alignment. GATK's Picard tool [3] was used to reorder aligned BAM files for contig consistency which is required by the Dragen variant calling algorithm.

3.7.2.5 DeepVariant calling

All pipelines for the DeepVariant experiment evaluation used BWA-MEM version `0.7.17-r1188` against the `hs38d1` human genome reference or VG Giraffe against the same 1000GP graph as described in 3.7.2.2. Alignments were then indel realigned using ABRA2[76]. The DeepVariant v1.1.0 code base was used for all experiments.

DeepVariant learns to call variants by training on input data from gold standard samples HG001-HG007. In all DeepVariant models, the HG003 sample is fully withheld from training, allowing a full genome sample to be evaluated. Because the DeepVariant production models for Illumina are trained on samples mapped with BWA, DeepVariant likely optimizes for the mapping accuracy and quality profile of BWA. We used DeepVariant v1.1.0 production model for the BWA-MEM experiments and a custom model for the GIRAFFE alignments that were trained using the same methods that were used to train the DeepVariant model for BWA-MEM alignments. The custom model data can be found here `https://storage.googleapis.com/cmarkell-vg-wdl-dev/giraffe_manuscript_data/deepvariant_models/model.ckpt-25600*`.

3.8 Code Availability and Data Access

An overview of the data generated for this paper, and key input data to reproduce the analyses, is available at https://cglgenomics.ucsc.edu/giraffe-data/. The dataset is available via IPFS at https://ipfs.io/ipfs/QmceMYoTf1UiMED6c9WdsdPfJ5zf4NuXLbL8oReJT6qYc6.

Archived copies of the code and final re-usable work products have been deposited in Zenodo as DOI 10.5281/zenodo.4774363, referenced here as [103]. This archive also includes `vg`, `toil-vg`, and `toil` source code and Docker containers used in this work, as well as the `giraffe-sv-paper` orchestration scripts. "Final" versions of vg and toil-vg, including all features needed to reproduce this work, are 9907ab2 for vg and 99101f2 for toil-vg.

The latest version of the vg toolkit, including the Giraffe mapper, is customarily distributed at https://github.com/vgteam/vg. The scripts used for the analysis presented in this study were developed at https://github.com/vgteam/giraffe-sv-paper, a git bundle of which is archived in Zenodo [103].

Data used in the Giraffe read mapping experiments, including the 1000GP, HGSVC, and yeast target graphs, the linear control graphs, the graphs used to simulate reads, and the simulated reads themselves, can be found at https://cgl.gi.ucsc.edu/data/giraffe/mapping/.

The SV pangenomes and SV catalogs annotated with allele frequencies are hosted at https://cgl.gi.ucsc.edu/data/giraffe/calling/ and archived in [103]. This repository also includes SVs with strong inter-super-population frequency patterns, SV-eQTLs,

and SVs overlapping protein-coding genes.

To build the 1000GP and HGSVC graphs, we used the GRCh38 no-alt analysis set (accession GCA_000001405.15), and the `hs38d1` decoy sequences (accession GCA_000786075.2), both available from NCBI, in addition to the variant call files distributed by the respective projects.

3.9 Acknowledgements

Research reported in this chapter was supported by the National Human Genome Research Institute of the National Institutes of Health under Award Numbers U41HG010972, 1R01HG008742, U01HG010961, R01HG009737 and 2U41HG007234. Research reported in this chapter was also supported by the National Heart, Lung, And Blood Institute of the National Institutes of Health under Award Number U01HL137183 and the BioData Catalyst Fellows Program of the National Institutes of Health through the University of North Carolina at Chapel Hill, under Award Number 1 OT3 HL147154. The high coverage sequencing data for the 1000 Genomes Project were generated at the New York Genome Center with funds provided by NHGRI Grant 3UM1HG008901-03S1. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

Part III

Pedigree-Backed Genome Pipeline

Chapter 4

Complete Pedigree-Based Graph Workflow for Rare Candidate Variant Analysis

4.1 Preamble

Methods that use a linear genome reference for genome sequencing data analysis are reference biased. In the field of clinical genetics for rare diseases, a resulting reduction in genotyping accuracy in some regions has likely prevented the resolution of some cases. Pangenome graphs embed population variation into a reference structure. While pangenome graphs have helped to reduce reference mapping bias, further performance improvements are possible. We introduce VG-Pedigree, a pedigree-aware workflow based on the pangenome-mapping tool of Giraffe [102] and the variant-calling tool *DeepTrio* [56] using a specially-trained model for Giraffe-based alignments. We demonstrate mapping and variant calling improvements in both single-nucleotide variants (SNVs) and insertion and deletion (INDEL) variants over those pro-

duced by alignments created using BWA-MEM to a linear-reference and Giraffe mapping to a pangenome graph containing data from the 1000 Genomes Project. We have also adapted and upgraded the deleterious-variant (DV) detecting methods and programs of Gu et al. into a streamlined workflow [45]. We used these workflows in combination to detect small lists of candidate DVs among 15 family quartets and quintets of the Undiagnosed Diseases Program (UDP). All candidate DVs that were previously diagnosed using the Mendelian models covered by the previously published Gu et al. methods were recapitulated by these workflows. The results of these experiments indicate a slightly greater absolute count of DVs are detected in the proband population than in their matched unaffected siblings.

4.2 Introduction

Recent advances in genome sequencing technology are improving the accuracy of detecting genetic variants [120]. However, the use of a single genome reference for read alignment and variant calling still presents a problem. A sequence mapping algorithm best aligns sequences to a reference when those sequences are present in the reference. Where a sample's genome deviates significantly enough from the reference, reads will fail to map properly [97]. This reference bias can be reduced using pangenome graphs. Pangenome Graphs represent multiple genomes as a series of variants [44]. These graphs are further enhanced by incorporating haplotype information that is available in phased genotype datasets. This haplotype information is embedded in a haplotype index [102]. In previous work, we have found that mapping error, in both simulation and real-data experiments, is reduced by using population variant data in

pangenome graph references [44, 102].

Parent-child trios provide evidence of sequence transmission between generations. This helps to identify which variants in the child occurred as de-novo mutations, since these variants will generally be absent in the parents. This information also helps to determine phasing orientation of heterozygotes in the child which can aid in detecting compound-heterozygous candidate DVs. In typical clinical diagnostics, in particular for the case of rare diseases, parental genomes are sequenced to help improve the chances of successful clinical diagnosis of a proband [20].

The Undiagnosed Disease Program (UDP) of the National Human Genome Research Institute (NHGRI) is charged with diagnosing previously undiagnosed individuals and discovering new variants of clinical significance [38, 36, 37, 39, 106]. In 2009 the UDP started examining cases that have remained undiagnosed after previous exhaustive clinical examination. One part of their process involved sequencing the genomes of patients, including some that included parents, an affected proband and one or more unaffected siblings. Since the beginning of the UDP, they have seen more than 500 different disorders and achieved a diagnostic success rate of over 30 percent, including the discovery of new disorders [45]. Most of the pediatric cases examined by the UDP over the past 10 years have already had negative diagnostic results from clinical exomes. The UDP applies further technologies, including whole genome sequencing, RNA sequencing, and SNP-chip analysis to more completely explore non-exonic and intergenic regions in an attempt to solve negative exome cases [45]. One of the more difficult tasks of gene discovery is the detection of variants in highly polymorphic, repetitive, and incompletely-represented regions of the genome, exactly where pangenome graphs can potentially extend

accuracy and precision.

We first describe VG-Pedigree, a software workflow for mapping and variant calling next generation sequencing data. The workflow leverages pedigrees in genome graphs, and uses machine-learning for variant calling. Intermediary results from VG-Pedigree are subsequently used to identify candidate deleterious variants by using a significantly upgraded, fully automated single stage implementation of the UDP candidate analysis workflow [45]. These upgrades include better software portability and usability, change to the GRCh38 reference, use of better population datasets and newer deleterious predictors than those used in the previous version. The final upgrade was the addition of a new software module to detect and quantify large scale mosaicism.

This workflow analyzed 15 UDN quartet+ pedigrees containing one affected child (Proband), one or two unaffected siblings, and two unaffected parents to demonstrate its capabilities. Of the probands that had a known diagnosis, the corresponding DVs detected based on the Mendelian models covered by the previously published candidate analysis workflow were recapitulated by this workflow. This unified workflow was designed to run from machine output FASTA sequence data to a final short candidate list, but it is modular. The first part of VG-Pedigree produces an intermediate set of BAM files and a jointly called VCF file. The candidate analysis in the second part could be run using any set of genome BAM formated files plus a joint-called VCF formatted pedigree dataset.

4.3 Results

4.3.1 Overview of VG-Pedigree

VG-Pedigree goes through a number of stages before final variant calling (Fig. 4.1A). First, the set of short reads in the parent-parent-child trio of the pedigree are mapped to a pangenome graph reference based on the 1000 Genomes Project dataset, termed 1000GP, using VG Giraffe, and variants are then called using *DeepTrio* (Fig. 4.1B) [9, 102]. Next, variants in 1000 Genomes Project haplotypes that appear missing in the DeepTrio-called variants are imputed. The purpose of this is to fill in common variants that were possibly missed by the variant callers in order to facilitate the phasing of more complete haploblocks. The resulting variant file is phased using both alignment and pedigree information (Fig. 4.1C). A parental graph reference is then constructed using only the parental genotypes from the joint-called VCF file (Fig. 4.1D). A haplotype index of this graph reference for VG Giraffe is generated from the phased genotypes of the parental samples. Once this graph is constructed, the proband and siblings' reads are re-mapped to this new parental graph reference and variants are re-called using the new mappings (Fig. 4.1B). Finally, the newly-called variants of the child and sibling samples are joint-called with the old parental variants to form the final joint-called pedigree VCF.

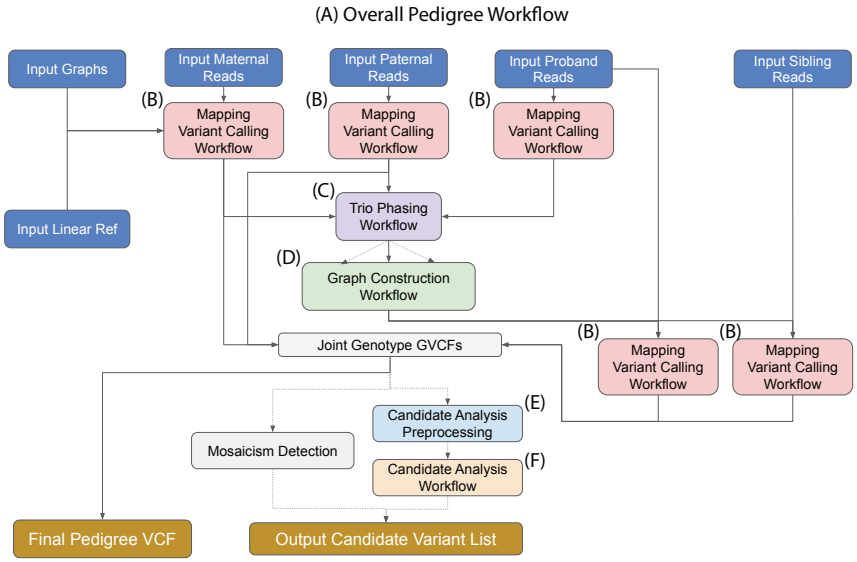


Figure 4.1: Toil-VG Pedigree workflow. Dotted lines indicate optional pathways in the workflow. (A) Overall workflow diagram. (B) Single sample alignment and variant-calling workflow. (C) Trio joint-genotyping and phasing workflow. (D) Parental graph construction workflow. (E) Workflow for preprocessing and annotation of pedigree variants required for candidate analysis. (F) The candidate analysis workflow.

The candidate analysis workflow takes as input the set of alignments and variant calls from VG-Pedigree and outputs a final set of candidate DVs for the proband. This is done through a series of filters and annotations. First, SnpEff is used to annotate the type and function of variants within the joint-called pedigree VCF file [19]. The deleteriousness of these variants is predicted using the Combined Annotation Dependent Depletion (CADD) software tool [90] (Fig. 4.1E). Next, a series of filtration and analysis methods are applied to the annotated variants and the workflow outputs a set of candidate DVs for the proband (Fig. 4.1F). The methods

applied in the candidate analysis workflow are an implementation of the methods described in the Gu et al. study [45]. In this paper, we present enhancements to the methods and software of the candidate analysis workflow. An additional module of the analysis workflow has also been developed which automatically detects the presence and type of mosaicism in the designated proband. These methods and improvements together provide a more complete and accurate dataset from which to discover rare variants that are causal to genetic diseases over the previous iteration.

We evaluated performance of this workflow based on four main metrics. First, we evaluated the ability of the workflow to accurately align reads to the correct position in a genome. Second, we assessed the accuracy of variant calls based on those alignments. Thirdly, we looked at the ability of the analysis workflow to capture DVs in the proband population versus the unaffected sibling population. Finally, we examined the runtime and costs of running this workflow using a commercial cloud environment.

4.3.2 Mapping Evaluation

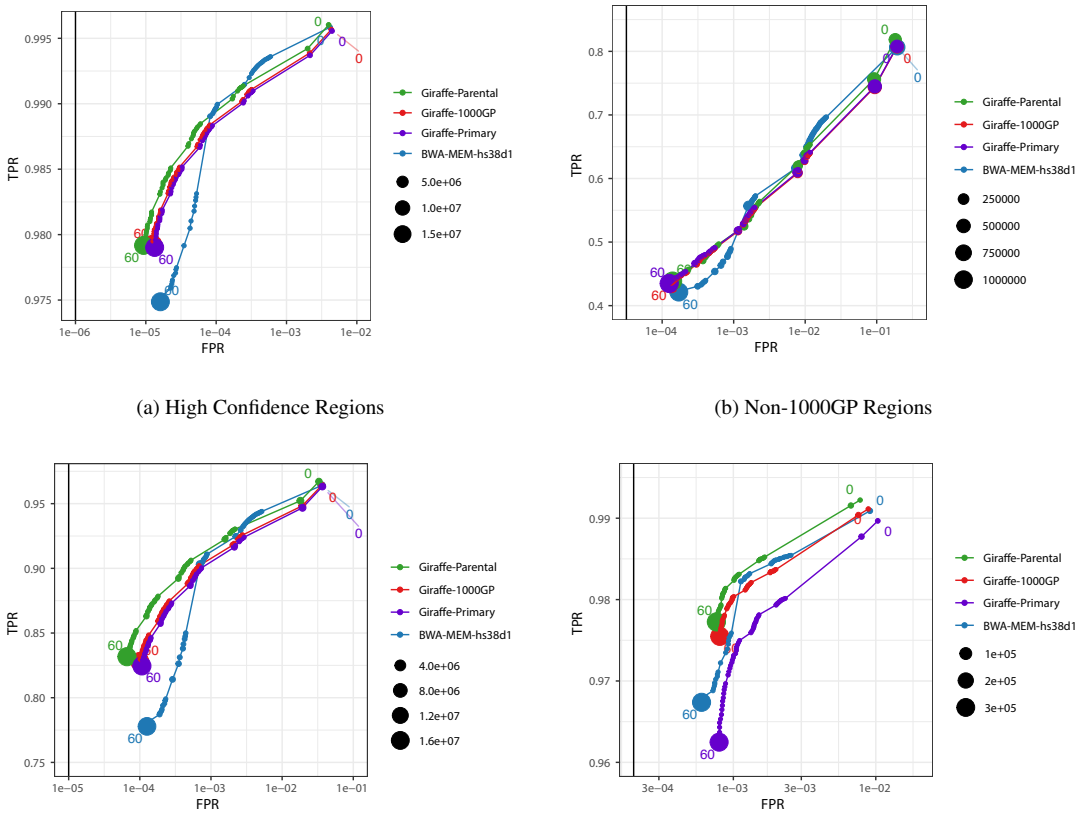
Mapping was evaluated with both simulated and real sequencing data. The former considers measures of mapping reads with a known position. This was done by simulating reads from haplotypes whose corresponding path locations in the graph are known, so that we could identify when a read was mapped to the correct location on the graph. We simulated reads by first constructing sample graphs using benchmark sample variation data and then generated paired-ended reads using error models and pair distance distributions based on real-read data. We also made sure to only simulate reads from benchmark samples that were

not contained within the pangenome references used in the graph mapping methods. During evaluation, simulated reads mapped to the linear references were injected to graph reference space for comparison with graph mappers (see supplementary methods 4.5.3). Figure 4.2 illustrates the performance of 10 million read pairs that are simulated from the Genome-in-a-Bottle (GIAB) HG002 version 4.2.1 high confidence variant sets [58]. We also examined stratified performance across regions of interest using 100 million reads simulated from the GIAB high confidence regions. These regions were all defined by GIAB: [58] low complexity regions that comprise regions of low sequence variability, low mappability regions that are made up of duplicated and paralogous sequence, the Major Histocompatibility Complex (MHC) which is known for maintaining a high density of variation, 1000GP variant regions excluded from the GIAB sample (1000GP-excluded), and, specifically for HG002, the complex medically relevant genes (CMRG) included in a study by Wagner et al. [116].

All conditions evaluated consist of the combination of a mapper and a reference (see supplementary methods 4.5.4). The *Giraffe-Parent* condition used VG Giraffe [102] to align reads to the parental graph reference as produced by the workflow up to graph construction (Fig. 4.1D). The *Giraffe-1000GP* condition used VG Giraffe to align reads to the pangenome reference. The *Giraffe-Primary* condition used VG Giraffe to align reads to a linear graph reference as produced using only the `hs38d1` reference with no variation. And the *BWA-MEM-hs38d1* condition used BWA-MEM [64] to align reads to the `hs38d1` human reference genome.

Figure 4.2 shows the receiver-operator-curves (ROC) of each tested mapper in all confident regions, 1000GP-excluded regions, low mappability regions, and MHC regions. The curves are stratified by mapping quality (MAPQ). In each evaluated region, *Giraffe-Parent* pro-

duced the highest F1, both for reads with MAPQ60 and across all reads. When looking at 1000GP-excluded variants within stratified regions, *Giraffe-Parent* produced the highest total F1 across low complexity regions (Fig. B.1), low mappability regions (Fig. B.2), MHC regions (Fig. B.3), and CMRG regions (Fig. B.4).



(a) High Confidence Regions

(b) Non-1000GP Regions

(c) Low Mappability Regions

(d) MHC Regions

Figure 4.2: Mapping performance of 100 million read pairs simulated from HG002 high confident datasets. Four different alignments are compared across four different regions and ROC curves are plotted with a log-scaled false positive rate on the x-axis and a linear-scaled true positive rate on the y-axis with the mapping quality as the discriminating factor. Green curves represent graph alignments against the parental graph reference constructed from HG003 and HG004 illumina read graph alignments. Red curves represent alignments against the 1000GP graph reference. Purple curves represent alignments to the primary GRCh38 linear graph reference. Blue curves represent linear alignments against the `hs38d1` reference using BWA-MEM. (A) Alignments in GIAB v4.2.1 confident regions (from 1 million simulated read set). (B) Alignments in non-1000GP confident regions (from 1 million simulated Illumina read set). (C) Alignments in GIAB v4.2.1 low mappability regions (from 100 million simulated Illumina read set). (D) Alignments in GIAB v4.2.1 MHC regions (from 100 million simulated Illumina read set).

For all GIAB high confidence regions, *Giraffe-Parent* gave the most accurate alignments relative to the other examined mappers. *Giraffe-Parent* also achieved the highest total of correctly mapped reads in all but the CMRG regions, the highest total of reads mapped at MAPQ60 in low mappability MHC and CMRG regions, and the highest average percent identity between aligned reads and the reference sequence across all regions (Table B.1). In the high confident 1000GP-excluded regions of the HG002 sample, *Giraffe-Parent* achieved the highest proportion of correctly mapped reads, MAPQ60 reads, and average sequence identity (Table B.2). *Giraffe-Parent* also produced the highest proportions of perfectly-aligned and gaplessly-aligned reads, and the lowest proportion of soft-clipping reads across all examined confident (Table B.3) and 1000GP-excluded regions (Table B.4).

4.3.3 Variant Calling Evaluation

In addition to examining the mapping performance of the workflow we measured the accuracy of variants called in each workflow. Here we use the version 4.2.1 release of the HG001, HG002 and HG005 truth-set benchmarks as published by GIAB [58, 117]. Real-TimeGenomics' `vcfeval` tool [22] and Illumina's `hap.py` haplotype aware variant comparison tool [58] were used when comparing the results of variants called using alignments of real reads to various combinations of mappers and references. The mappers and references used include VG Giraffe against the parental graph (*Giraffe-Parent*), which is the method used by VG-Pedigree, and, for comparison, VG Giraffe against the 1000GP graph (termed *Giraffe-1000GP*), `BWA-MEM` against the linear `hs38d1` reference (*BWA-MEM-hs38d1*), and Illumina's `Dragen` platform version 3.7.5 [51, 58, 82] against the linear `hs38d1` reference (*Dragen-hs38d1*)

(see supplementary methods 4.5.5 and 4.5.6).

We tested our VG-Pedigree pipeline using *DeepTrio* version 1.1.0 with trained child and parent models for variant-calling comparison in HG001. Training used the Ashkenazi (HG002, HG003, HG004), and Han Chinese (HG005, HG006, HG007) trio alignments using the *Giraffe-1000GP* method for model training (see supplementary methods 4.5.6.2). The DeepTrio-called variants achieve the highest accuracy (F1: 0.9976) using *Giraffe-Parent* (Table 4.1A-B). This represents a total variant error (false positive and false negative) reduction of 4,844 variants between *Giraffe-Parent* and *BWA-MEM-hs38d1* relative to an error reduction of 2,925 variants between *Giraffe-1000GP* and *BWA-MEM-hs38d1*. In the 1000GP-excluded variants, the *Giraffe-Parent* accuracy (F1: 0.9748) outperforms *Giraffe-1000GP* (F1: 0.9717) by a greater margin than *Giraffe-1000GP* outperforms *BWA-MEM-hs38d1* (F1: 0.9691). This reflects an error reduction of 3,210 variants between *Giraffe-Parent* and *BWA-MEM-hs38d1* relative to an error reduction of 1,481 variants between *Giraffe-1000GP* and *BWA-MEM-hs38d1*.

We then assessed HG002 and HG005 using the same training method for the model used in evaluating HG001. The models were re-trained with *Giraffe-1000GP*-aligned read data for all trio samples except with chromosome 20 completely held out for validation purposes. Supplementary Figure B.10C-D and Supplementary Table B.5 show the results of training for HG002 and Supplementary Figure B.11C-D and Supplementary Table B.6 for HG005 results. The total number of errors in chromosome 20 reduced from 1,070 to 1,051 (1.78%) and from 1,130 to 909 (19.56%) variants for HG002 and HG005, respectively.

Pipeline	TP	FP	FN	Precision	Sensitivity	F1
Giraffe-Parent	**3,711,135**	6,444	**11,258**	0.9983	**0.9970**	**0.9976**
Giraffe-1000GP	3,708,607	5,687	13,934	0.9985	0.9963	0.9974
BWA-MEM-hs38d1	3,705,297	5,532	17,014	0.9985	0.9954	0.9970
Dragen-hs38d1	3,704,307	**4,586**	18,001	**0.9988**	0.9952	0.9970

(a) *DeepTrio* HG001 All High Confident Regions

Pipeline	TP	FP	FN	Precision	Sensitivity	F1
Giraffe-Parent	**285,663**	5,222	**9,468**	0.9820	**0.9677**	**0.9748**
Giraffe-1000GP	283,261	4,422	11,997	0.9846	0.9591	0.9717
BWA-MEM-hs38d1	281,355	4,356	13,544	0.9848	0.9538	0.9691
Dragen-hs38d1	280,317	**3,398**	14,589	**0.9880**	0.9503	0.9688

(b) *DeepTrio* HG001 All High Confident Regions, 1000GP-excluded

Pipeline	TP	FP	FN	Precision	Sensitivity	F1
Giraffe-Parent	**3,710,974**	8,234	**11,439**	0.9978	**0.9969**	**0.9974**
Giraffe-1000GP	3,705,842	8,751	16,704	0.9976	0.9955	0.9966
BWA-MEM-hs38d1	3,701,516	8,594	20,806	0.9977	0.9944	0.9960
Dragen-hs38d1	3,700,322	**7,181**	22,004	**0.9981**	0.9941	0.9961

(c) *DeepVariant* HG001 All High Confident Regions

Pipeline	TP	FP	FN	Precision	Sensitivity	F1
Giraffe-Parent	**285,782**	6,404	**9,248**	0.9781	**0.9685**	**0.9733**
Giraffe-1000GP	280,890	6,704	14,264	0.9767	0.9514	0.9639
BWA-MEM-hs38d1	279,005	7,010	15,926	0.9755	0.9457	0.9604
Dragen-hs38d1	277,765	**5,610**	17,173	**0.9802**	0.9415	0.9605

(d) *DeepVariant* HG001 All High Confident Regions, 1000GP-excluded

Table 4.1: **VCFeval HG001 *DeepTrio* and *DeepVariant* Performance** VCFeval performance of the graph-based and linear-based pipelines with respect to HG001 GIAB v4.2.1 truth variant call sets stratified by (A) *DeepTrio* on all HG001 regions, (B) *DeepTrio* on HG001 regions excluding 1000GP variants, (C) *DeepVariant* on all HG001 regions, and (D) *DeepVariant* on HG001 regions excluding 1000GP variants. All mapped reads were called using *DeepTrio* and *DeepVariant* v1.1.0 genotyper using trained models. Best values in each column are highlighted in bold text.

We also tested *Giraffe-Parent* using the default *Deep-Trio* version 1.1.0 models, which were not trained with Giraffe alignments. We found that in using the HG005 and HG002 trios *Giraffe-Parent* or *Giraffe-1000GP* with the default *DeepTrio* models outperforms the results achieved using standard BWA-MEM (Table B.7A-B). The same performance gains are observed for *Giraffe-Parent* in more difficult regions for both HG002 and HG005 samples (Tables B.8 and B.9).

ROC curves for DeepTrio calls stratified by genotype quality also show performance gains. Figure 4.3 shows the ROC curves between the graph-based and linear-based alignment methods in HG001 for all confident regions and 1000GP-excluded variants respectively. Supplementary Figures B.10A-B and B.11A-B illustrates performance in the same regions but for the HG002, and HG005 samples using the default DeepTrio models, respectively.

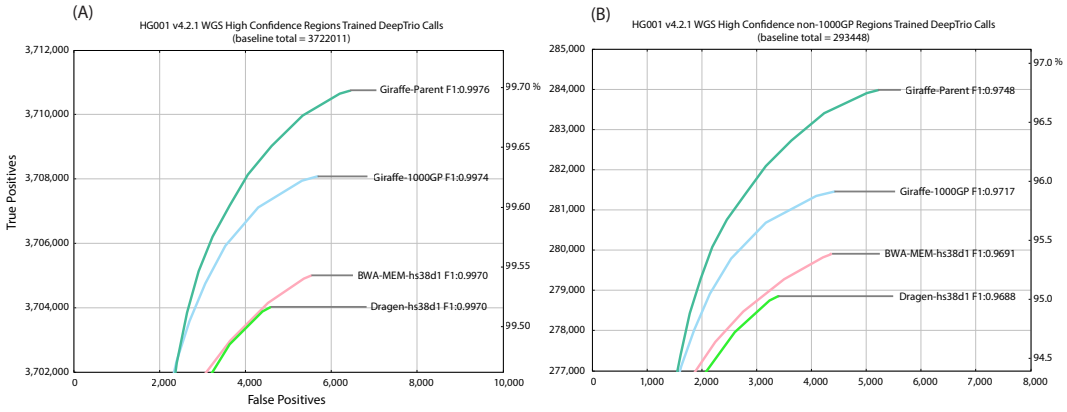


Figure 4.3: ROC curves of *DeepTrio* variant calling performance of the graph-based and linear-based pipelines with respect to HG001 GIAB v4.2.1 truth variant call sets stratified by (A) HG001 high confident whole genome regions using trained *DeepTrio* models, (B) HG001 high confident whole genome regions excluding 1000GP variants using trained *DeepTrio* models.

We also examined the difficult regions of the genome more deeply for the HG001, HG002 and HG005 GIAB samples using the sample-specific stratification [116, 117]. *Giraffe-Parent* outperformed the other examined methods in the sample-specific complex variants containing single heterozygous SNPs and INDELs or compound-heterozygous SNPs except for regions that contain compound-heterozygous variants where at least one of the variants is an INDEL. In those regions, either *BWA-MEM-hs38d1* or *Dragen-hs38d1* achieved the highest F1 scores relative to the Giraffe methods (Tables B.11 B.12 B.13).

4.3.3.1 Comparing to DeepVariant

To compare the mapping performance with non-trio-based calling methods, we ran the DeepVariant single sample genotyper on the same alignments [86]. This evaluation assesses gains in variant calling accuracy brought by mapping to a graph containing the subject's parental information (*Giraffe-Parent*) vs. simply mapping to a linear reference, or a population based pangenome graph (*Giraffe-1000GP*).

During evaluation of DeepVariant calls, like in our DeepTrio evaluations, we focused on using models that were not trained with pangenome graph alignments of the samples used in evaluation. For HG001 alignments, a trained DeepVariant model was used in evaluating HG001 whole genome results. This model was trained using just the *Giraffe-1000GP*-aligned HG002 and HG004 sample reads. For evaluations of DeepVariant calls on HG002 and HG005 alignments, the default models of DeepVariant version 1.1.0 were used. In HG001, the *Giraffe-Parent* method achieves the highest accuracy (F1: 0.9974) representing a total variant error reduction of 9,727 variants between *Giraffe-Parent* and *BWA-MEM-hs38d1* relative to an error

reduction of 3,945 variants between *Giraffe-1000GP* and *BWA-MEM-hs38d1* (Table 4.1C).

4.3.3.2 Illumina Dragen Calling

We additionally tested using Illumina's *Dragen* platform version 3.7.5 variant caller in place of DeepTrio [51]. The Dragen variant caller uses an algorithm similar to that of GATK HaplotypeCaller, and, like DeepVariant, does not use the parental read mappings [87].

We used Dragen to call variants against the Giraffe pangenome and BWA-MEM linear reference mappings. Once again, *Giraffe-Parent* produced the most accurate variant calls for HG002 and HG005. *Giraffe-Parent* produced the highest F1 score (0.9965) in all confident regions for HG002 (Table B.10). This is in contrast with the F1 performance of *Giraffe-1000GP* (0.9953) and *BWA-MEM-hs38d1* (0.9940). Total error is reduced by 18,754 variants between *Giraffe-Parent* and *BWA-MEM-hs38d1* relative to an error reduction of 9,995 between *Giraffe-1000GP* and *BWA-MEM-hs38d1*. For HG005, *Giraffe-Parent* produced the highest F1 score (0.9958) in all confident regions (Table B.14). This is in contrast with *Giraffe-1000GP* (F1: 0.9944) and *BWA-MEM-hs38d1* (F1: 0.9931). Total error is reduced by 20,724 variants between *Giraffe-Parent* and *BWA-MEM-hs38d1* relative to an error reduction of 10,489 between *Giraffe-1000GP* and *BWA-MEM-hs38d1*.

Breaking down the analysis to SNPs and INDELs reveals the same trend. The *Giraffe-Parent* produced the highest F1 scores in HG002 in all examined regions except for the CMRG genes, where *Dragen-hs38d1* achieves a higher accuracy in INDELs (F1: 0.959108) relative to *Giraffe-Parent* (F1: 0.958785) (Tables B.15 B.16 B.17 B.18 B.19, and B.20). Similar statistics are observed in HG005, where *Giraffe-Parent* alignments produce the highest F1 in all SNPs

and INDELs across all confident regions (Tables B.21 B.22 B.23 B.24 B.25).

4.3.3.3 Illumina Dragen Graph Comparison

Illumina's *Dragen* platform version 3.7.5 has also implemented a graph-based mapper. To compare, we also examined the performance of the Dragen graph implementation for mapping and variant calling (termed *Dragen-Graph-hs38d1*) [51]. The *Giraffe-Parent* with *DeepTrio* calling method outperformed *Dragen-Graph-hs38d1* across all confident regions of HG001, HG002 and HG005 GIAB benchmarks (Table B.26).

4.3.4 Candidate Analysis Evaluation

As a quality-control procedure, we investigated the workflow's ability to identify DVs that are relevant to clinical disorders. We ran the workflow on nuclear pedigrees of at least 4 individuals in size. Out of the UDP set of such 50 cohorts with identified candidate variants, a set of 15 cohorts were randomly chosen. The 15 cohorts include 15 probands and 22 unaffected siblings comprising 18 females and 19 males. 10 out of 12 of the UDP probands from these cohorts that have a known genetic diagnosis had their causal variants recapitulated by this workflow. The list of Mendelian models detected include homozygous recessive, de-novo, hemizygous, X-linked, mitochondrial, and compound-heterozygous genotypes. Of the 12 examined probands that have a diagnosis attributed to a CLIA-validated variant, 5 were identified with de-novo dominant non-synonymous changes in an exonic region, 2 had a de-novo dominant frameshift in an exonic region, 2 had compound-heterozygous variants where both were non-synonymous changes in exonic regions, and 1 had a compound-heterozygous variant with a

non-synonymous change in an exon and a change in an intronic/splice-site region. Of the 2 that were missed, 1 had a large structural variant deletion which the candidate analysis workflow was not designed to detect. The other was a male with a de-novo variant on the X-chromosome with a low CADD score that did not pass the workflows default CADD threshold. Supplementary Table B.27 shows the number and type of candidate variants detected by the workflow for all 37 individuals.

In addition, we compared the number and type of clinically-relevant variants that are identified between the affected proband population and their matched unaffected sibling population to indirectly evaluate the pipeline's ability to identify DVs. This analysis runs in two steps. First, for each family, the affected offspring are set as the proband in the workflow and the unaffected offspring are set as the unaffected siblings. Then for the second step, for each family, the unaffected offspring are set as the proband and the affected offspring are set as the unaffected siblings. Finally, the set of candidate DVs from the probands in the first step are compared against the set of candidate DVs from their matched unaffected siblings in the second step.

There is an expected baseline load of rare deleterious variants that all individuals inherit due to de-novo mutation and inefficient selection against segregating variants [47]. Figure 4.4A shows the distribution between these two populations in the 15 pedigree cohort sample set. Figure 4.4B shows the distribution of differences in the candidate DVs between the matched proband and siblings. X-linked recessive candidate DVs were excluded from both populations in order to improve comparability between male and female samples. Compound-heterozygous candidate pairs and candidate alleles that occupy the same locus are also counted as one can-

didate for the purposes of this comparison. The number of candidate DVs in the proband population are significantly different from their matched unaffected sibling's set of candidate DVs (Wilcoxon signed-rank test p-value=0.03). Given a large enough sample set, we might expect the median number of rare deleterious variants in the proband population to be slightly different from the median number of rare deleterious variants in the unaffected sibling population. Due to two factors, the proband's level of genetic burden is hypothesised to be slightly greater than that of their unaffected siblings: all probands in this analysis currently show phenotypic expression of their disease, and the unaffected siblings are of similar age.

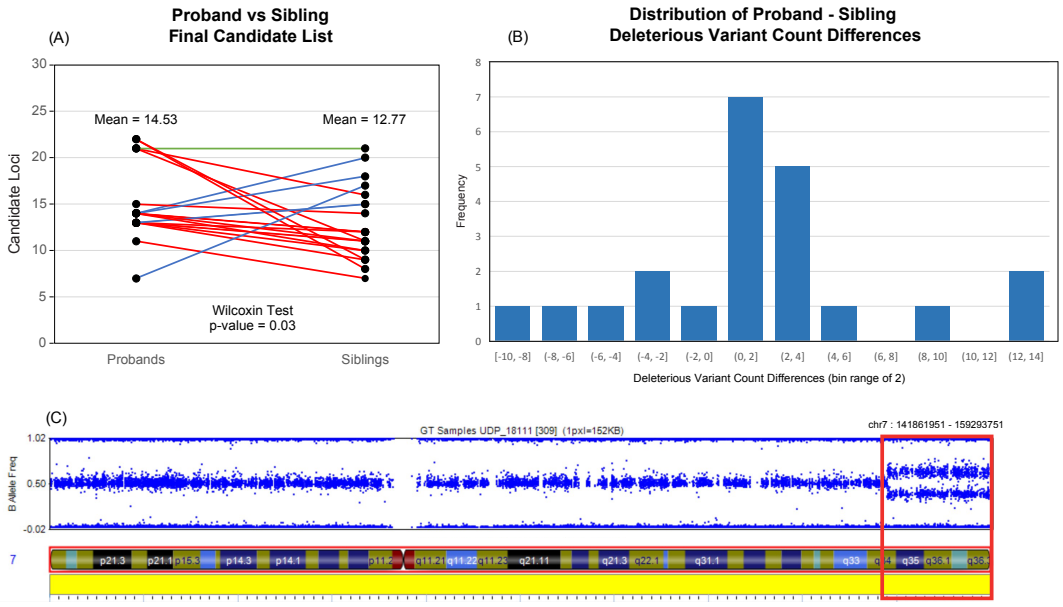


Figure 4.4: Proband-sibling pairwise candidate analysis results on 15 nuclear families of at least quartet in size, comprising a population of 15 probands and 22 siblings. Plot (A) shows the average number of candidate variants between the probands and sibling populations. 17 red lines represent proband-sibling pairs where the proband has more DVs than their matched sibling, 5 blue lines represent probands have less DVs than their matched sibling (blue), and 1 green line where probands have the same number of DVs as their matched sibling. The proband population holds an average of 14.53 DVs while the sibling population has an average of 12.77 DVs. A one-tailed Wilcoxon signed-rank test of the hypothesis that the probands have greater numbers of DVs than their matched siblings produced a p-value of 0.0333. (B) The distribution of proband - sibling DV list size differences. (C) A mosaic region identified by the workflow (red box) overlaid with the snp-chip B allele frequency plot for a UDP sample.

In addition, we ran the workflow on 4 undiagnosed cases that have previously shown a negative or inconclusive clinical exome and negative commercial genome assay results. From these samples, we have produced a number of candidate DVs. Of the 4 cases, 2 have candidate DVs that match their phenotypic profile and are being examined for clinical function, the other 2 cases are undergoing further investigation. One of the 2 cases had an identified mosaic region on chromosome 7 detected by the candidate analysis workflow (Figure 4.4C). Concurrently, we ran the analysis on the HG001(NA12878), HG002, HG005 probands and, as expected, did not detect any signs of mosaicism. Supplementary Tables B.28 and B.29 shows the number and type of candidate variants detected by the workflow.

4.3.5 Runtime Evaluation

The workflows examined are runnable on the Terra platform [10]. When running on a quartet with 30 to 35x coverage paired read data, the workflow takes a little more than 8000 CPU hours for a total cost of approximately $100 (Table B.30). The VG-Pedigree pipeline makes up the majority of the computation at about 8000 CPU hours and costs $92-95, while the candidate analysis workflow runs in about 200 CPU hours costs $3-5. Costs can vary based on the load of the cloud compute system and the availability of lower-cost preemptable nodes.

4.4 Discussion

There is growing evidence that rare variants have the effect sizes, diversity and abundance necessary to explain a substantial portion of human genetic load [48, 99, 65]. Pedi-

grees can help resolve harder-to-study regions by giving orthogonal evidence in the form of Mendelian inheritance to enhance the statistical power and phasing accuracy to categorize compound-heterozygote and de-novo variation from a list of called variants [98, 85, 92, 109]. Graph-based approaches leverage additional variation information during read mapping to mitigate the problems of alignment to complex regions of the genome [44, 102]. The methods and software developed in this project are designed to maximize the biological information available to detect and interpret individual-level variation. The software developed is scalable so that it can easily run on high performance compute clusters that support common batch systems like Slurm [121] or Kubernetes [53]. It is publicly accessible in the `toil-vg` GitHub repository and in `WDL` format which is published in the Dockstore repository [81, 10].

Alignment and genotyping performance of short-sequenced reads is improved across all examined confident regions in the GIAB samples. This is due to the parental genotypes contributed to the pangenome reference used in the *Giraffe-Parent* method that better match the child's reads. The result of these improvements translate to better coverage, mapping qu ality and greater variant calling accuracy in both confident and difficult regions of the genome. All examined UDP cases that have a known genetic diagnosis based on the Mendelian models covered by the previously published candidate analysis workflow have their causal variants recapitulated by this workflow [45]. The candidate analysis evaluation indicates detectable differences in the number of candidate DVs identified between the affected and unaffected offspring populations. This result shows a similar trend to that of the analysis done on exome datasets from a larger sample set, which also showed a statistically significant difference [45]. The main improvement in this analysis over the previous analysis is that this analysis covers the whole

genome including intronic and intergenic regions.

A number of areas can be improved within this workflow. One example is the training model used in DeepTrio. Our training used a very limited number of benchmark samples, which was limited further to leave benchmark data for testing and development. Given these limitations, there is room to improve the *DeepTrio* model when additional well-sequenced and diverse benchmark samples become available.

Variant calls from graph-based alignments are prone to error due to the conversion of the native graph alignment map (GAM) format output from VG alignments to the linear reference BAM format. Information about the exact path of reads is lost during this projection step which can result in reads appearing different from the linear reference genome when the variant is already present in a path in the graph reference.

Structural Variants (SV) are an important component to the set of rare variants that contribute to disease [119, 4]. In previous work, there have been efforts to tailor pangenome graphs and variant caller algorithms to improve the accuracy of detecting SV [102]. Another avenue to improve this workflow is to apply pangenome graphs with incorporated SV information as a module that runs concurrently to the VG-Pedigree workflow. One of the samples in the candidate DV analysis was missed by this workflow as it contained a large SV deletion. Incorporating SVs into the VG-Pedigree workflow would aid in the detection of such variants.

Refinements to the CADD scoring metrics can be made to enhance the detection of specific variants. One of the samples in the candidate DV analysis that was missed by this workflow was a male with a de-novo variant on the X-chromosome with a low CADD score that did not pass the workflows default CADD threshold. To remedy this, the CADD threshold

for de-novo male X-linked variants can either be lowered to enhance sensitivity or the CADD program would need to adjust their scoring metrics to take into account such variants and up-weigh their scores.

Further runtime improvements could also be made. The workflow takes about 1.5 days and approximately 8000 total CPU hours at a cost of about $100 to process one family. This is moderately expensive and slow relative to traditional methods, which have well-tuned hardware acceleration solutions and years of work optimizing computation time. GPU acceleration or field-programmable-gate-array (FPGA) implementations of the graph alignment algorithm could substantially accelerate the computation of the graph-based algorithms.

There are a number of refinements that could be made to the most expensive parts of this workflow. Reference construction of the parental graph could be improved by altering and pruning the haplotype index with the haplotypes discovered by the trio-backed phasing stage of the pipeline. The use of graph-based variant callers would remove the need to surject alignments to linear BAM files and therefore maintain potentially more information that could be used to produce more accurate calls.

Additional orthogonal methods can be applied to the workflows presented. The reconstruction of sequences in a sample from sequence data alone, known as de-novo genome assembly, can be used to support evidence of the genotypes detected in this workflow. One tool known as `WHdenovo` can apply pedigree information and long-sequence reads to enhance the construction of sample-specific assemblies that can resolve potential genotyping errors in this workflow [42, 41].

New pangenome graphs are continuously being updated and tested as more popula-

tion variation is characterized. The Telomere-to-Telomere genome project (T2T) has recently released a genome reference which exhaustively captures the centromeric and telomeric sequence better than the previous GRCh38 version of the human genome [80]. The Human Pangenome Reference Consortium (HPRC) is a group of research institutions that are tasked with the development of a pangenome reference using the latest methods and data. By characterizing regions of the genome not well represented by existing variant datasets, the pangenome references developed by the HPRC that incorporate new T2T sequences should further improve the performance and accuracy of the workflows presented in this paper.

4.5 Methods

4.5.1 VG Pedigree Workflow

Pangenome graphs provide a framework for leveraging genomic variation information to create a better-informed mapping procedure than that provided by a linear genomic reference. The workflow presented here goes through a number of stages (Fig. 4.1A). The first stage establishes parental haplotypes to construct a parental-backed graph reference. It takes short reads from a trio and aligns each to a population-informed graph reference. We use a graph based on the 1000 Genomes dataset [9, 102]. It is still the largest and most diverse set of phased genotypes available to the public with broad consent. The 1000GP graph is based on the `hs38d1` human reference genome and the 1000 Genomes Project phase 3 variant set that has been lifted over from GRCh37 to GRCh38 genome coordinate space, and is available in a publicly accessible Google Cloud bucket.

Alignment of the parent-child trio to the 1000GP graph goes through a number of steps that split and merge read alignments to enable distributed computation (Fig. 4.1B, Supplementary Fig. B.5). This greatly reduces time spent aligning reads, which is a major bottleneck for the workflow. Afterwards, each chunked alignment is projected back to the linear genome reference coordinate space and corrected for duplicates and missing mate information and INDELs are realigned using ABRA2 [77]. Following alignment, samples in the trio are variant-called, producing a per-sample gVCF genotype called file. A trio-based *DeepVariant* extension [86], Google's *DeepTrio* [56], is used to call variants in this workflow. *DeepTrio* first generates images based on the alignments between the parent and child reads. Then the *DeepTrio* variant caller is run concurrently to call gVCFs for each contig for each sample in the trio. The gVCFs are next joint-called with the Glnexus package [122] in order to merge and recall potentially uncalled variants in the trio. Joint-calling gVCFs enhances DeepVariant-based calls by reexamining trio variant sites that were confidently called in one sample but not another. The joint-called trio VCF is then divided by autosomal and sex-chromosomal contigs, with the mitochondrial contig only preserving the maternal set of called genotypes and the Y chromosomal contig preserving the paternal set of called genotypes.

A number of different schemes for phasing these variants were explored using combinations of *Eagle* [66], *WhatsHap* [71], and *SHAPEIT4* [26]. Supplementary Table B.31 illustrates the performance of combinations of these programs when phasing the GIAB HG002 sample. Supplementary Table B.32 shows phasing performance for the GIAB Ashkenazi trio with respect to GRCh38- or GRCh37-based graph alignments. Using *Eagle* followed by *WhatsHap* produced the largest blocks of phased variants while maintaining a switch error rate close

to, or better than, the method with the largest median haplotype block size from this list: *WhatsHap* in combination with *SHAPEIT4*. Following the alignment and variant calling step, a phasing sub-pipeline is run on these contig VCFs using the Eagle-WhatsHap phasing method (Fig. 4.1C, Fig. B.6). Missing genotypes are imputed using *Eagle* version 2.4.1 [66]. Finally the contig VCFs are phased with trio- and read-backed methods using *WhatsHap* [71]. That final set of contig VCFs are then filtered down to just the parental genotype sets and passed into the graph construction workflow.

Following the phasing stage of the workflow, the phased variants from that step and a linear reference in FASTA are passed as input into the graph construction step (Fig. 4.1D, Fig. B.7). VG mappers use a variety of indexes [102]. To facilitate this need, the construction workflow generates a combination of indexes based on the requirements of the VG Giraffe mapper.

After constructing the parental graph, the offspring reads can be realigned to it. GVCFs are called from offspring alignments to the parental graph reference. (Fig. 4.1B, Fig. B.5). Finally, variants are jointly called, once again with the Glnexus package [122], by combining previously-computed gVCFs of the 1000GP-aligned parents with gVCFs derived from the parental graph-aligned offspring.

The methods developed here for the VG-Pedigree workflow are implemented in the software framework "toil-vg" under the 'toil-vg pedigree' subcommand which makes use of the TOIL workflow engine [114] for cloud-based and cluster-compute systems and is available on GitHub at `https://github.com/vgteam/toil-vg`. The workflow is also made available in `WDL` format in the Dockstore [81, 10] repository at `https://dockstore.org/workflows/gi`

thub.com/vgteam/vg_wdl/vg-pedigree-giraffe-deeptrio:master.

4.5.2 Candidate Analysis Workflow

A primary endpoint goal for this workflow is variant detection to identify likely causes of the genetic disorders in the UDP cases. Traditional variant filtration techniques narrow down a set of variants, but they are usually not exhaustive enough to narrow the list down to an actionable number of variants without truncation [84, 55]. Further, they often do not specialise in the detection of compound-heterozygous candidates in non-coding regions. Traditionally, a large proportion of work is needed to validate the clinical functionality for each variant [11]. Given this downstream cost, this workflow focuses on reducing that cost by minimizing the number of variants that need to be examined in the final list. The analysis workflow takes in a very large set of variants and filters them by examining a series of variant attributes each of which follows an order of most-certain to least-certain true-positive data types (Fig. 4.1F).

Additional improvements and features were added to this implementation of the methods developed in the Gu et al. study. In this paper, we have adapted all components and annotations used by the workflow to be compatible with the GRCh38 reference genome coordinate system. The CADD engine software suite has been updated to version 1.6 which incorporates greater accuracy in determining deleterious variants located in splice sites and introns [90]. We have also updated the population annotation dataset to use GnomAD v3.1 which has incorporated a larger proportion of samples producing more accurate and exhaustive population allele frequencies [54]. The maximum minor allele frequency (MAXMAF) calculation implemented in the `population/deleterious-backed variant filtration` module was altered to use

a binomial instead of a poisson distribution (Fig. B.8D). A critical bug was patched that was found to erroneously output X-linked candidate variants for females. We implemented a new module that automatically detects the presence, location and type of copy number variant (CNV) mosaicism in the proband.

The alignment and variant calling workflow output is processed with various annotation programs before they are able to be passed as input into the candidate analysis workflow. Post processing the final datasets comprises SnpEff annotation, INDEL-realignment, and converting to a one variant per row format that has pedigree consistent INDELs, for each of the samples in the pedigree (Fig. 4.1E, Fig. B.9). The CADD [90] software suite is used in this analysis workflow to predict the deleteriousness of a given variant. Any variants that are unique to the CADD database in the joint VCF have a deleterious score calculated by the software.

The analysis portion of the workflow examines and filters the pedigree variant file in the context of Mendelian inheritance, alignments against the parental-based graph reference, population variant frequency, and predictions of variant effects on gene function and expression [45] (see supplementary methods 4.5.7). Using these filters generates a set of variants that are further filtered by examining the BAM files for sequence and alignment noise surrounding each variant [45]. This produced a final short list for clinical examination. The workflow then cleans up the resulting candidate list of identifiable errors and artifacts. Typical candidate lists produced by this pipeline consist of 10-50 variants (Tables B.27 and B.28). These lists include compound-heterozygous variants located in non-coding regions of the genome.

One new implementation of the workflow is the detection of CNV mosaicism. Mosaicism is a genetic event where a single sample possesses multiple populations of cells that

possess different proportions of variants. The goal of the program is to detect stretches of phased variants that show consistent and significant evidence for deviation in allele depth (AD) contributed by the mother and father. The first step is to phase a set of heterozygous genotypes in the proband by examining the parental genotypes. The phasing done here is more stringent than in the previous method described in the vg pedigree workflow because we are looking for a sequence of easily phasable SNPs and so the procedure is rule-based instead of *WhatsHap* which is based on statistical models. A given genotype in the proband is phaseable if two conditions are met: at least one parent has a homozygous genotype, and the other parent is heterozygous. If a large enough proportion of genotypes are phased in this way, the program examines regions of sufficient length for consecutive stretches of allele balance deviation. A sliding window of 10,000 phased genotypes is used to scan each chromosome and find the boundaries of the mosaic region. For each SNP within this window, the AD of one parent is subtracted from the AD of the other parent. A t-test is applied to the list of AD differences within the window to test if the distribution is significantly different from the null model of no difference. If the t-test statistic is greater than the input threshold, then a region of possible mosaicism is detected and subsequently logged in a separate file for further examination. This threshold was determined empirically against mosaic-positive samples obtained by the UDP. This differs from traditional CNV callers in that this program incorporates trio information to look for partial deletion or duplication events at megabase scales at a continuous level of granularity.

This program can also determine three types of mosaicism: uniparental isodisomy-disomy, trisomy-disomy, and monosomy-disomy. In uniparental isodisomy-disomy mosaicism the individual has populations of cells where a proportion of their genome shares both copies

from only one of their parents, and the rest of their cells have inherited a copy from both parents. These types of mosaics are detected by examining the total read depth of the child and parents within the candidate mosaic region. If the proportion of total read depth between the child and parents are the same, and the proportion of ADs of the phasable SNPs between the child and parents are not the same, then the program will classify the mosaic region as uniparental isodisomy-disomy.

In trisomy-disomy mosaicism, the individual has populations of cells where a proportion of their genome has inherited two copies of the same chromosome from one of their parents and one copy from the other parent while the rest of their cells have inherited a copy from both parents. If the proportion of total read depth in the child is greater than their parents, then the region is classified as trisomy disomy mosaicism. Alternatively, in monosomy-disomy mosaicism, the individual inherits only one copy from only one parent in some of their cells, and the rest of their cells inherit one copy from each parent. In this case, if the total read depth in the child is less than that of their parents, then the region is classified as monosomy disomy mosaicism.

All modules have been implemented in software containers to improve portability and interoperability with other workflow engines [53, 95]. The candidate analysis workflow is implemented within the "toil-vg" software package under the `toil-vg analysis` subcommand. The candidate analysis workflow is also available in `WDL` format in the Dockstore repository at https://dockstore.org/workflows/github.com/cmarkello/bmtb_wdl/bmtb:main.

4.5.3 Read Simulation

To simulate reads using the vg framework we generated pangenome graphs representing the haplotypes of HG002. To build these graphs we used a reference genome (hs38d1) and a variant population dataset (GIAB HG002 version 4.2.1 high-confidence variant sets), using the vg construct command to build the pangenome graph. Since the hs38d1 reference genome is also contained in the graph whose mapping to which we were evaluating we used the reference location of in hs38d1 to evaluate if a read was correctly mapped.

First a sample graph was constructed using the vg container

`quay.io/vgteam/vg:ci-2890-655a9622c3d60e87f14b88d943fbd8554214a97`,

on the Genome-in-a-Bottle(GIAB) `HG002` sample trio-phased variant data from https://ftp-trace.ncbi.nlm.nih.gov/giab/ftp/release/AshkenazimTrio/HG002_NA24385_son/NISTv4.2.1/GRCh38/SupplementaryFiles/HG002_GRCh38_1_22_v4.2.1_benchmark_phased_MHCassembly_StrandSeqANDTrio.vcf.gz, and the linear reference sequence `GCA_000001405.15_GRCh38_no_alt_plus_hs38d1_analysis_set.compact_decoys.fna.gz` from https://storage.googleapis.com/cmarkell-vg-wdl-dev/grch38_inputs/GCA_000001405.15_GRCh38_no_alt_plus_hs38d1_analysis_set.compact_decoys.fna.gz.

Next, reads were simulated using the commands from the following script https://github.com/cmarkello/vg-pedigree-paper/blob/main/scripts/wgs_mapping_simulation/sim_reads.sh.

The simulated reads were mapped using `BWA-MEM-hs38d1` against the linear reference, `VG`

Giraffe against Primary, 1000GP and the Parental graph reference as produced by the VG Pedigree workflow using commands from the script https://github.com/cmarkello/vg-pedigree-paper/blob/main/scripts/wgs_mapping_simulation/run_mapevals.sh. Mappings made by BWA-MEM-hs38d1 against the linear reference were injected to graph space using the same sample graph of HG002 and hs38d1 that was used to simulate reads.

Finally, the aligned reads were evaluated for mapping accuracy using the following commands from the script

https://github.com/cmarkello/vg-pedigree-paper/blob/main/scripts/wgs_mapping_simulation/run_mapevals.sh

followed by the rendering of receiver-operator-curves of the mapping results with the commands from the script https://github.com/cmarkello/vg-pedigree-paper/blob/main/scripts/wgs_mapping_simulation/plot_roc_simulated_mapped_reads.sh. Graph Alignment Map (GAM) file stats were computed using 'vg stats' from VG version v1.31.0. Percent identity between aligned reads and the reference sequence were calculated using 'vg gamcompare' from VG version v1.31.0.

4.5.4 Graph Construction

All genome graphs including the 1000GP 1000 Genomes Project population graph reference and the Primary linear graph reference were constructed using VG version v1.27.0. A custom form of the GRCh38 cannonical FASTA sequence was generated where all decoy

contigs were merged together into a single decoy contig in order to reduce the contig number of the file and is available at https://storage.googleapis.com/cmarkell-vg-wdl-dev/grch38_inputs/GCA_000001405.15_GRCh38_no_alt_plus_hs38d1_analysis_set.compact_decoys.fna.gz. This FASTA file was used as the reference framework for these graph references.

All methods of graph construction can be reproduced using the following scripts from the script repository URL https://github.com/cmarkello/vg-pedigree-paper/tree/main/scripts/graph_construction.

4.5.5 Read Mapping

Real reads obtained from the GIAB sample trios `HG002` and `HG005` were mapped using 4 mappers: Illumina's Dragen-hs38d1 version `3.7.5`, BWA-MEM-hs38d1 version `0.7.17-r1188`, Giraffe from VG version `v1.31.0`, and the VG Pedigree mapping workflow using VG version `v1.31.0`.

The commands for running Illumina's Dragen-hs38d1 module are described in the script from the following URL: https://github.com/cmarkello/vg-pedigree-paper/blob/main/scripts/wgs_mapping_experiments/Dragen-hs38d1_map.sh.

The commands for running BWA-MEM-hs38d1 are described in the script from the URL: https://github.com/cmarkello/vg-pedigree-paper/blob/main/scripts/wgs_mapping_experiments/bwamem_map.sh.

The commands for running VG Giraffe are described in the script from the URL: https://github.com/cmarkello/vg-pedigree-paper/blob/main/scripts/wgs_mapping_experim

ents/giraffe_map.sh.

The commands for running the VG Pedigree workflow are described in the script from the URL: https://github.com/cmarkello/vg-pedigree-paper/blob/main/scripts/wgs_mapping_experiments/giraffe_pedigree_map.sh.

4.5.6 Variant Calling and Evaluation

4.5.6.1 Variant Callers

We used two different variant callers for the purposes of evaluating robustness in mapping performance. Google's DeepTrio version `1.1.0` was used as the main variant caller of choice and Illumina's Dragen-hs38d1 module version `3.7.5` was used as the variant caller for alternative method comparison.

All mapped `HG002` and `HG005` trio samples from each of the mappers from Section 4.5.5 were called using both variant callers. For DeepTrio calling, the default model that comes with version `1.1.0` was used to call genotypes over the whole genome for each sample. For DeepTrio calling using the models that were trained using the VG Giraffe-based alignments were run on chromosome 20 since that contig was held-out during the training process described in 4.5.6.2.

Called variants were subsequently evaluated by running Illumina's `Hap.py` version `0.3.12` from the container `jmcdani20/hap.py:v0.3.12` and RealTimeGenomics' `vcfeval` version `3.12.1` using the GIAB version `4.2.1` truth-sets for the `HG002` and `HG005` samples.

Commands for running all variant calling and evaluations can be found in the scripts located in the following URL: https://github.com/cmarkello/vg-pedigree-paper/tre

e/main/scripts/wgs_calling_experiments.

4.5.6.2 DeepTrio and DeepVariant Training

To enable DeepTrio to make maximal use of the pedigree graph, DeepTrio was re-trained on samples mapped against the pangenome graph. Re-training used two trios, HG002-HG003-HG004 and HG005-HG006-HG007. Each sample was mapped at 35x coverage. To improve generalization across sequencing depth, training examples were generated using random downsampling of examples by 1.0, 0.8, 0.7, 0.6, and 0.5 (corresponding to average sequence depth of 35x, 28x, 24.5x, 21x, and 17.5x). Training used examples from chr1-chr19, with chr21 and chr22 used as a tune set to select a model, and chr20 fully withheld as an independent evaluation dataset. Training used the default parameters for DeepTrio, with the exception of minimum mapping quality being set to 1, which enables DeepTrio to better take advantage of the pangenome graph's improvement in difficult to map regions. One model was trained to call variants in the child, and another model was separately trained to call variants in the parent. DeepTrio uses each of these models during runtime to call variants for both parents and child.

Two sets of DeepTrio trained models were used in evaluation. The model trained using samples HG002-HG003-HG004 and HG005-HG006-HG007 was applied to the whole genome evaluations of the HG001 sample. The model trained using samples HG002-HG003-HG004, HG005-HG006-HG007 and HG001-NA12891-NA12892 was applied to the chromosome 20 evaluations of the HG002 and HG005 samples.

A re-trained DeepVariant model was used in evaluating HG001 whole genome results. This model was trained using just the HG002 and HG004 sample alignments against the

pangenome graph. The training examples were generated using random downsampling of examples by 1.0, 0.8, 0.7, and 0.6 (corresponding to average sequence depth of 35x, 28x, 24.5x, and 21x). All other training methods and parameters were similar to those used in training the DeepTrio models.

4.5.7 Candidate Analysis Workflow Modules

The first steps of the analysis workflow find false negative variant genotypes in the parents relative to the child such that they appear de-novo in the proband. It performs an intense pedigree-aware Bayesian re-genotyping to recover many apparent de-novo states back into simple dominant mendelian inheritance from a parent (Supplementary Fig. B.8A-B). A full description of this method and heuristic derivation can be found in supplementary manuals 1 and 2 of the Gu et al. paper [45].

The next module identifies and filters variants for various modes of Mendelian inheritance, population frequencies, and predictions of deleteriousness (Supplementary Fig. B.8C). Mendelian models include homozygous recessive (HR), de-novo (DN), mendelian-inconsistent regions that focus primarily on the proband's homozygous variant or hemizygous genotypes (MI), X-linked (XL), mitochondrial, and compound-heterozygous genotypes (CM). CM genotypes are examined in regions of defined gene loci using compound Phred-scaled CADD scores known as Virtual Mendelian Model (VMM) for each combination of heterozygous pairs of the CH [45]. A full description of this method and heuristic derivation can be found in supplementary manuals 3 of the Gu et al. paper [45].

After identifying Mendelian model candidacy, a population-based and deleterious-

based filter is applied that is based on CADD, VMM and an assortment of population databases (Supplementary Fig. B.8D). The databases used for this task include 1000Genomes [21], UK10K [118], ExAC [63], GnomAD [54], and the UDP's set of internal samples. A full description of this method and heuristic derivation can be found in supplementary manual 5 of the Gu et al. paper [45].

For the remaining variants, their loci are further examined through a BAM file curation procedure (Supplementary Fig. B.8E). Variants at these loci are filtered based on density of mismatches and consistency of pileups across sequence reads that are aligned in this region. A full description of this method and heuristic derivation can be found in supplementary manual 4 of the Gu et al. paper [45].

The last module investigates potentially false positive de-novo variants due to the density of mismatches in a region surrounding the variant in the alignment data and filters them out (Supplementary Fig. B.8F). The full description of this method and heuristic derivation can be found in supplementary manual 6 of the Gu et al. paper [45].

In parallel to the filtering modules described above, two additional types of variants are examined for candidacy. The first of which looks for variants that have no coverage in the proband but evidence for coverage is present in everyone else in every population dataset. These variants are referred to as called-no-coverage (CNC). A module in the analysis pipeline examines these variants and checks for their presence in other population databases (Supplementary Fig. B.8G). This method supplements the proband's set of read coverage information with the read coverage information collected by the UDP lab to enhance the detection of large double deletions on the scale of exons. This module is described further in manual 7 of the Gu et al.

paper [45].

Code Availability and Data Access

The scripts available for running the graph reference construction, mapping simulation and variant calling experiments is provided at https://github.com/cmarkello/vg-pedigree-paper/scripts. The repository can be downloaded directory using

```
git clone https://github.com/cmarkello/vg-pedigree-paper.git.
```

The main VG Pedigree workflow is available both in TOIL format from the pedigree sub-command of the toil-vg program as available from https://github.com/vgteam/toil-vg.git. The workflow is also available in WDL format and is available on Dockstore at the following URL: https://dockstore.org/workflows/github.com/vgteam/vg_wdl/vg-pedigree-giraffe-deeptrio:master.

The candidate analysis workflow has been made available as a separate program for interoperability purposes and is available in TOIL format from the analysis sub-command of the toil-vg program as available from https://github.com/vgteam/toil-vg.git. The workflow is also written in WDL format and is available on Dockstore at the following URL: https://dockstore.org/workflows/github.com/cmarkello/bmtb_wdl/bmtb:main.

Both the VG-Pedigree workflow and the candidate analysis workflow are implemented in the software workflow engine TOIL [114] for cloud-based and cluster-compute sys-

tems under the software framework *toil-vg*. They are callable using the `toil-vg pedigree` and `toil-vg analysis` subcommands, respectively. *toil-vg* is available on GitHub at https://github.com/vgteam/toil-vg. The workflows are also made available in `WDL` format in the Dockstore [81] repository at https://dockstore.org/workflows/github.com/vgteam/vg_wdl/vg-pedigree-giraffe-deeptrio:master and https://dockstore.org/workflows/github.com/cmarkello/bmtb_wdl/bmtb:main.

Input data used in the mapping evaluation, variant calling evaluation and runtime evaluation are all publicly available and listed in the scripts posted in the github repository: https://github.com/cmarkello/vg-pedigree-paper. Input data used in the candidate analysis evaluation experiments have been or are being submitted to the database of Genotypes and Phenotypes (dbGaP).

Low complexity, low mappability and MHC regions were defined by the following bed files, respectively, and intersected using Bedtools against the GIAB sample-specific all-confident region benchmark bed files: https://ftp-trace.ncbi.nlm.nih.gov/giab/ftp/release/genome-stratifications/v2.0/GRCh38/union/GRCh38_alldifficultregions.bed.gz, https://ftp-trace.ncbi.nlm.nih.gov/giab/ftp/release/genome-stratifications/v2.0/GRCh38/union/GRCh38_alllowmapandsegdupregions.bed.gz, https://ftp-trace.ncbi.nlm.nih.gov/giab/ftp/release/genome-stratifications/v2.0/GRCh38/OtherDifficult/GRCh38_MHC.bed.gz [88, 117]. The analysis of called variants in the HG002 complex medically relevant genes used the HG002 CMRG v1.00 VCF https://ftp-trace.ncbi.nlm.nih.gov/giab/ftp/release/AshkenazimTrio/HG002_NA24385_son/CMRG_v1.00/GRCh38/SmallVariant/HG002_GRCh38_CMRG_smallvar_v1.00.v

cf.gz and BED file https://ftp-trace.ncbi.nlm.nih.gov/giab/ftp/release/AshkenazimTrio/HG002_NA24385_son/CMRG_v1.00/GRCh38/SmallVariant/HG002_GRCh38_CMRG_smallvar_v1.00.bed [116]. Sample-specific difficult region bed files were extracted from https://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/release/genome-stratifications/v3.0/GRCh38/GenomeSpecific/.

Candidate deleterious variants for the proband and sibling populations of the 15 cohort proband-sibling pairwise analysis can be found in the supplemental files "proband_cv.xlsx" and "sibling_cv.xlsx", respectively.

Scripts for reproducing the methods for graph construction can be found in https://github.com/cmarkello/vg-pedigree-paper/tree/main/scripts/graph_construction.

Scripts for reproducing the mapping evaluation experiments can be found in https://github.com/cmarkello/vg-pedigree-paper/tree/main/scripts/wgs_mapping_simulation.

Scripts for reproducing the real-data mapping and variant calling evaluation can be found in https://github.com/cmarkello/vg-pedigree-paper/tree/main/scripts/wgs_mapping_experiments and https://github.com/cmarkello/vg-pedigree-paper/tree/main/scripts/wgs_calling_experiments, respectively.

4.6 Acknowledgements

We gratefully thank Dr. William Gahl, Dr. David Adams and the members of the NHGRI Undiagnosed Diseases Program for providing the resources, experiment execution assistance and data access that has made this project possible. We would also like to thank Jouni Sirén, Erik Garrison, Xian Chang, Jean Monlong, Adam Novak and the rest of the Variation Graph team at the UCSC Genomics Institute for providing the tools and methods developed from which much of this work is built upon. All pipelines and evaluations used the computational resources of the NIH HPC Biowulf cluster at the National Institutes of Health, Bethesda, MD (https://hpc.nih.gov).

Work pertaining to the processing and access to NIHUDP data was supported in part by the Intramural Research Program of the National Human Genome Research Institute and the Common Fund, Office of the Director, National Institutes of Health. Research reported in this chapter was supported by the National Institutes of Health under Award Numbers U41HG010972, R01HG010485, U01HG010961, OT3HL142481, OT2OD026682, U01HL137183, and 2U41HG007234. The views expressed in this chapter are those of the author and do not necessarily represent the views of the National Institutes of Health.

Part IV

Tools for Analysing Variants

Chapter 5

BRCA Exchange Repository

5.1 Preamble

In this chapter I present contributions that I've made towards the development of the BRCA Exchange project which pertains to the clinical interpretation of variants after variant discovery is made. BRCA Exchange is a database and website which aggregates *BRCA1* and *BRCA2* gene variation data for the purposes of facilitating collaborative research activity [23]. Much of what I contributed to related to the collection of data from various *BRCA1* and *BRCA2* database silos from around the world. I also contributed to the design and implementation of some components of the website. These primarily included the graphical user interfaces (GUI) of database filters and column selections for the data-table as well as an interactive visual representation of allele location and frequencies of various types of variants in the form of a lollipop chart.

5.2 Introduction

Over the past 60 years, geneticists began to relate genetic measurements with measures of clinical significance. Recent advances in genetic sequencing technologies gave researchers the ability to investigate these genetic traits in more granular detail, down to the level of single nucleotide changes. This level of precision eventually led to the growing development of clinical diagnostics which eventually resulted in variant databases that have been curated by different academic institutions from around the world. The advent of these databases being housed by different global institutions naturally fractured the development of standards of variant interpretation and clinical significance. Today, a number of different variant databases with clinical interpretation are available and each have attempted to bridge this gap in standardization such that variants in one database can be compared or combined with variants in another database. These include but aren't limited to the Online Mendelian Inheritance in Man (OMIM) [7], the Human Genome Organization (HUGO) [1], the Human Genome Variation Society (HGVS) [27], the Human Gene Mutation Database (HGMD) [108], the Leiden Open Variation Database (LOVD) [33], and the ClinVar database [60].

The Global Alians for Genomics and Health (GA4GH) has been tasked with facilitating the organization and standardization of variant representation and interpretation with the goal of combining the worlds collection of data to further enhance the quality of genetic research and clinical application [111]. The BRCA Exchange Challenge was a pilot project under this organization that was focused on the genes *BRCA1* and *BRCA2* due to their popularity and the wealth of available data for those genes. The goal of BRCA Exchange was to demonstrate

the viability of combining data from various global variant databases in the form of a website that services a centralized database [23].

5.3 Variant Database

The BRCA Exchange variant dataset is comprised of a large number of global databases. These include ClinVar[60], Ex-UV[112], LOVD[33], GnomAD[54], the Breast Cancer Information Core (BIC)[110], and the ENIGMA consortium[107]. The resulting database houses more than 18,000 *BRCA1* variants and more than 22,000 *BRCA2* variants. Variants were filtered based on reference bias with the current reference genome or had erroneous nomenclature terms as determined by the HGVS standard. Variants with different representations but are genomically equivalent are determined by deriving an alternative allele string for each variant that consists of the variant plus flanking genomic bases, comparing these allele strings and merging any variants that yield equivalent allele strings. With this strategy, the pipeline detects equivalent variants that show no apparent similarity in their HGVS strings or genomic coordinates, such as complex indels that produce the same alternative allele string through differing combinations of insertions and deletions. Figure 5.1 shows the amount of overlap in variants within the databases in BRCA Exchange.

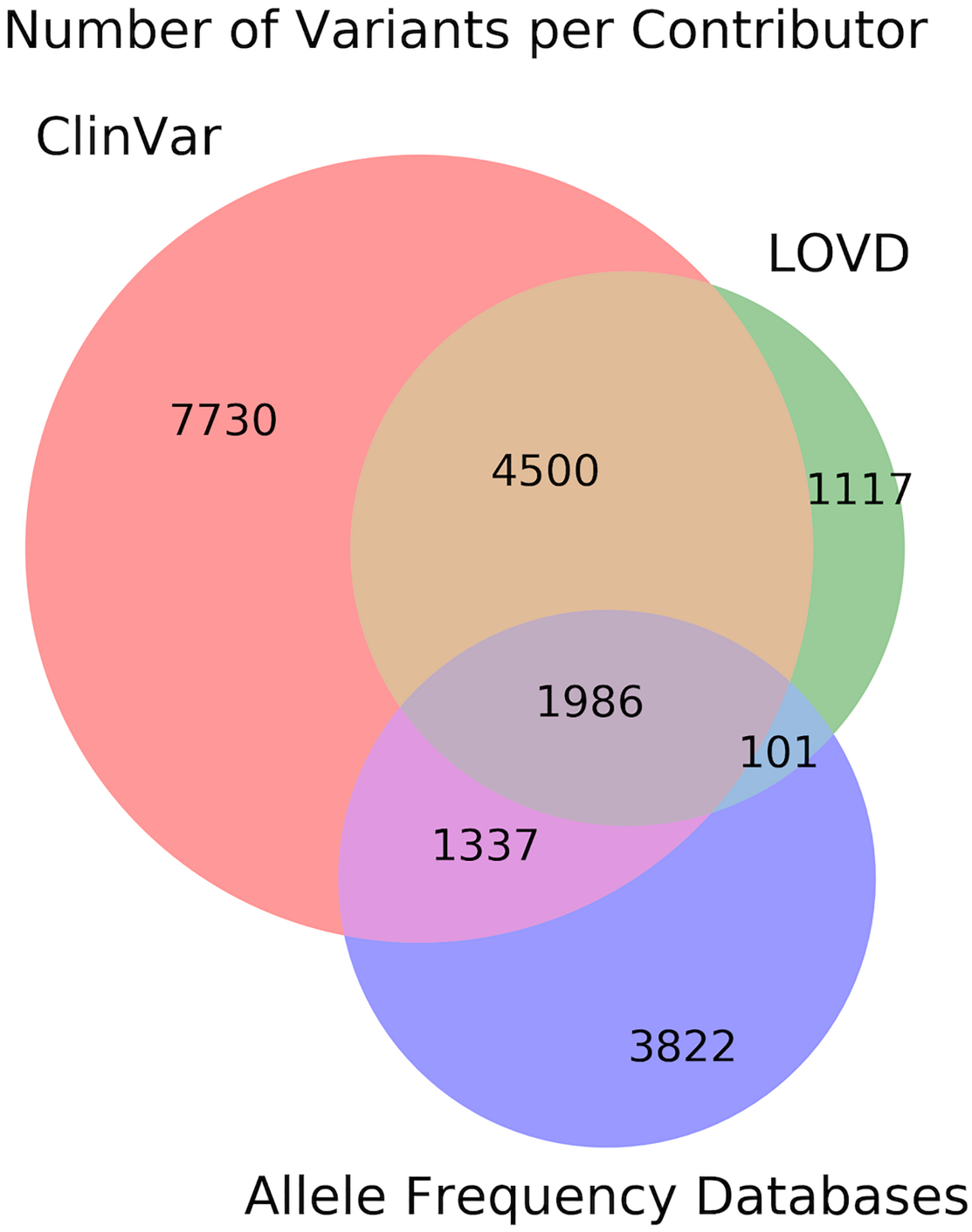


Figure 5.1: BRCA Exchange Database Overlap between ClinVar Leiden Open Variation Database (LOVD) and the allele frequency databases: Exome Aggregation Consortium (ExAC), 1000 Genomes project, and the Exome Sequencing Project (ESP).

5.4 Visualization of Variants

A number of interactive features were implemented in the BRCA Exchange website. The components I contributed to included the visual layout and interactivity of the column and database filters. Figure 5.2 shows a sample of the graphical layout of this feature.

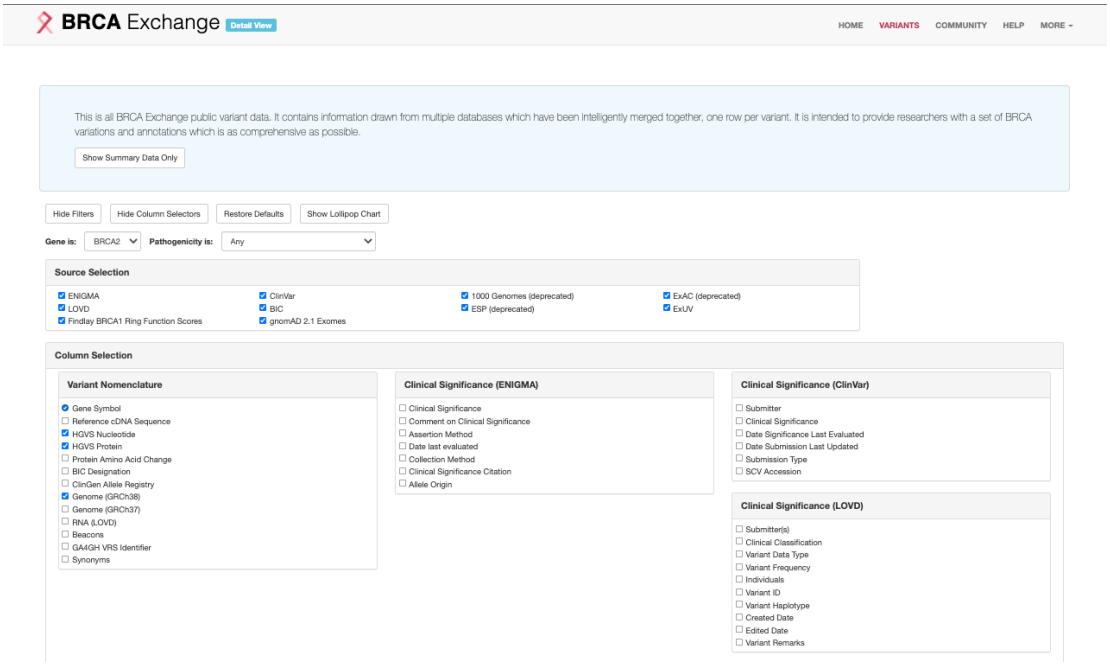


Figure 5.2: A screenshot of the BRCA Exchange website illustrating the column and source filtering and selection web-interface.

An interactive visualization of the variant set was implemented in the form of a lollipop chart. The chart displays for each variant a vertical line oriented by location on the x-axis and a circle at the top of the vertical line representing the class of the variant in the y-axis and by the color of the circle. Figure 5.3 shows an example of the interactive chart.

(a) Lollipop Pathogenicity

(b) Lollipop Allele Frequency

Figure 5.3: A screenshot of the BRCA Exchange website illustrating the interactive lollipop chart. (a) The lollipop chart representing variants by pathogenicity class. (b) The lollipop chart representing variants by allele frequency class.

5.5 Code Availability and Data Access

The main website for BRCA Exchange can be accessed at https://brcaexchange.org/. Data and source code for all of the components of the website can be accessed on github at https://github.com/BRCAChallenge/brca-exchange.

5.6 Acknowledgements

Individuals involved in the BRCA Challenge project include the following: Gaia Andreoletti, Dixie Baker, Steven Brenner, Matthew Brush, Sandrine Caputo, Laurent Castera, Fiona Cunningham, Miguel de la Hoya, Mark Diekhans, Jill Dolinsky, Selina Dwight, Diana Eccles, Bingjian Feng, Marc Fiume, Paul Flicek, Pascale Gaudet, Encarna Gomez Garcia, Melissa Haendel, Max Haeussler, Eric Hahnen, Claude Houdayer, Sarah Hunt, Paul James, Matt Lebo, Jennifer Lee, Jordan Lerner-Ellis, Mike Lin, Steve Lincoln, Adriana Malheiro, Arjen Mesenkamp, Alvaro Monteiro, E. Natzijl-Visser, Joanne Ngeow, Kathryn North, Helen Parkinson, Justin Paschall, George Patrinos, Bette Phimister, Paolo Radice, Irene Rainville, Matthew Rasmussen, George Riley, Etienne Rouleau, Rita Schmutzler, Kent Shefchek, Heidi Sofia, Melissa Southey, J. Stuart, Joseph Thomas, Amanda Toland, Rebecca Truty, Clare Turnbull, Dominique Vaur, Maikke P.G. Vreeswijk, Logan Walker, Michael Walsh, Barbara Wappenschmidt, J. Weitzel, Matt Wright, Vadim Zalunin, Alexander Zaranek, Daniel Zerbino, Alicia Zhou, Jingchun Zhou, and Justin Zook.

The authors Melissa S. Cline, Benedict Paten, David Goldgar, Sean Tavtigian, Gunnar Rätsch, and Amanda B. Spurdle are GA4GH Driver Project Champions for the BRCA Chal-

lenge.

Chapter 6

Co-occurrence Analysis of VUS

6.1 Preamble

In this chapter I present contributions that I've made towards the development of a method for interpreting the pathogenicity of variants of unknown significance (VUS) via the use of co-occuring genotypes in a dataset of samples. These methods have been applied to a study which investigated the set of *BRCA1* and *BRCA2* variants in a cohort of samples that were derrived from the BioBank Japan repository [15]. Here, I describe the pipeline for examining variants that co-occur with other variants within the same sample which implicitly gives some evidence for benign classification for that variant.

6.2 Introduction

One classic example of variants that are known to impact clinical outcomes is the increased risk of cancer in breast, ovarian, pancreatic, prostate and skin tissues that is caused by the presence of pathogenic variants in the *BRCA1* and *BRCA2* genes[24]. Genetic testing is typically done to examine and determine if a patient or their family have inherited risk factors pertaining to those clinical outcomes. Patients and family members can use this information to practice mitigation efforts like increased cancer screening and risk-reducing medication in order to detect and treat cancer before it becomes unmanageable[24]. However, these strategies are limited to the knowledge of which variants in an individual are known to be pathogenic or benign. If a large portion of these variants are identified as VUS, then the individual will be unable to determine what their medical needs are. Many databases containing information about the pathogenic classification of variants house many germline variants that are labeled as variants of unknown significance (VUS). As of May 2021, ClinVar, one of the major repositories for clinical classification of germline variants, has as many as 34.3% of *BRCA1* and *BRCA2* variants are labeled as VUS while another 4.8% of variants have classifications that conflict between different institutions. Currently in other variant databases like gnomAD, many more variants have been identified in individuals but have no clinical classification tied to them[54].

The issue of the accumulation of VUS variants in these databases is related to the phenomenon that many VUS are rare variants in which institutions that aggregate these variants don't obtain enough sample data to determine significance. However, through the use of variant-level summaries on aggregated data, some of these VUS variants can be reinterpreted to be

likely benign based on information. The American College of Molecular Geneticists (ACMG) have developed standard forms of evidence for the practice of variant interpretation[91]. One of these standards includes the observations of VUS variants *in cis* and *in trans* with known pathogenic variants (PM3 and BP2, depending on the disorder). In this study, we developed analysis workflows which aggregate datasets and look for classifications of tumor pathogenic classification for *BRCA1* and *BRCA2* variants in samples derived from the BioBank Japan cohort[74].

6.3 Results

One of the analysis workflows that we have developed looks at examining variants that co-occour with other variants within the same sample. In fully penetrant genetic diseases where the pattern of inheritance is recessive, if an individual without the disease phenotype contains a VUS *in trans*, or on the opposite copy of the gene, with a variant of known pathogenicity within the locus boundary of the same gene, then that observation indicates evidence that the VUS has benign impact. An example is for *BRCA2* (and more recently *BRCA1*), where individuals that express Fanconi Anemia show co-occurrences of two pathogenic variants in both copies of the gene. Fanconi Anemia is a rare disorder that's marked by deficient homologous DNA repair activity, early onset of cancer, bone marrow failure, and a life expectancy of approximately 40 years[8]. In this example, if an individual who is older than 40 does not show signs of Fanconi Anemia and displays a VUS homozygous or heterozygous genotype *in trans* with a pathogenic heterozygous genotype within the same *BRCA2* or *BRCA1* gene locus boundaries,

then the VUS variant has strong evidence for benign classification. Figure 6.1 demonstrates various configurations of genotypes that can indicate benign classification of VUS in a recessive disorder.

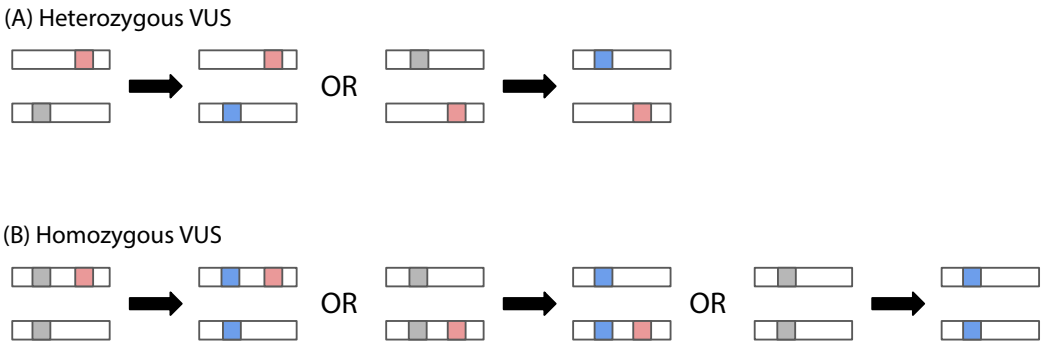


Figure 6.1: An overview of the logic applied to co-occurrence analysis when evaluating variant pathogenicity. Grey boxes represent VUS variants. Red boxes represent pathogenic variants. Blue boxes represent benign variants. White boxes represent gene locus boundaries. (A) Example of heterozygous genotype configurations that contribute to evidence of benign classification for a VUS. (B) Examples of homozygous genotype configurations that contribute to evidence of benign classification for a VUS.

One issue with this analysis is that, for heterozygous genotypes, it is dependent on knowing the phase of the genotypes in order to determine *cis* or *trans* configurations of the genotypes. In this case, genotypes can be phased a number of ways. The use of long read technology can capture the orientation of neighboring alleles which can determine which copy

of the gene a particular allele in a genotype resides. One can also use statistical methods that use genetic linkage to probabilistically determine the likely phase of two neighboring alleles. Finally, if family or offspring information is available then Mendelian inheritance information can be used to determine which parents a set of alleles likely came from which can inform which copy of the gene a specific allele comes from.

Another problem of variant comparison comes from the issue of variant representation. There are instances where two haplotypes can be represented using two different sets of genotypes as illustrated in Figure 6.2. Each pair of genotypes are valid representations, but they are not directly comparable since they represent different positions and variant types in the haplotypes. To overcome this we decomposed the set of variants into a general, haplotype-independent notation and then expand that notation consistently back into an analyzable format. The Human Genome Variation Society (HGVS) have laid out the framework for doing this by using relative notation. We applied HGVS python software libraries to do this by projecting a variant from genomic coordinate space to cDNA-relative coordinate space by choosing a transcript pertaining to the gene that that variant is found. Then we projected that variant back to genomic coordinate space to get a standardized representation of that variant. Processing all variants in this way gives us the ability to directly compare variants within and across datasets.

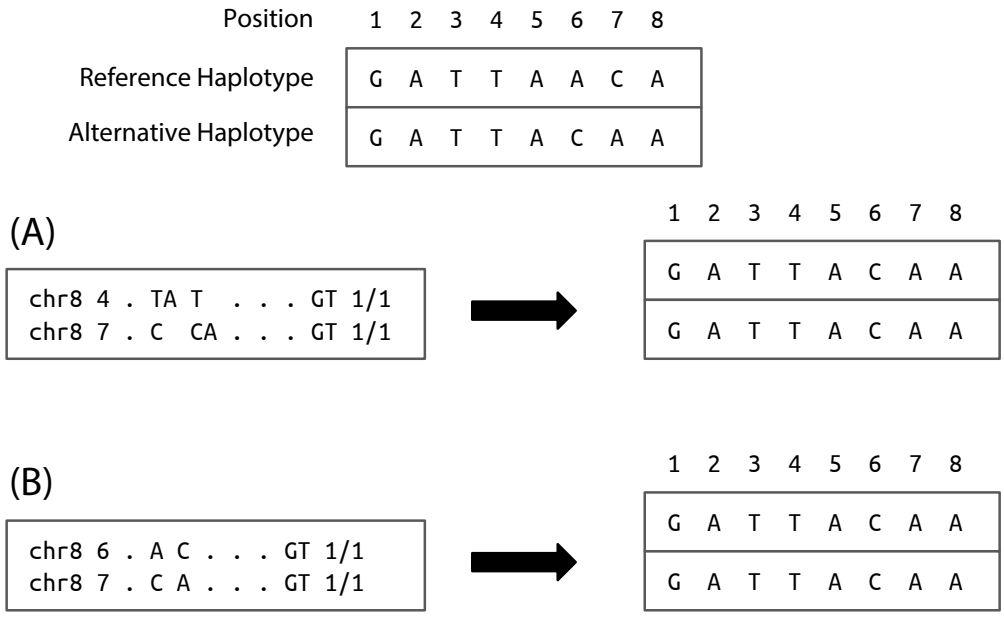


Figure 6.2: An example of a pair of haplotypes where the variant representation in VCF format can be ambiguous. (A) A pair of INDEL genotypes produces two alternative haplotypes. (B) A pair of SNP genotypes produces the same pair of alternative haplotypes.

We applied this method to the BioBank Japan cohort of 23,731 Japanese individuals and found 19 *BRCA* variants that were previously unexamined by the *BRCA* ClinGene Variant Curation Expert Panel (VCEP) expert panel to have evidence of benign classification. Of the 19 VUS variants, 5 variants had benign evidence supported by co-occourrence analysis. Two variants were present in a homozygous genotype and had a single co-occurrence with a pathogenic variant in control individuals. This evidence along with additional HGMD BS1 and BP4 evidence based on allele frequency and BayesDel[31] prediction of benign impact, the data strongly suggests a benign classification for these two variants. The other 3 variants each

had a single heterozygous co-occourrence with a pathogenic variant and have BayesDel prediction scores that indicate benign impact (BP4). However, these 3 variants are heterozygous and the dataset is not phased, so there's not enough evidence from the co-occurrence alone to support benign classification, so further information from additional cohorts would be needed to disambiguate the finding.

6.4 Code Availability

The workflows are available at Dockstore at https://dockstore.org/workflows/github.com/BRCAChallenge/federated-analysis/cooccurrence:master, and the source code is available on GitHub at https://github.com/BRCAChallenge/federated-analysis.

6.5 Acknowledgements

We gratefully thank Gunnar Rätsch for instigating this project, and the members of the BRCA Challenge Evidence Gathering Group for discussion on the analytical design.

Part V

Discussion

Chapter 7

Discussion

For the past 10 years, the field of genomics has been focused in enhancing the ability to detect rare and more complex alleles that have not been sufficiently captured with the more traditional methods of linear-based approaches. Pangenomics offers many avenues towards helping to reduce the reference-bias that has been present in linear-based sequence alignment methods. With the development of pangenomes came the ability to tailor and curate references that enhance the mappability of sequences with more specific origins. This, in-turn, has aided the detection of variants that are unique at the individual level. The flexibility and interoperability of pangenomes also enables the application of pedigree information which can be used to generate parental graph references.

The work presented in this dissertation presents a number of key findings. In chapter 3, I detailed the optimization and performance enhancement of pangenome references and mappers and demonstrated how they have improved variant detection accuracy when compared to standard linear-based approaches. In chapter 4, I presented methods that I developed which

incorporate the tools developed in chapter 3 towards a unified pipeline that generates parental graph references. In that chapter I also demonstrated the application of this workflow towards the processing of real patient datasets and have shown that they recapitulate key variants that were previously discovered in those samples. In chapter 5 and 6, I have described projects that I have been involved with which further illustrate some use-cases for discovery of such variants which can be applied towards clinically-actionable decisions in patients.

The work here can also be easily extended to use multiple pangenomes that each center around a different landscape of variation. Structural variants are an important type of rare variants that contributes to disease [119, 4]. In previous work, there have been efforts to tailor pangenome graphs and variant caller algorithms to improve the accuracy of detecting structural variants [102]. These efforts can easily be merged into the pipelines and methods employed in this dissertation to more exhaustively cover a larger portion of the known pool of rare variation. Multiple pangenomes can also be constructed that focus on either a set of variants in a specific panel of genes [29]. This is a viable direction when abundant haploblock data is available only within specific genes or regions of the genomes from a particular study or database. Different sub-population datasets with allele frequency information that are derived from sufficiently large and random samplesets can be used to generate different population pangenome references that better match the likely allele content of an individual. A sample with an identified common ancestry with a subpopulation can make use of the more specific subpopulation pangenome in order to further avoid the possibility of spurious mapping [16]. Combining this with an alignment-free method of examining read sequences could help determine which subpopulation graph reference would make a more optimal starting reference for the initial alignments

in VG-Pedigree. This method has the significant drawback of assuming perfect subpopulation stratification of humans when the reality is that there are innumerable admixture events which can change the allele frequency landscape overtime. The method also assumes that the subpopulations as defined by population databases like the 1000 Genomes Project accurately define the diversity of human populations. It is for those reasons that a unified pangenome reference would give the most agnostic approach to tackling the reference-bias issue.

A number of things could be done to further improve the variant calling technologies used in this dissertation. One is an extension of Google's DeepVariant variant caller whereby the software could call variants directly from alignments made to the pangenome reference. Currently DeepVariant can only call variants from alignment data that is represented in a linear format like the BAM files that are output from BWA-MEM. In order to adapt pangenome alignments to be callable by DeepVariant, one would need to project the alignment data into linear space. This is a lossy process where the specific information about which haplotype path a particular read mapped to can be lost. Another improvement to DeepVariant would be the implementation of a pedigree-aware variant caller. Currently DeepTrio implements a simpler approach whereby a parent-parent-child trio set of alignments are stacked together and a set of image files are generated. The variant caller does not currently take into account any probabilistic models of de-novo variation and so any Mendelian error, where the genotypes of the trio don't match any expected inheritance models, will not be handeled by the variant caller. This can be remedied by first detecting Mendelian errors and then use a bayesian model that's parameterized by the alignment and basequality information of the trio to determine the most likely set of genotypes that correct the error. The algorithm can then process each de-novo variant at

random until the number of de-novo variants in the child have reached the expected number as determined by the baseline or region-specific mutation rate.

Complementing the developments presented in this dissertation, the field of genomics over the last 10 years has been adding a number of new tools and technologies. Long read sequencing is one technology which has matured to a point where the accuracy and length of the sequenced reads can resolve variants that reside within the variable and repetitive sequence regions of the genome. Circular consensus sequences (CCS), PacBio long-reads and Nanopore sequencing have all helped in filling in the missing gaps of the genome that were present since the initial draft of the human genome was completed in 2003[50, 52, 120]. These sequencing technologies helped to expand the resolution of the centromeric and telomeric regions of the human genome reference. The Telomere-to-telomere project (T2T) has recently published work on completing a human genome sequence which spans these regions[80]. In adapting the T2T results to pangenomes, the Human Pangenome Reference Consortium (HPRC) was formed to develop a pangenome reference using the latest methods and data. By characterizing regions of the genome not well represented by existing variant datasets, the pangenome references developed by the HPRC that incorporate new T2T sequences should further improve the performance and accuracy of the workflows presented in this dissertation.

The current trend in pangenomics development has primarily been focused on first predicting possible use-cases and not on centering the development around the needs of real-world case-studies. Moving forward, I believe some of the more impactful engineering efforts will be done by first identifying specific genetic phenomenon and clinical cases and then to rework the characteristics of pangenomes around accurately detecting those genetic features.

Though there is room for improvement within pangenomics, I am hopeful that the technologies developed here can serve as a solid foundation for future discoveries in genetics research and medicine. I am confident that the expansion of pangenomic research will become more and more relevant and useful to the field of clinical genetics.

Appendix A

Appendix A: Supplementary Information for the VG Giraffe paper

A.1 Preamble

This appendix covers supplementary figures and tables included in the preprint "Genotyping common, large structural variations in 5,202 genomes using pangenomes, the Giraffe mapper, and the vg toolkit" as described in chapter 3.

A.2 Supplementary Figures

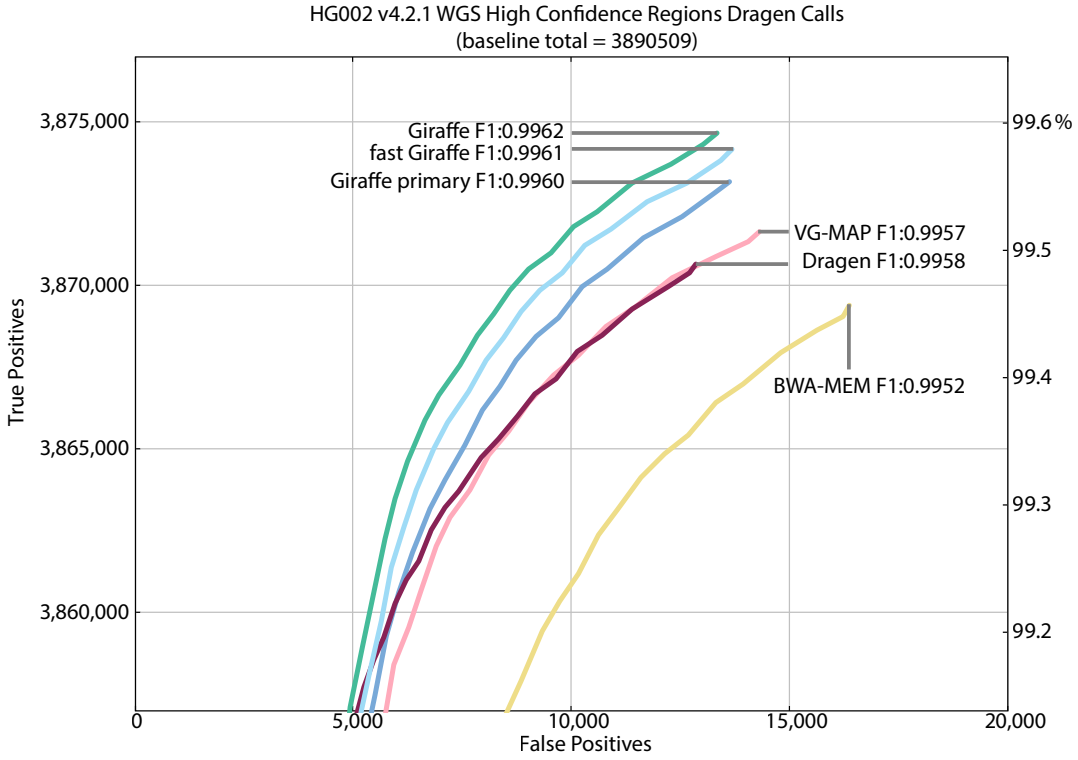


Figure A.1: True positive and false positive genotypes made using the Dragen genotyper with projected mappings from Giraffe and other mappers, using 250bp paired-end reads from the HG002 GIAB sample and evaluated against the HG002 GIAB v4.2.1 truth variant call sets in high confidence regions. The ROC curve discrimination threshold is based on variant call quality.

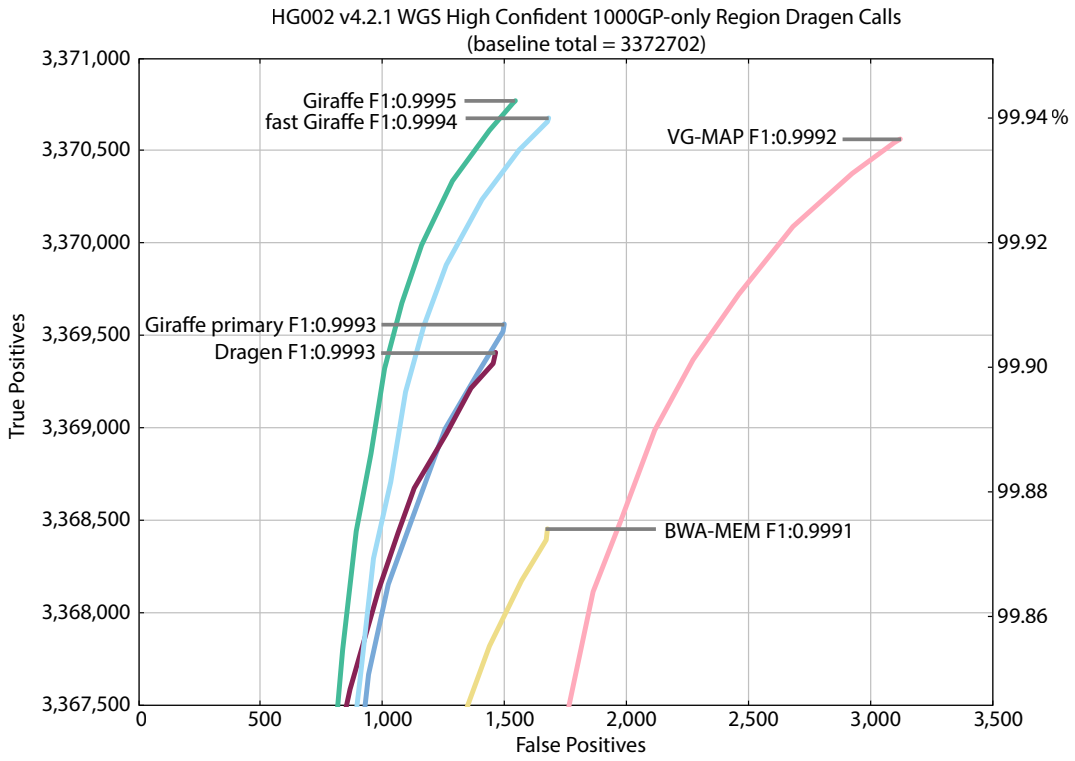


Figure A.2: True positive and false positive genotypes made using the Dragen genotyper with projected mappings from Giraffe and other mappers, using 150bp paired-end reads from the HG002 GIAB sample and evaluated against the HG002 GIAB v4.2.1 truth variant call sets in high confidence regions only within 1000GP variant regions. The ROC curve discrimination threshold is based on variant call quality.

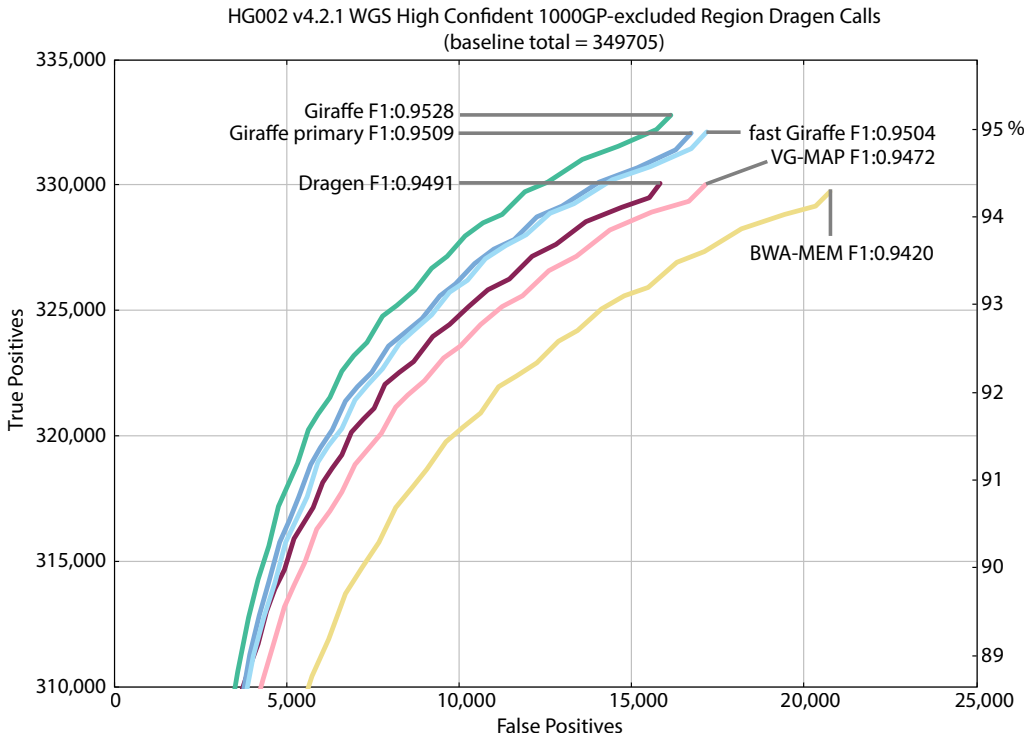


Figure A.3: True positive and false positive genotypes made using the Dragen genotyper with projected mappings from Giraffe and other mappers, using 150bp paired-end reads from the HG002 GIAB sample and evaluated against the HG002 GIAB v4.2.1 truth variant call sets in high confidence regions with 1000GP variant regions excluded. The ROC curve discrimination threshold is based on variant call quality.

A.3 Supplementary Tables

Pipeline	TP	FP	FN	Precision	Sensitivity	F-measure
BWA-MEM	3,869,299	16,380	21,134	0.9958	0.9946	0.9952
DRAGEN	3,870,566	**12,863**	19,864	**0.9967**	0.9949	0.9958
VG-MAP	3,871,583	14,332	18,859	0.9963	0.9952	0.9957
Giraffe primary	3,873,118	13,640	17,331	0.9965	0.9955	0.9960
Giraffe	**3,874,597**	13,359	**15,840**	0.9966	**0.9959**	**0.9962**
fast Giraffe	3,874,104	13,695	16,334	0.9965	0.9958	0.9961

Table A.1: VCFeval performance of linear and graph-based pipelines against grch38-based references using 250bp paired reads with respect to HG002 GIAB v4.2.1 truth variant call sets in high confidence regions. Best values in each column are highlighted in bold text.

Pipeline	Var Type	TP	FN	FP	Recall	Precision	F1
BWA-MEM	INDELS	522,705	2,764	2,214	0.994740	0.995955	0.995347
	SNPS	3,346,810	18,317	14,146	0.994557	0.995792	0.995174
DRAGEN	INDELS	522,672	2,797	**2,043**	0.994677	**0.996265**	0.995471
	SNPS	3,348,116	17,011	**10,799**	0.994945	**0.996786**	0.995865
VG-MAP	INDELS	522,905	2,564	2,260	0.995121	0.995873	0.995497
	SNPS	3,348,892	16,235	12,052	0.995176	0.996415	0.995795
Giraffe primary	INDELS	522,834	2,635	2,316	0.994985	0.995771	0.995378
	SNPS	3,350,496	14,631	11,308	0.995652	0.996637	0.996144
Giraffe	INDELS	**523,148**	**2,321**	2,177	**0.995583**	0.996026	**0.995804**
	SNPS	**3,351,672**	**13,455**	11,167	**0.996002**	0.996680	**0.996341**
fast Giraffe	INDELS	523,119	2,350	2,178	0.995528	0.996024	0.995776
	SNPS	3,351,207	13,920	11,503	0.995863	0.996580	0.996222

Table A.2: Hap.py performance of linear and graph-based pipelines against grch38-based references using 250bp paired reads with respect to HG002 GIAB v4.2.1 truth variant call sets in high confidence regions. Best values in each column are highlighted in bold text.

Pipeline	TP	FP	FN	Precision	Sensitivity	F-measure
BWA-MEM	3,370,702	1,678	4,248	0.9995	0.9987	0.9991
DRAGEN	3,371,710	**1,462**	3,296	**0.9996**	0.9990	0.9993
VG-MAP	3,372,845	3,121	2,141	0.9991	**0.9994**	0.9992
Giraffe primary	3,371,806	1,501	3,143	**0.9996**	0.9991	0.9993
Giraffe	**3,373,044**	1,543	**1,934**	0.9995	**0.9994**	**0.9995**
fast Giraffe	3,372,947	1,684	2,028	0.9995	**0.9994**	0.9994

Table A.3: VCFeval performance of linear and graph-based pipelines against grch38-based references using 150bp paired-end reads with respect to HG002 GIAB v4.2.1 truth variant call sets in high confidence regions only found in the 1000GP variant set used in constructing the graph references used by VG-MAP and Giraffe. Best values in each column are highlighted in bold text.

Pipeline	TP	FP	FN	Precision	Sensitivity	F-measure
BWA-MEM	331,019	20,738	19,972	0.9410	0.9429	0.9420
DRAGEN	331,381	**15,819**	19,643	**0.9544**	0.9438	0.9491
VG-MAP	331,322	17,144	19,694	0.9508	0.9437	0.9472
Giraffe primary	333,362	16,714	17,638	0.9523	0.9496	0.9509
Giraffe	**334,122**	16,144	**16,901**	0.9539	**0.9517**	**0.9528**
fast Giraffe	333,452	17,182	17,569	0.9510	0.9498	0.9504

Table A.4: VCFeval performance of linear and graph-based pipelines against grch38-based references using 150bp paired-end reads with respect to HG002 GIAB v4.2.1 truth variant call sets in high confidence regions excluding the 1000GP variant set used in constructing the graph references used by VG-MAP and Giraffe. Best values in each column are highlighted in bold text.

Mapper	Reads	Graph	vg version	Docker	xg	gcsa	lcp	gbwt	min	gg	dist
VG-MAP	150 bp	1000GP	v1.31.0	Quay							
VG-MAP	250 bp	1000GP	v1.31.0	Quay							
Giraffe	150 bp	Primary	v1.31.0	Quay							
Giraffe	250 bp	Primary	v1.31.0	Quay							
Giraffe	150 bp	1000GP	v1.31.0	Quay							
Giraffe	250 bp	1000GP	v1.31.0	Quay							
Fast Giraffe	150 bp	1000GP	v1.31.0	Quay							
Fast Giraffe	250 bp	1000GP	v1.31.0	Quay							

Table A.5: Table of parameters for GRCh38-based genotyping experiment vg runs

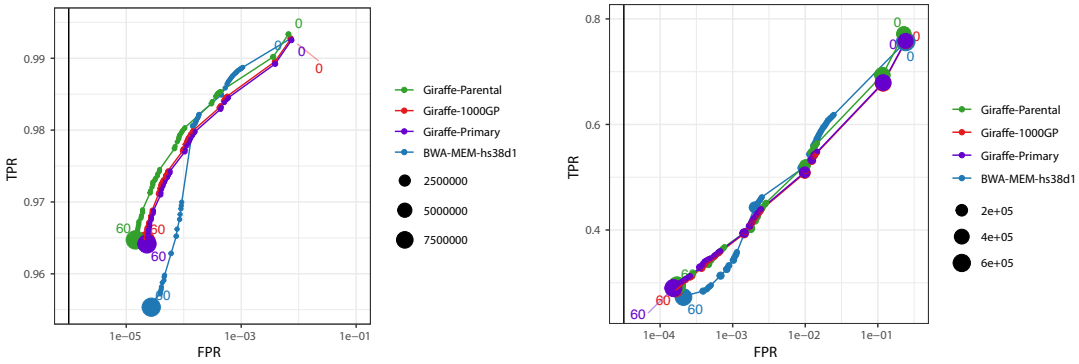
Appendix B

Appendix B: Supplementary Information for the VG-Pedigree paper

B.1 Preamble

This appendix covers supplementary figures and tables included in the preprint ”A Complete Pedigree-Based Graph Workflow for Rare Candidate Variant Analysis” as described in chapter 4.

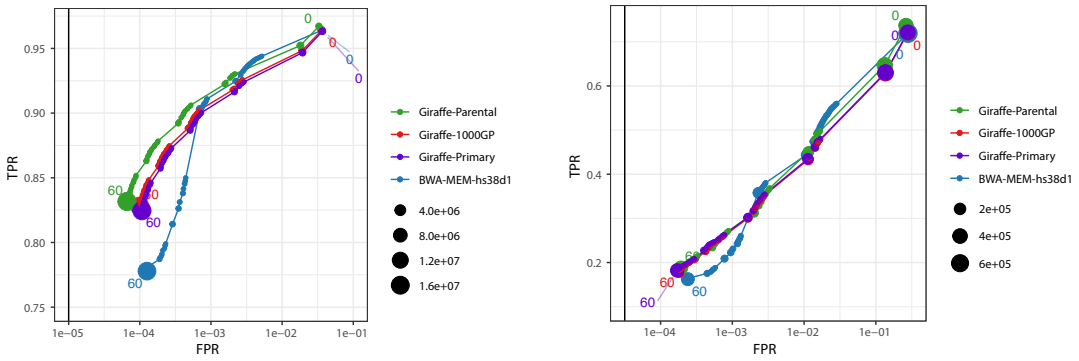
B.2 Supplementary Figures



(a) High Confidence Regions

(b) Non-1000GP Regions

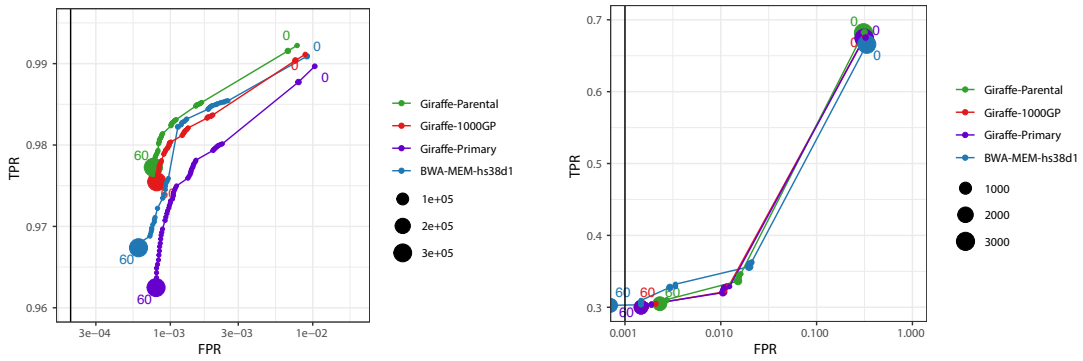
Figure B.1: **Low Complexity Region mapeval HG002** Mapping performance of 10 million read pairs simulated from HG002 high confident datasets. Four different alignments are compared across two different regions and ROC curves are plotted with a log-scaled false positive rate on the x-axis and a linear-scaled true positive rate on the y-axis with mapping quality as the discriminating factor. Green curves represent graph alignments against the parental graph reference constructed from HG003 and HG004 illumina read graph alignments. Red curves represent alignments against the snp1kg graph reference. Purple curves represent alignments to the primary GRCh38 linear graph reference. Blue curves represent linear alignments against the hs38d1 reference using BWA-MEM-hs38d1. (A) Alignments in GIAB v4.2.1 confident regions. (B) Alignments in non-1000GP confident regions.



(a) High Confidence Regions

(b) Non-1000GP Regions

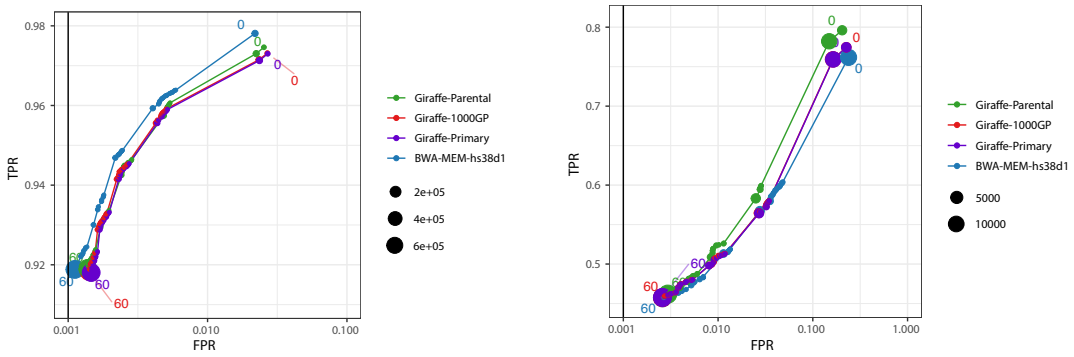
Figure B.2: **Low Mappability Region mapeval HG002** Mapping performance of 100 million read pairs simulated from HG002 high confident datasets. Four different alignments are compared across two different regions and ROC curves are plotted with a log-scaled false positive rate on the x-axis and a linear-scaled true positive rate on the y-axis with mapping quality as the discriminating factor. Green curves represent graph alignments against the parental graph reference constructed from HG003 and HG004 illumina read graph alignments. Red curves represent alignments against the snp1kg graph reference. Purple curves represent alignments to the primary GRCh38 linear graph reference. Blue curves represent linear alignments against the hs38d1 reference using BWA-MEM-hs38d1. (A) Alignments in GIAB v4.2.1 confident regions. (B) Alignments in non-1000GP confident regions.



(a) High Confidence Regions

(b) Non-1000GP Regions

Figure B.3: **MHC Region mapeval HG002** Mapping performance of 100 million read pairs simulated from HG002 high confident datasets. Four different alignments are compared across two different regions and ROC curves are plotted with a log-scaled false positive rate on the x-axis and a linear-scaled true positive rate on the y-axis with mapping quality as the discriminating factor. Green curves represent graph alignments against the parental graph reference constructed from HG003 and HG004 illumina read graph alignments. Red curves represent alignments against the snp1kg graph reference. Purple curves represent alignments to the primary GRCh38 linear graph reference. Blue curves represent linear alignments against the hs38d1 reference using BWA-MEM-hs38d1. (A) Alignments in GIAB v4.2.1 confident regions. (B) Alignments in non-1000GP confident regions.



(a) High Confidence Regions

(b) Non-1000GP Regions

Figure B.4: **CMRG Region mapeval HG002**Mapping performance of 100 million read pairs simulated from HG002 high confident datasets. Four different alignments are compared across two different regions and ROC curves are plotted with a log-scaled false positive rate on the x-axis and a linear-scaled true positive rate on the y-axis with mapping quality as the discriminating factor. Green curves represent graph alignments against the parental graph reference constructed from HG003 and HG004 illumina read graph alignments. Red curves represent alignments against the snp1kg graph reference. Purple curves represent alignments to the primary GRCh38 linear graph reference. Blue curves represent linear alignments against the hs38d1 reference using BWA-MEM-hs38d1. (A) Alignments in GIAB v4.2.1 confident regions. (B) Alignments in non-1000GP confident regions.

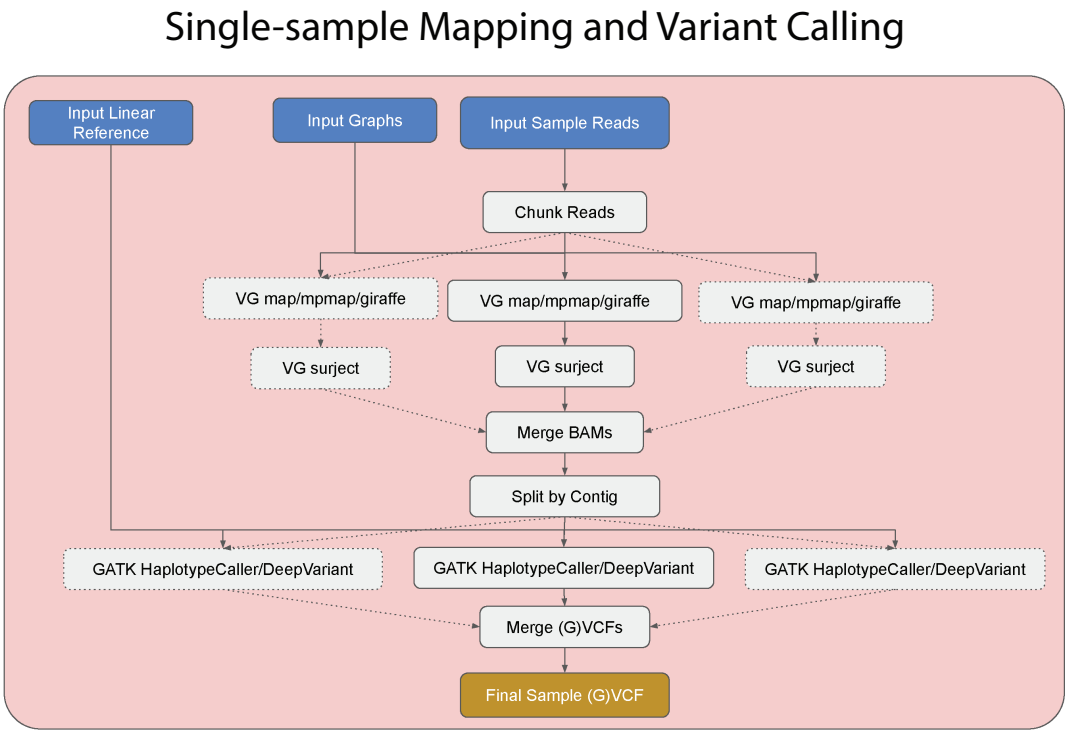


Figure B.5: Single Sample Mapping and Variant Calling Workflow

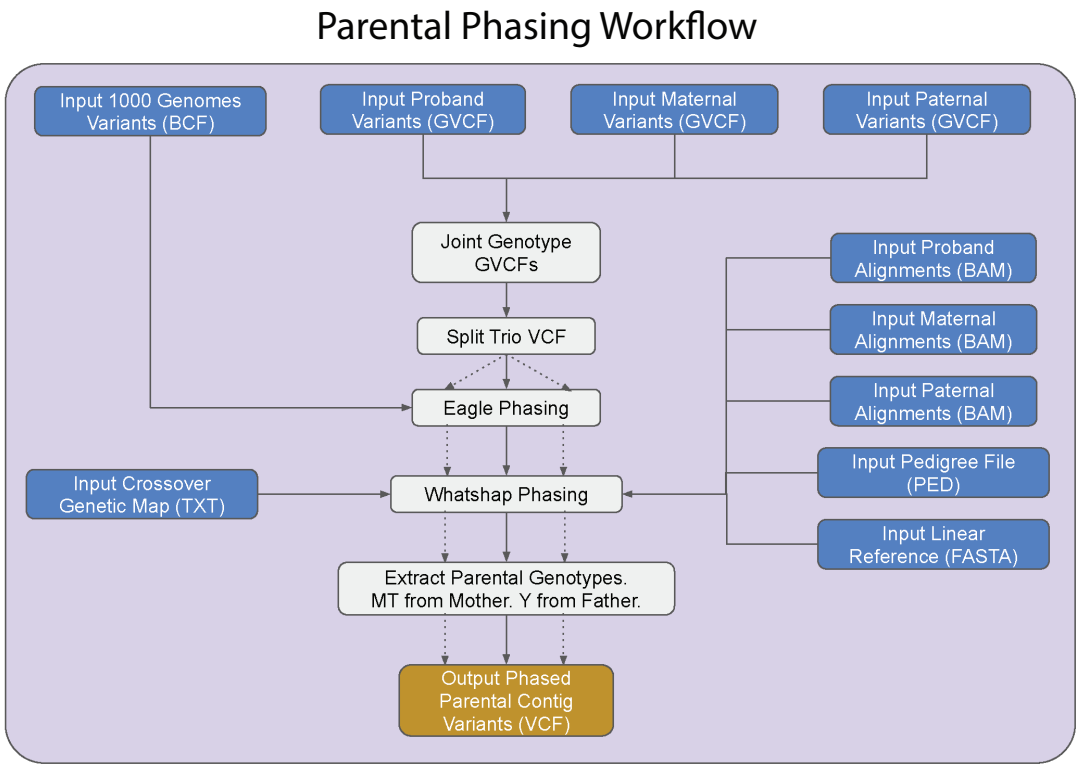


Figure B.6: Parental Phasing Workflow

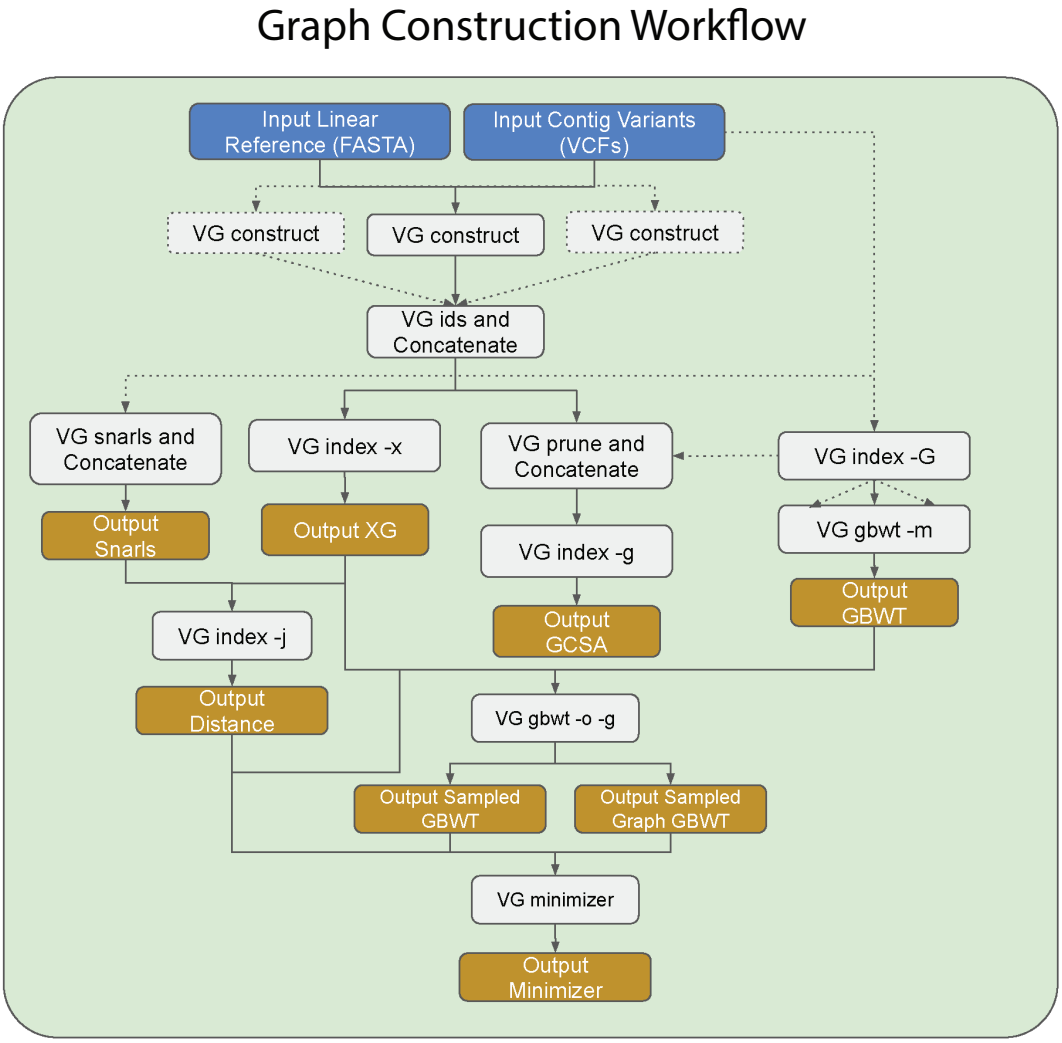


Figure B.7: Graph Construction Workflow

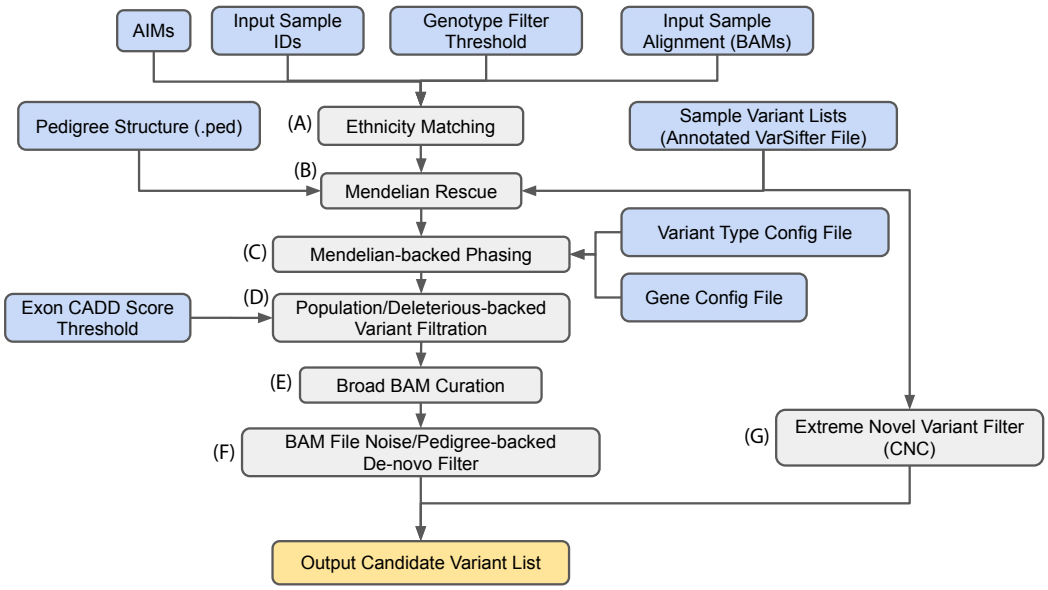


Figure B.8: Pedigree Analysis Workflow

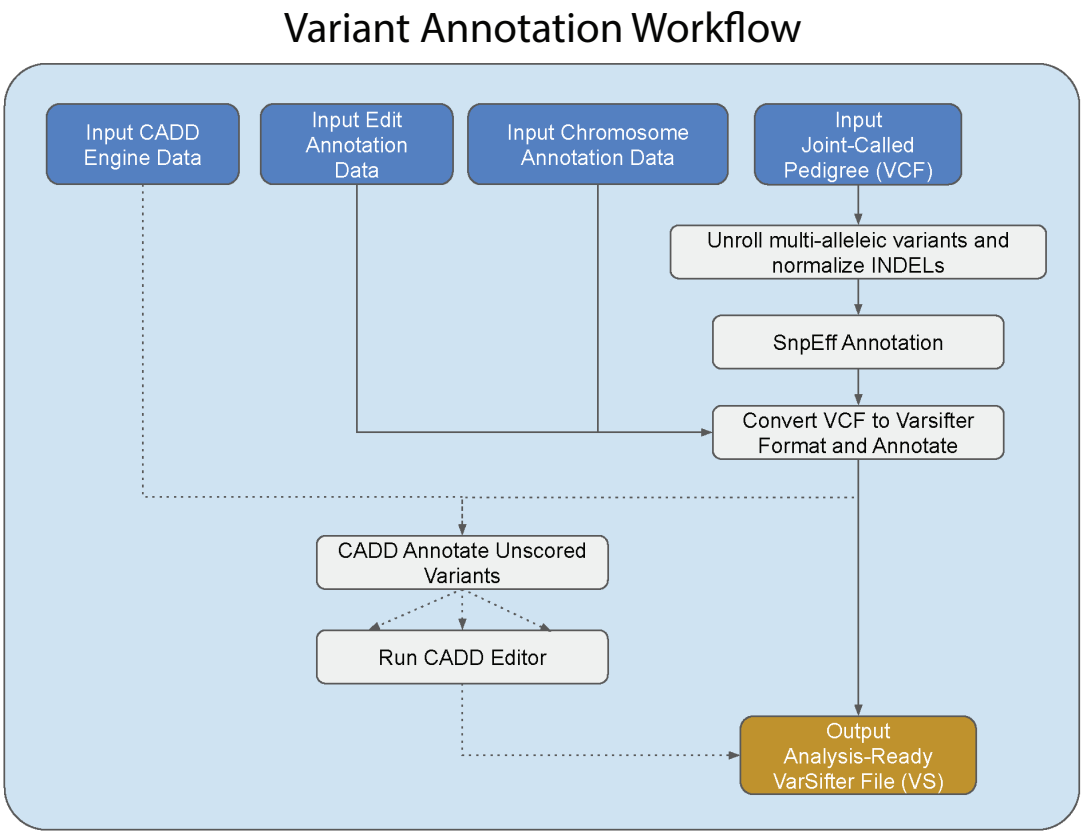


Figure B.9: Variant Annotation Workflow

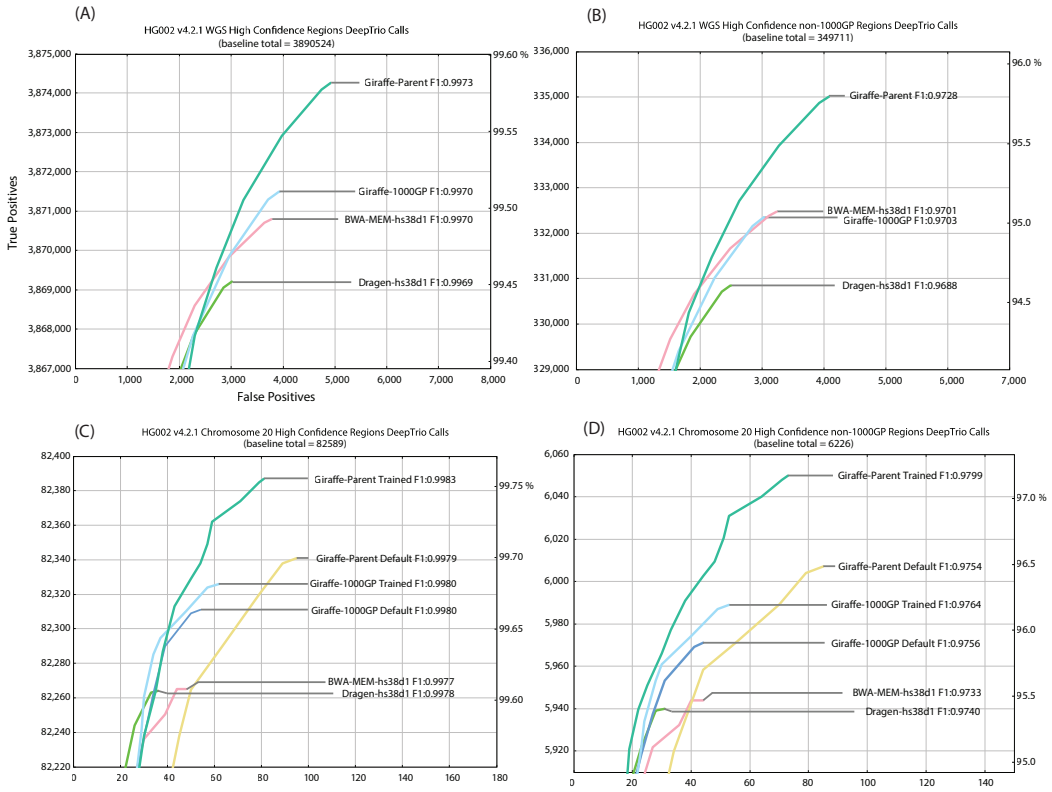


Figure B.10: ROC curves of DeepTrio variant calling performance of the graph-based and linear-based pipelines with respect to HG002 GIAB v4.2.1 truth variant call sets stratified by (A) HG002 high confident whole genome regions using default deeptrio models, (B) HG002 high confident whole genome regions excluding 1000GP variants using default deeptrio models, (C) HG002 high confident chr20 regions on default and trained deeptrio models, (D) HG002 high confident chr20 regions excluding 1000GP on default and trained deeptrio models.

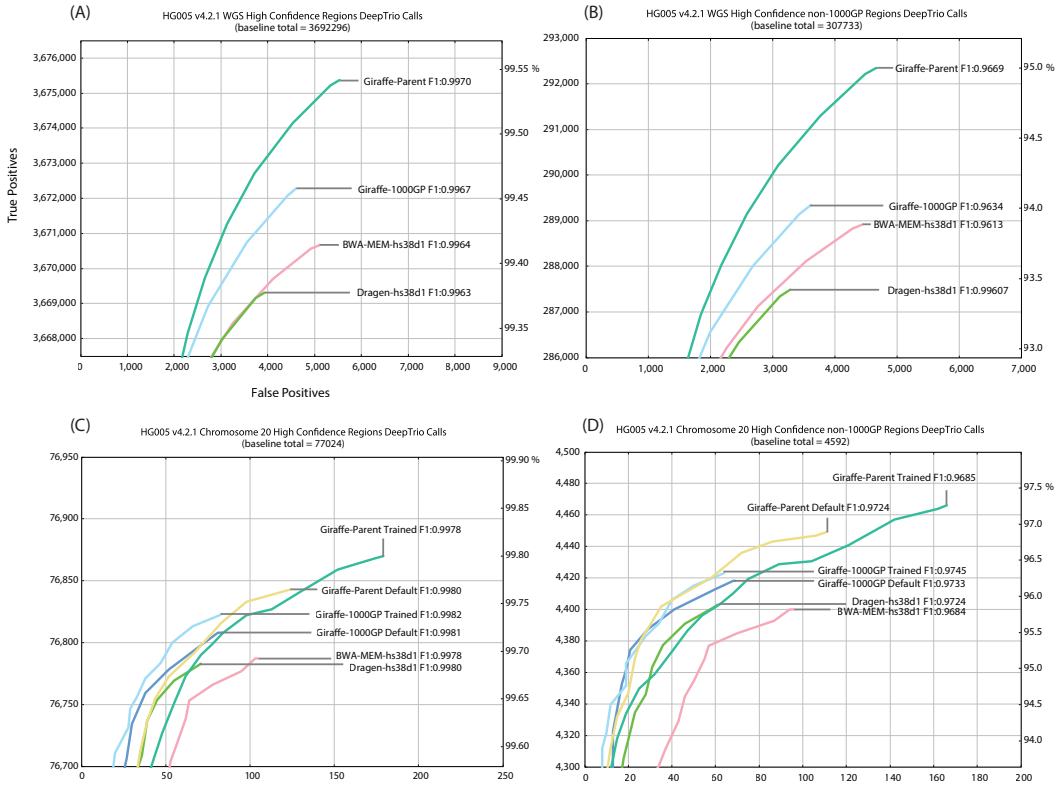


Figure B.11: ROC curves of DeepTrio variant calling performance of the graph-based and linear-based pipelines with respect to HG005 GIAB v4.2 truth variant call sets stratified by (A) high confident HG005 regions using default deeptrio models, (B) HG005 regions excluding 1000GP variants using default deeptrio models, (C) HG005 chr20 regions on default and trained deeptrio models, and (D) HG005 chr20 regions exclusing 1000GP on default and trained deeptrio models.

B.3 Supplementary Tables

Pipeline	Total Correct	Total at MAPQ 60	% Correct at MAPQ 60	% Identity
BWA-MEM-hs38d1	18,114,687	17,733,384	97.89506	NA
Giraffe-Primary	18,109,463	17,808,858	98.34006	99.5028
Giraffe-1000GP	18,111,426	**17,813,218**	**98.35348**	99.5997
Giraffe-Parent	**18,118,011**	17,811,602	98.30881	**99.6076**

(a) High Confidence Regions

Pipeline	Total Correct	Total at MAPQ 60	% Correct at MAPQ 60	% Identity
BWA-MEM-hs38d1	9,988,015	9,612,067	96.23601	NA
Giraffe-Primary	9,985,663	9,700,694	97.14622	99.5002
Giraffe-1000GP	9,987,485	9,704,710	**97.16871**	99.6002
Giraffe-Parent	**9,994,079**	**9,706,146**	97.11896	**99.6173**

(b) All Difficult Regions

Pipeline	Total Correct	Total at MAPQ 60	% Correct at MAPQ 60	% Identity
BWA-MEM-hs38d1	19,358,236	15,621,865	99.98379	NA
Giraffe-Primary	19,341,252	16,559,558	99.98702	99.3660
Giraffe-1000GP	19,357,906	16,628,206	99.98794	99.4458
Giraffe-Parent	**19,419,540**	**16,702,174**	**99.99207**	**99.4830**

(c) Low Mappability Regions

Pipeline	Total Correct	Total at MAPQ 60	% Correct at MAPQ 60	% Identity
BWA-MEM-hs38d1	318,157	310,796	**99.93822**	NA
Giraffe-Primary	317,762	309,288	99.91755	99.0593
Giraffe-1000GP	318,232	313,465	99.91769	99.4125
Giraffe-Parent	**318,583**	**314,020**	99.92230	**99.4847**

(d) MHC Regions

Pipeline	Total Correct	Total at MAPQ 60	% Correct at MAPQ 60	% Identity
BWA-MEM-hs38d1	**831,328**	781,909	**99.87850**	NA
Giraffe-Primary	826,982	781,539	99.84095	99.3621
Giraffe-1000GP	827,118	781,745	99.84675	99.4776
Giraffe-Parent	828,373	**782,353**	99.85007	**99.5284**

(e) CMRG Regions

Table B.1: Mapping statistics on HG002 simulated read alignments against various GRCh38-based graph references and BWA-MEM-hs38d1 alignments against the linear GRCh38 reference.

Pipeline	Total Correct	Total at MAPQ 60	% Correct at MAPQ 60	% Identity
BWA-MEM-hs38d1	2,030,502	1,061,413	99.95949	NA
Giraffe-Primary	2,033,842	1,095,548	**99.97116**	98.8193
Giraffe-1000GP	2,032,683	1,094,500	99.96958	98.8106
Giraffe-Parent	**2,060,518**	**1,104,532**	99.96813	**98.8738**

(a) High Confidence Regions

Pipeline	Total Correct	Total at MAPQ 60	% Correct at MAPQ 60	% Identity
BWA-MEM-hs38d1	1,515,808	546,811	99.92319	NA
Giraffe-Primary	1,518,070	580,904	**99.94750**	98.6305
Giraffe-1000GP	1,516,914	579,978	99.94431	98.6120
Giraffe-Parent	**1,544,713**	**590,796**	99.94211	**98.6816**

(b) All Difficult Regions

Pipeline	Total Correct	Total at MAPQ 60	% Correct at MAPQ 60	% Identity
BWA-MEM-hs38d1	1,249,470	282,325	99.85265	NA
Giraffe-Primary	1,253,071	316,560	**99.90649**	98.4929
Giraffe-1000GP	1,251,903	315,668	99.90085	98.4690
Giraffe-Parent	**1,279,696**	**326,888**	99.89782	**98.5353**

(c) Low Mappability Regions

Pipeline	Total Correct	Total at MAPQ 60	% Correct at MAPQ 60	% Identity
BWA-MEM-hs38d1	3,168	1,440	**100.00000**	NA
Giraffe-Primary	3,213	1,434	99.51185	99.3212
Giraffe-1000GP	3,219	1,436	99.51253	99.3531
Giraffe-Parent	**3,250**	**1,462**	99.24761	**99.3903**

(d) MHC Regions

Pipeline	Total Correct	Total at MAPQ 60	% Correct at MAPQ 60	% Identity
BWA-MEM-hs38d1	23,831	14,387	99.43699	NA
Giraffe-Primary	24,229	14,400	**99.43750**	99.1547
Giraffe-1000GP	24,231	14,390	99.43711	99.1596
Giraffe-Parent	**24,905**	**14,540**	99.36726	**99.2803**

(e) CMRG Regions

Table B.2: Mapping statistics on HG002 simulated read alignments against various GRCh38-based graph references and BWA-MEM-hs38d1 alignments against the linear GRCh38 reference within non-1000GP variant dataset regions.

Pipeline	% Total Aligned	% Total Perfect	% Total Gapless	Total Softclips
Giraffe-Parent	**99.9487**	**59.9905**	**95.6780**	**91185 bp in 5180 read events**
Giraffe-1000GP	99.9484	59.5811	95.4316	132915 bp in 8348 read events
Giraffe-Primary	99.9484	52.4840	94.1270	229332 bp in 16041 read events

(a) High Confidence Regions

Pipeline	% Total Aligned	% Total Perfect	% Total Gapless	Total Softclips
Giraffe-Parent	99.9582	**59.9252**	**95.7237**	**53191 bp in 2769 read events**
Giraffe-1000GP	99.9577	59.0901	95.1374	97954 bp in 5946 read events
Giraffe-Primary	**99.9588**	51.8425	93.3032	163696 bp in 11340 read events

(b) All Difficult Regions

Pipeline	% Total Aligned	% Total Perfect	% Total Gapless	Total Softclips
Giraffe-Parent	99.8358	**59.3039**	**95.4556**	**219196 bp in 8690 read events**
Giraffe-1000GP	99.8339	57.0887	95.1969	291307 bp in 13304 read events
Giraffe-Primary	**99.8394**	51.2567	94.2947	402816 bp in 20980 read events

(c) Low Mappability Regions

Pipeline	% Total Aligned	% Total Perfect	% Total Gapless	Total Softclips
Giraffe-Parent	**99.9651**	**58.9368**	**95.5042**	**58373 bp in 1170 read events**
Giraffe-1000GP	99.9639	57.1014	93.9220	69204 bp in 1650 read events
Giraffe-Primary	99.9402	45.3226	91.3136	100793 bp in 2951 read events

(d) MHC Regions

Pipeline	% Total Aligned	% Total Perfect	% Total Gapless	Total Softclips
Giraffe-Parent	**99.9487**	**59.9619**	**95.5591**	**94701 bp in 2144 read events**
Giraffe-1000GP	99.9485	58.0663	94.1551	117680 bp in 3253 read events
Giraffe-Primary	99.9482	51.0891	92.6229	133603 bp in 4128 read events

(e) CMRG Regions

Table B.3: GAM statistics on HG002 simulated read alignments against various GRCh38-based graph references.

Pipeline	% Total Aligned	% Total Perfect	% Total Gapless	Total Softclips
Giraffe-Parent	99.2513	**57.9923**	**94.7110**	**58129 bp in 2075 read events**
Giraffe-1000GP	99.2456	54.3073	94.4219	69967 bp in 2810 read events
Giraffe-Primary	**99.2685**	53.4428	94.2612	75430 bp in 3136 read events

(a) High Confidence Regions

Pipeline	% Total Aligned	% Total Perfect	% Total Gapless	Total Softclips
Giraffe-Parent	99.0653	**57.6372**	**94.6650**	**51799 bp in 1710 read events**
Giraffe-1000GP	99.0583	53.5654	94.2688	61319 bp in 2324 read events
Giraffe-Primary	**99.0870**	53.0669	94.2073	65488 bp in 2537 read events

(b) All Difficult Regions

Pipeline	% Total Aligned	% Total Perfect	% Total Gapless	Total Softclips
Giraffe-Parent	98.9241	**57.3295**	**94.5397**	**49098 bp in 1547 read events**
Giraffe-1000GP	98.9160	53.5318	94.3970	54949 bp in 1919 read events
Giraffe-Primary	**98.9492**	53.1273	94.3692	58048 bp in 2094 read events

(c) Low Mappability Regions

Pipeline	% Total Aligned	% Total Perfect	% Total Gapless	Total Softclips
Giraffe-Parent	**99.7268**	**60.7608**	**95.4813**	3 bp in 1 read events
Giraffe-1000GP	**99.7268**	57.7764	94.5776	**0 bp in 0 read events**
Giraffe-Primary	**99.7268**	55.9899	94.3464	53 bp in 5 read events

(d) MHC Regions

Pipeline	% Total Aligned	% Total Perfect	% Total Gapless	Total Softclips
Giraffe-Parent	**99.6867**	**59.2507**	**94.7993**	**2187 bp in 65 read events**
Giraffe-1000GP	99.6803	53.0399	92.5010	3090 bp in 139 read events
Giraffe-Primary	**99.6867**	52.5860	92.2932	3255 bp in 143 read events

(e) CMRG Regions

Table B.4: GAM statistics on HG002 simulated read alignments against various GRCh38-based graph references in regions that don't overlap the 1000GP variant set.

Pipeline	Var Type	TP	FN	FP	Recall	Precision	F1
HG002	INDELS	11,208	48	**18**	0.995736	**0.998456**	**0.997094**
	SNPS	71,064	269	28	0.996229	0.999606	0.997915
HG003	INDELS	10,582	46	**19**	0.995672	**0.998282**	**0.996975**
	SNPS	69,928	238	49	0.996608	0.999300	0.997952
HG004	INDELS	**10,950**	**50**	**25**	**0.995455**	**0.997819**	**0.996635**
	SNPS	**71,426**	**233**	47	0.996748	0.999343	**0.998044**

(a) Default

Pipeline	Var Type	TP	FN	FP	Recall	Precision	F1
HG002	INDELS	**11,210**	**46**	25	**0.995913**	0.997857	0.996884
	SNPS	**71,099**	**234**	28	**0.996720**	**0.999607**	**0.998161**
HG003	INDELS	**10,585**	**43**	37	**0.995954**	0.996662	0.996308
	SNPS	**69,958**	**208**	**48**	**0.997036**	**0.999315**	**0.998174**
HG004	INDELS	10,948	51	34	0.995273	0.997036	0.996154
	SNPS	71,405	254	**42**	**0.996455**	**0.999413**	0.997932

(b) Trained

Table B.5: **DeepTrio Trained High Confident Regions Chromosome 20 HG002 Trio** Hap.py performance of DeepTrio using the (A) default DeepTrio model and (B) trained DeepTrio model against the 1000 Genomes Project grch38-based graph reference using 150bp paired-end reads with respect to HG002 GIAB v4.2.1 truth variant call sets in high confidence regions on chromosome 20. Best values for each sample are highlighted in bold text.

Pipeline	Var Type	TP	FN	FP	Recall	Precision	F1
HG005	INDELS	**8,430**	**25**	**9**	**0.997043**	**0.998967**	**0.998004**
	SNPS	68,392	205	**49**	0.997012	**0.999284**	0.998147
HG006	INDELS	8,999	108	77	0.988141	0.991736	0.989935
	SNPS	66,086	118	175	0.998218	0.997361	0.997789
HG007	INDELS	9,162	94	71	0.989844	0.992528	0.991184
	SNPS	67,267	88	111	0.998693	0.998353	0.998523

(a) Default

Pipeline	Var Type	TP	FN	FP	Recall	Precision	F1
HG005	INDELS	8,429	26	21	0.996925	0.997592	0.997258
	SNPS	**68,422**	**175**	58	**0.997449**	0.999153	**0.998300**
HG006	INDELS	**9,010**	**97**	**86**	**0.989349**	**0.990801**	**0.990074**
	SNPS	**66,097**	**107**	**62**	**0.998384**	**0.999064**	**0.998724**
HG007	INDELS	**9,183**	**73**	**66**	**0.992113**	**0.993072**	**0.992593**
	SNPS	**67,285**	**70**	**68**	**0.998961**	**0.998991**	**0.998976**

(b) Trained

Table B.6: **DeepTrio Trained High Confident Regions Chromosome 20 HG005 Trio** Hap.py performance of DeepTrio using the (A) default DeepTrio model and (B) trained DeepTrio model against the 1000 Genomes Project grch38-based graph reference using 150bp paired-end reads with respect to HG005 GIAB v4.2.1 truth variant call sets in high confidence regions on chromosome 20. Best values in each column are highlighted in bold text.

Pipeline	TP	FP	FN	Precision	Sensitivity	F1
Giraffe-Parent	**3,874,697**	4,930	**16,260**	0.9987	**0.9958**	**0.9973**
Giraffe-1000GP	3,871,969	3,934	19,021	0.9990	0.9951	0.9970
BWA-MEM-hs38d1	3,871,240	3,790	19,728	0.9990	0.9949	0.9970
Dragen-hs38d1	3,869,647	**3,028**	21,325	**0.9992**	0.9945	0.9969

(a) DeepTrio HG002 All High Confident Regions

Pipeline	TP	FP	FN	Precision	Sensitivity	F1
Giraffe-Parent	**337,167**	4,096	**14,691**	0.9880	**0.9580**	**0.9728**
Giraffe-1000GP	334,524	3,036	17,362	0.9910	0.9504	0.9703
BWA-MEM-hs38d1	334,456	3,239	17,241	0.9904	0.9507	0.9701
Dragen-hs38d1	332,840	**2,496**	18,862	**0.9926**	0.9461	0.9688

(b) DeepTrio HG002 All High Confident Regions, 1000GP excluded

Pipeline	TP	FP	FN	Precision	Sensitivity	F1
Giraffe-Parent	**3,873,547**	5,993	**17,426**	0.9985	**0.9955**	**0.9970**
Giraffe-1000GP	3,868,404	5,514	22,623	0.9986	0.9942	0.9964
BWA-MEM-hs38d1	3,867,786	5,099	23,169	0.9987	0.9940	0.9964
Dragen-hs38d1	3,865,635	**4,227**	25,328	**0.9989**	0.9935	0.9962

(c) DeepVariant HG002 All High Confident Regions

Pipeline	TP	FP	FN	Precision	Sensitivity	F1
Giraffe-Parent	**3,675,759**	5,538	**16,931**	0.9985	**0.9954**	**0.9970**
Giraffe-1000GP	3,672,686	4,614	20,010	0.9987	0.9946	0.9967
BWA-MEM-hs38d1	3,671,038	5,100	21,631	0.9986	0.9941	0.9964
Dragen-hs38d1	3,669,697	**3,933**	22,978	**0.9989**	0.9938	0.9963

(d) HG005 All High Confident Regions

Pipeline	TP	FP	FN	Precision	Sensitivity	F1
Giraffe-Parent	**293,972**	4,673	**15,392**	0.9844	**0.9500**	**0.9669**
Giraffe-1000GP	290,980	3,616	18,390	0.9877	0.9402	0.9634
BWA-MEM-hs38d1	290,333	4,451	18,816	0.9849	0.9389	0.9613
Dragen-hs38d1	288,921	**3,291**	20,242	**0.9887**	0.9342	0.9607

(e) HG005 All High Confident Regions, 1000GP excluded

Pipeline	TP	FP	FN	Precision	Sensitivity	F1
Giraffe-Parent	**3,674,849**	7,372	**17,866**	0.9980	**0.9952**	**0.9966**
Giraffe-1000GP	3,669,103	7,534	23,650	0.9980	0.9936	0.9958
BWA-MEM-hs38d1	3,667,163	7,728	25,519	0.9979	0.9931	0.9955
Dragen-hs38d1	3,665,339	**6,266**	27,352	**0.9983**	0.9926	0.9954

(f) DeepVariant HG005 All High Confident Regions

Table B.7: **VCFeval HG002 and HG005 DeepTrio and DeepVariant Performance** VCFeval performance of the graph-based and linear-based pipelines with respect to HG002 and HG005 GIAB v4.2.1 truth variant call sets stratified by (A) DeepTrio on all HG002 regions, (B) DeepTrio on HG002 regions excluding 1000GP variants, (C) DeepVariant on all HG002 regions (D) DeepTrio on all HG005 regions, (E) DeepTrio on HG005 regions excluding 1000GP variants, and (F) DeepVariant on all HG005 regions. All mapped reads were called using DeepTrio and DeepVariant v1.1.0 genotyper using default models. Best values in each column are highlighted in bold text.

Pipeline	TP	FP	FN	Precision	Sensitivity	F1
Giraffe-Parent	**981,231**	4,798	**15,946**	0.9951	**0.9840**	**0.9895**
Giraffe-1000GP	978,541	3,795	18,672	0.9961	0.9813	0.9886
BWA-MEM-hs38d1	977,936	3,641	19,243	0.9963	0.9807	0.9884
Dragen-hs38d1	976,334	**2,877**	20,850	**0.9971**	0.9791	0.9880

(a) Low Complexity Regions

Pipeline	TP	FP	FN	Precision	Sensitivity	F1
Giraffe-Parent	**262,599**	3,550	**14,203**	0.9867	**0.9487**	**0.9673**
Giraffe-1000GP	259,901	2,596	16,907	0.9901	0.9389	0.9638
BWA-MEM-hs38d1	259,050	2,774	17,749	0.9894	0.9359	0.9619
Dragen-hs38d1	257,425	**2,025**	19,379	**0.9922**	0.9300	0.9601

(b) Low Mappability Regions

Pipeline	TP	FP	FN	Precision	Sensitivity	F1
Giraffe-Parent	**21,521**	106	**318**	0.9951	**0.9854**	**0.9902**
Giraffe-1000GP	21,489	136	350	0.9937	0.9840	0.9888
BWA-MEM-hs38d1	21,478	107	356	0.9950	0.9837	0.9893
Dragen-hs38d1	21,500	**100**	334	**0.9954**	0.9847	0.9900

(c) MHC Regions

Pipeline	TP	FP	FN	Precision	Sensitivity	F1
Giraffe-Parent	**20,400**	246	**832**	0.9881	**0.9608**	**0.9743**
Giraffe-1000GP	20,375	237	862	0.9885	0.9594	0.9737
BWA-MEM-hs38d1	20,361	243	877	0.9882	0.9587	0.9732
Dragen-hs38d1	20,362	**224**	874	**0.9891**	0.9588	0.9737

(d) Hard-to-sequence Medically Relevant Genes

Table B.8: **VCFeval HG002 DeepTrio Performance Difficult** VCFeval performance of the graph-based and linear-based pipelines with respect to HG002 GIAB v4.2.1 truth variant call sets stratified by (A) low complexity and highly repetative regions, (B) low mappability regions, (C) MHC regions, and (D) Difficult Medically Relevant Genes. All mapped reads were called using DeepTrio v1.1.0 genotyper. Best values in each column are highlighted in bold text.

Pipeline	TP	FP	FN	Precision	Sensitivity	F1
Giraffe-Parent	**836,745**	5,227	**16,495**	0.9938	**0.9807**	**0.9872**
Giraffe-1000GP	833,695	4,320	19,552	0.9948	0.9771	0.9859
BWA-MEM-hs38d1	832,268	4,795	20,935	0.9943	0.9755	0.9848
Dragen-hs38d1	830,857	**3,644**	22,352	**0.9956**	0.9738	0.9846

(a) Low Complexity Regions

Pipeline	TP	FP	FN	Precision	Sensitivity	F1
Giraffe-Parent	**260,810**	4,321	**15,212**	0.9837	**0.9449**	**0.9639**
Giraffe-1000GP	257,763	3,414	18,259	0.9869	0.9338	0.9596
BWA-MEM-hs38d1	256,456	4,101	19,558	0.9843	0.9291	0.9559
Dragen-hs38d1	254,978	**2,967**	21,040	**0.9885**	0.9237	0.9550

(b) Low Mappability Regions

Pipeline	TP	FP	FN	Precision	Sensitivity	F1
Giraffe-Parent	**19,671**	191	**898**	0.9904	**0.9564**	**0.9731**
Giraffe-1000GP	19,653	237	910	0.9881	0.9558	0.9717
BWA-MEM-hs38d1	19,407	178	1,159	0.9909	0.9437	0.9667
Dragen-hs38d1	19,461	**166**	1,101	**0.9915**	0.9465	0.9685

(c) MHC Regions

Table B.9: **VCFeval HG005 DeepTrio Performance Difficult Regions** VCFeval performance of the graph-based and linear-based pipelines with respect to HG005 GIAB v4.2.1 truth variant call sets stratified by (A) low complexity and highly repetative regions, (B) low mappability regions, and (C) MHC regions. All mapped reads were called using DeepTrio v1.1.0 genotyper. Best values in each column are highlighted in bold text.

Pipeline	TP	FP	FN	Precision	Sensitivity	F1
Giraffe-Parent	**3,876,233**	**13,399**	**14,197**	**0.9966**	**0.9964**	**0.9965**
Giraffe-1000GP	3,871,253	17,172	19,183	0.9956	0.9951	0.9953
BWA-MEM-hs38d1	3,866,486	22,395	23,955	0.9942	0.9938	0.9940
Dragen-hs38d1	3,867,357	17,366	23,081	0.9955	0.9941	0.9948

(a) All High Confident Regions

Pipeline	TP	FP	FN	Precision	Sensitivity	F1
Giraffe-Parent	**338,774**	**12,182**	**12,244**	**0.9653**	**0.9650**	**0.9651**
Giraffe-1000GP	334,036	15,583	16,986	0.9554	0.9514	0.9534
BWA-MEM-hs38d1	331,192	20,657	19,836	0.9413	0.9433	0.9423
Dragen-hs38d1	331,387	15,817	19,647	0.9544	0.9438	0.9491

(b) All High Confident Regions, 1000GP excluded

Pipeline	TP	FP	FN	Precision	Sensitivity	F1
Giraffe-Parent	**983,197**	**12,086**	**13,498**	**0.9879**	**0.9865**	**0.9872**
Giraffe-1000GP	978,268	15,735	18,428	0.9842	0.9815	0.9828
BWA-MEM-hs38d1	973,691	20,820	23,006	0.9791	0.9769	0.9780
Dragen-hs38d1	974,330	15,753	22,364	0.9841	0.9776	0.9808

(c) Low Complexity Regions

Pipeline	TP	FP	FN	Precision	Sensitivity	F1
Giraffe-Parent	**265,206**	**10,528**	**11,575**	**0.9618**	**0.9582**	**0.9600**
Giraffe-1000GP	260,558	13,973	16,223	0.9491	0.9414	0.9452
BWA-MEM-hs38d1	255,796	19,064	20,984	0.9306	0.9242	0.9274
Dragen-hs38d1	256,373	13,970	20,407	0.9483	0.9263	0.9372

(d) Low Mappability Regions

Pipeline	TP	FP	FN	Precision	Sensitivity	F1
Giraffe-Parent	**21431**	**113**	**380**	**0.9948**	**0.9826**	**0.9886**
Giraffe-1000GP	21394	235	416	0.9891	0.9810	0.9850
BWA-MEM-hs38d1	21309	312	505	0.9856	0.9769	0.9812
Dragen-hs38d1	21361	316	450	0.9854	0.9794	0.9824

(e) MHC Regions

Pipeline	TP	FP	FN	Precision	Sensitivity	F1
Giraffe-Parent	**20406**	**556**	**817**	**0.9735**	**0.9615**	**0.9675**
Giraffe-1000GP	20319	758	914	0.9640	0.9569	0.9605
BWA-MEM-hs38d1	20278	732	948	0.9652	0.9553	0.9602
Dragen-hs38d1	20310	636	921	0.9696	0.9566	0.9631

(f) Hard-to-sequence Medically Relevant Genes

Table B.10: VCFeval performance of the graph-based and linear-based pipelines with respect to HG002 GIAB v4.2.1 truth variant call sets stratified by (A) all regions, (B) regions excluding 1000GP variants, (C) low complexity and highly repetative regions, (D) low mappability regions, (E) MHC regions, and (F) Difficult Medically Relevant Genes. All mapped reads were called using Illuminas Dragen v3.7.5 genotyper. Best values in each column are highlighted in bold text.

Pipeline	TP	FP	FN	Precision	Sensitivity	F1
Giraffe-Parent	**369,556**	1,280	**4,563**	0.9965	**0.9878**	**0.9922**
Giraffe-1000GP	368,523	1,295	5,619	0.9965	0.9850	0.9907
BWA-MEM-hs38d1	366,035	1,047	8,044	0.9971	0.9785	0.9877
Dragen-hs38d1	366,121	**972**	7,949	**0.9974**	0.9787	0.9880

(a) complex and SVs

Pipeline	TP	FP	FN	Precision	Sensitivity	F1
Giraffe-Parent	**207,109**	313	**2,964**	0.9985	**0.9859**	**0.9922**
Giraffe-1000GP	206,287	275	3,790	0.9987	0.9820	0.9902
BWA-MEM-hs38d1	204,137	295	5,915	0.9986	0.9718	0.9850
Dragen-hs38d1	204,230	**223**	5,818	**0.9989**	0.9723	0.9854

(b) snps within 10bp slop50

Pipeline	TP	FP	FN	Precision	Sensitivity	F1
Giraffe-Parent	**38,740**	141	**761**	0.9964	**0.9807**	**0.9885**
Giraffe-1000GP	38,578	148	924	0.9962	0.9766	0.9863
BWA-MEM-hs38d1	38,105	79	1,378	0.9979	0.9650	0.9812
Dragen-hs38d1	38,153	**77**	1,327	**0.9980**	0.9663	0.9819

(c) complex indel 10bp slop50

Pipeline	TP	FP	FN	Precision	Sensitivity	F1
Giraffe-Parent	**57,488**	**143**	**392**	**0.9975**	**0.9932**	**0.9954**
Giraffe-1000GP	57,362	163	517	0.9972	0.9911	0.9941
BWA-MEM-hs38d1	57,026	163	847	0.9971	0.9854	0.9912
Dragen-hs38d1	57,077	157	795	0.9973	0.9863	0.9917

(d) comphet snp 10bp slop50

Pipeline	TP	FP	FN	Precision	Sensitivity	F1
Giraffe-Parent	47,048	695	781	0.9854	0.9836	0.9845
Giraffe-1000GP	**47,075**	706	**769**	0.9852	**0.9839**	0.9846
BWA-MEM-hs38d1	46,980	604	829	0.9873	0.9826	0.9850
Dragen-hs38d1	46,983	**596**	825	**0.9875**	0.9827	**0.9851**

(e) comphet indel 10bp slop50

Table B.11: VCFeval performance of the graph-based and linear-based pipelines with respect to HG001 GIAB v4.2.1 truth variant call sets stratified by (A) union of all SV, CNV, complex and compound heterozygous variant bed files used for v4.2.1 of the GIAB benchmark for each sample, (B) regions containing at least two variants on one haplotype within 10bp of each other, and all variants are snps, with 50bp slop added on each side, (C) regions containing at least two variants on one haplotype within 10bp of each other, and at least one of the variants is an indel, with 50bp slop added on each side, (D) regions containing at least one variant on each haplotype within 10bp of each other, and all variants are snps, with 50bp slop added on each side and (E) regions containing at least one variant on each haplotype within 10bp of each other, and at least one of the variants is an indel, with 50bp slop added on each side. All mapped reads were called using DeepTrio v1.1.0 genotyper where `Giraffe-Parent` and `Giraffe-1000GP` used the trained model while `BWA-MEM-hs38d1` and `Dragen-hs38d1` used the default model that was tuned to `BWA-MEM` alignments. Best values in each column are highlighted in bold text.

Pipeline	TP	FP	FN	Precision	Sensitivity	F1
Giraffe-Parent	**398,897**	1,254	**5,131**	0.9969	**0.9873**	**0.9921**
Giraffe-1000GP	397,899	1,145	6,139	0.9971	0.9848	0.9909
BWA-MEM-hs38d1	396,630	846	7,419	0.9979	0.9816	0.9897
Dragen-hs38d1	396,600	**809**	7,449	**0.9980**	0.9815	0.9897

(a) complex and SVs

Pipeline	TP	FP	FN	Precision	Sensitivity	F1
Giraffe-Parent	**215,240**	330	**3,208**	0.9985	**0.9853**	**0.9918**
Giraffe-1000GP	214,441	277	4,014	0.9987	0.9816	0.9901
BWA-MEM-hs38d1	213,187	213	5,260	0.9990	0.9759	0.9873
Dragen-hs38d1	213,193	**167**	5,251	**0.9992**	0.9760	0.9875

(b) snps within 10bp slop50

Pipeline	TP	FP	FN	Precision	Sensitivity	F1
Giraffe-Parent	**44,265**	150	**606**	0.9966	**0.9865**	**0.9915**
Giraffe-1000GP	44,134	160	737	0.9964	0.9835	0.9899
BWA-MEM-hs38d1	43,871	112	1006	0.9975	0.9775	0.9874
Dragen-hs38d1	43,903	**110**	969	**0.9975**	0.9784	0.9878

(c) complex indel 10bp slop50

Pipeline	TP	FP	FN	Precision	Sensitivity	F1
Giraffe-Parent	**63,416**	**120**	**410**	**0.9981**	**0.9936**	**0.9958**
Giraffe-1000GP	63,294	133	538	0.9979	0.9916	0.9947
BWA-MEM-hs38d1	63,080	147	745	0.9977	0.9883	0.9930
Dragen-hs38d1	63,111	143	717	0.9977	0.9888	0.9932

(d) comphet snp 10bp slop50

Pipeline	TP	FP	FN	Precision	Sensitivity	F1
Giraffe-Parent	56,912	566	598	0.9902	0.9896	0.9899
Giraffe-1000GP	56,907	541	608	0.9906	0.9894	0.9900
BWA-MEM-hs38d1	**56,994**	**426**	**537**	**0.9926**	**0.9906**	**0.9916**
Dragen-hs38d1	56,992	432	540	0.9925	0.9906	0.9915

(e) comphet indel 10bp slop50

Table B.12: VCFeval performance of the graph-based and linear-based pipelines with respect to HG002 GIAB v4.2.1 truth variant call sets stratified by (A) union of all SV, CNV, complex and compound heterozygous variant bed files used for v4.2.1 of the GIAB benchmark for each sample, (B) regions containing at least two variants on one haplotype within 10bp of each other, and all variants are snps, with 50bp slop added on each side, (C) regions containing at least two variants on one haplotype within 10bp of each other, and at least one of the variants is an indel, with 50bp slop added on each side, (D) regions containing at least one variant on each haplotype within 10bp of each other, and all variants are snps, with 50bp slop added on each side and (E) regions containing at least one variant on each haplotype within 10bp of each other, and at least one of the variants is an indel, with 50bp slop added on each side. All mapped reads were called using the DeepTrio v1.1.0 genotyper default model that was tuned to `BWA-MEM` alignments. Best values in each column are highlighted in bold text.

Pipeline	TP	FP	FN	Precision	Sensitivity	F1
Giraffe-Parent	**357,867**	998	**5,357**	0.9972	**0.9852**	**0.9912**
Giraffe-1000GP	356,744	981	6,466	0.9973	0.9822	0.9897
BWA-MEM-hs38d1	354,731	865	8,488	0.9976	0.9766	0.9870
Dragen-hs38d1	354,796	**800**	8,435	**0.9978**	0.9768	0.9871

(a) complex and SVs

Pipeline	TP	FP	FN	Precision	Sensitivity	F1
Giraffe-Parent	**208,724**	389	**3,739**	0.9981	**0.9824**	**0.9902**
Giraffe-1000GP	207,858	369	4,609	0.9982	0.9783	0.9882
BWA-MEM-hs38d1	206,066	353	6,392	0.9983	0.9699	0.9839
Dragen-hs38d1	206,132	**298**	6,329	**0.9986**	0.9702	0.9842

(b) snps within 10bp slop50

Pipeline	TP	FP	FN	Precision	Sensitivity	F1
Giraffe-Parent	**37,209**	158	**895**	0.9958	**0.9764**	**0.9860**
Giraffe-1000GP	37,034	186	1,058	0.9950	0.9722	0.9834
BWA-MEM-hs38d1	36,637	133	1,470	0.9964	0.9613	0.9785
Dragen-hs38d1	36,736	**119**	1,378	**0.9968**	0.9637	0.9800

(c) complex indel 10bp slop50

Pipeline	TP	FP	FN	Precision	Sensitivity	F1
Giraffe-Parent	**57,398**	**129**	**429**	**0.9978**	**0.9926**	**0.9952**
Giraffe-1000GP	57,313	141	517	0.9975	0.9911	0.9943
BWA-MEM-hs38d1	57,102	142	727	0.9975	0.9874	0.9924
Dragen-hs38d1	57,130	142	700	0.9975	0.9879	0.9927

(d) comphet snp 10bp slop50

Pipeline	TP	FP	FN	Precision	Sensitivity	F1
Giraffe-Parent	35,824	368	444	0.9898	0.9877	0.9888
Giraffe-1000GP	35,821	372	**436**	0.9897	**0.9880**	0.9888
BWA-MEM-hs38d1	35,798	344	460	0.9905	0.9873	0.9889
Dragen-hs38d1	**35,829**	**329**	**436**	**0.9909**	**0.9880**	**0.9894**

(e) comphet indel 10bp slop50

Table B.13: VCFeval performance of the graph-based and linear-based pipelines with respect to HG005 GIAB v4.2.1 truth variant call sets stratified by (A) union of all SV, CNV, complex and compound heterozygous variant bed files used for v4.2.1 of the GIAB benchmark for each sample, (B) regions containing at least two variants on one haplotype within 10bp of each other, and all variants are snps, with 50bp slop added on each side, (C) regions containing at least two variants on one haplotype within 10bp of each other, and at least one of the variants is an indel, with 50bp slop added on each side, (D) regions containing at least one variant on each haplotype within 10bp of each other, and all variants are snps, with 50bp slop added on each side and (E) regions containing at least one variant on each haplotype within 10bp of each other, and at least one of the variants is an indel, with 50bp slop added on each side. All mapped reads were called using DeepTrio v1.1.0 genotyper default model that was tuned to `BWA-MEM` alignments. Best values in each column are highlighted in bold text.

Pipeline	TP	FP	FN	Precision	Sensitivity	F1
Giraffe-Parent	**3,676,960**	**15,137**	**15,271**	**0.9959**	**0.9959**	**0.9959**
Giraffe-1000GP	3,671,554	19,961	20,682	0.9946	0.9944	0.9945
BWA-MEM-hs38d1	3,666,629	25,529	25,603	0.9931	0.9931	0.9931
Dragen-hs38d1	3,667,062	20,746	25,172	0.9944	0.9932	0.9938

(a) All High Confident Regions

Pipeline	TP	FP	FN	Precision	Sensitivity	F1
Giraffe-Parent	**295,692**	**13,440**	**12,950**	**0.9565**	**0.9579**	**0.9572**
Giraffe-1000GP	290,604	17,730	18,041	0.9425	0.9414	0.9419
BWA-MEM-hs38d1	287,788	23,311	20,835	0.9251	0.9323	0.9287
Dragen-hs38d1	287,890	18,300	20,739	0.9402	0.9326	0.9364

(b) All High Confident Regions, 1000GP excluded

Pipeline	TP	FP	FN	Precision	Sensitivity	F1
Giraffe-Parent	**838,794**	**12,502**	**14,055**	**0.9853**	**0.9835**	**0.9844**
Giraffe-1000GP	833,427	17,042	19,426	0.9800	0.9772	0.9786
BWA-MEM-hs38d1	828,706	22,846	24,137	0.9732	0.9717	0.9724
Dragen-hs38d1	829,353	17,211	23,493	0.9797	0.9725	0.9760

(c) Low Complexity Regions

Pipeline	TP	FP	FN	Precision	Sensitivity	F1
Giraffe-Parent	**263,249**	**11,319**	**12,734**	**0.9588**	**0.9538**	**0.9563**
Giraffe-1000GP	258,032	15,679	17,951	0.9427	0.9349	0.9388
BWA-MEM-hs38d1	253,309	21,529	22,667	0.9217	0.9178	0.9197
Dragen-hs38d1	253,923	15,752	22,055	0.9416	0.9201	0.9307

(d) Low Mappability Regions

Pipeline	TP	FP	FN	Precision	Sensitivity	F1
Giraffe-Parent	**19,612**	**184**	**942**	**0.9907**	**0.9542**	**0.9721**
Giraffe-1000GP	19,586	322	973	0.9838	0.9527	0.9680
BWA-MEM-hs38d1	19,365	437	1,190	0.9779	0.9422	0.9597
Dragen-hs38d1	19,429	423	1,128	0.9787	0.9452	0.9616

(e) MHC Regions

Table B.14: VCFeval performance of the graph-based and linear-based pipelines with respect to HG005 GIAB Draft v4.2.1 truth variant call sets stratified by (A) all regions, (B) regions excluding 1000GP variants, (C) low complexity and highly repetative regions, (D) low mappability regions, and (E) MHC regions. All mapped reads were called using Illuminas Dragen v3.7.5 genotyper. Best values in each column are highlighted in bold text.

Pipeline	Var Type	TP	FN	FP	Recall	Precision	F1
BWA-MEM-hs38d1	INDELS	522,588	2,881	2,628	0.994517	0.995202	0.994860
	SNPS	3,344,124	21,003	19,746	0.993759	0.994132	0.993945
Dragen-hs38d1	INDELS	522,618	2,851	2,478	0.994574	0.995475	0.995025
	SNPS	3,344,971	20,156	14,869	0.994010	0.995576	0.994792
Giraffe-1000GP	INDELS	522,857	2,612	2,493	0.995029	0.995450	0.995239
	SNPS	3,348,629	16,498	14,664	0.995097	0.995641	0.995369
Giraffe-Parent	INDELS	**523,367**	**2,102**	**2,150**	**0.996000**	**0.996077**	**0.996039**
	SNPS	**3,353,105**	**12,022**	**11,232**	**0.996427**	**0.996662**	**0.996545**

Table B.15: **All High Confident Regions** Hap.py performance of linear and graph-based pipelines against grch38-based references using 150bp paired-end reads with respect to HG002 GIAB v4.2.1 truth variant call sets in high confidence regions. All mapped reads were called using Illuminas Dragen v3.7.5 genotyper. Best values in each column are highlighted in bold text.

Pipeline	Var Type	TP	FN	FP	Recall	Precision	F1
BWA-MEM-hs38d1	INDELS	127,809	2,136	2,051	0.983562	0.985570	0.984565
	SNPS	206,391	17,741	18,657	0.920846	0.917379	0.919109
Dragen-hs38d1	INDELS	127,822	2,125	1,890	0.983647	0.986689	0.985166
	SNPS	206,576	17,560	13,979	0.921655	0.936836	0.929184
Giraffe-1000GP	INDELS	127,933	2,010	1,976	0.984532	0.986103	0.985317
	SNPS	209,107	15,027	13,667	0.932955	0.938858	0.935897
Giraffe-Parent	INDELS	**128,402**	**1,542**	**1,664**	**0.988133**	**0.988314**	**0.988224**
	SNPS	**213,380**	**10,755**	**10,574**	**0.952016**	**0.952944**	**0.952480**

Table B.16: **All High Confident Regions, 1000GP excluded** Hap.py performance of linear and graph-based pipelines against grch38-based references using 150bp paired-end reads with respect to HG002 GIAB v4.2.1 truth variant call sets in high confidence regions. All mapped reads were called using Illuminas Dragen v3.7.5 genotyper. Best values in each column are highlighted in bold text.

Pipeline	Var Type	TP	FN	FP	Recall	Precision	F1
BWA-MEM-hs38d1	INDELS	366,349	2,793	2,428	0.992434	0.993795	0.993114
	SNPS	607,821	20,142	18,370	0.967925	0.970707	0.969314
Dragen-hs38d1	INDELS	366,372	2,770	2,261	0.992496	0.994220	0.993357
	SNPS	608,443	19,520	13,471	0.968915	0.978371	0.973620
Giraffe-1000GP	INDELS	366,608	2,534	2,267	0.993135	0.994208	0.993671
	SNPS	612,142	15,821	13,452	0.974806	0.978528	0.976664
Giraffe-Parent	INDELS	**367,116**	**2,026**	**1,934**	**0.994512**	**0.995062**	**0.994787**
	SNPS	**616,565**	**11,398**	**10,134**	**0.981849**	**0.983853**	**0.982850**

Table B.17: **High Confident Low Complexity Regions** Hap.py performance of linear and graph-based pipelines against grch38-based references using 150bp paired-end reads with respect to HG002 GIAB v4.2.1 truth variant call sets in high confidence regions. All mapped reads were called using Illuminas Dragen v3.7.5 genotyper. Best values in each column are highlighted in bold text.

Pipeline	Var Type	TP	FN	FP	Recall	Precision	F1
BWA-MEM-hs38d1	INDELS	16,165	1,385	1,216	0.921083	0.931191	0.926109
	SNPS	239,599	19,596	17,853	0.924397	0.930654	0.927515
Dragen-hs38d1	INDELS	16,188	1,362	1,009	0.922393	0.942300	0.932240
	SNPS	240,153	19,042	12,964	0.926534	0.948782	0.937526
Giraffe-1000GP	INDELS	16,496	1,054	1,020	0.939943	0.942722	0.941331
	SNPS	244,030	15,165	12,957	0.941492	0.949580	0.945519
Giraffe-Parent	INDELS	**16,771**	**779**	**789**	**0.955613**	**0.955821**	**0.955717**
	SNPS	**248,401**	**10,794**	**9,740**	**0.958356**	**0.962268**	**0.960308**

Table B.18: **High Confident Low Mappability Regions** Hap.py performance of linear and graph-based pipelines against grch38-based references using 150bp paired-end reads with respect to HG002 GIAB v4.2.1 truth variant call sets in high confidence regions. All mapped reads were called using Illuminas Dragen v3.7.5 genotyper. Best values in each column are highlighted in bold text.

Pipeline	Var Type	TP	FN	FP	Recall	Precision	F1
BWA-MEM-hs38d1	INDELS	1,640	37	28	0.977937	0.984914	0.981413
	SNPS	19,753	424	239	0.978986	0.987936	0.983441
Dragen-hs38d1	INDELS	**1,642**	**35**	29	**0.979129**	0.984467	0.981791
	SNPS	19,809	368	239	0.981761	0.987963	0.984853
Giraffe-1000GP	INDELS	1,641	36	26	0.978533	0.986014	0.982259
	SNPS	19,841	336	164	0.983347	0.991723	0.987517
Giraffe-Parent	INDELS	1,641	36	**17**	0.978533	**0.990826**	**0.984641**
	SNPS	**19,879**	**298**	**49**	**0.985231**	**0.997517**	**0.991336**

Table B.19: **High Confident MHC Regions** Hap.py performance of linear and graph-based pipelines against grch38-based references using 150bp paired-end reads with respect to HG002 GIAB v4.2.1 truth variant call sets in high confidence regions. All mapped reads were called using Illuminas Dragen v3.7.5 genotyper. Best values in each column are highlighted in bold text.

Pipeline	Var Type	TP	FN	FP	Recall	Precision	F1
BWA-MEM-hs38d1	INDELS	3,438	189	139	0.947891	0.963641	0.955701
	SNPS	16,892	707	541	0.959827	0.968781	0.964283
Dragen-hs38d1	INDELS	3,446	181	**121**	0.950096	**0.968291**	**0.959108**
	SNPS	16,911	688	462	0.960907	0.973250	0.967039
Giraffe-1000GP	INDELS	3,442	185	151	0.948994	0.960585	0.954754
	SNPS	16,924	675	554	0.961646	0.968135	0.964880
Giraffe-Parent	INDELS	**3,450**	**177**	128	**0.951199**	0.966492	0.958785
	SNPS	**17,010**	**589**	**377**	**0.966532**	**0.978187**	**0.972325**

Table B.20: **High Confident Hard-to-sequence Medically Relevant Genes** Hap.py performance of linear and graph-based pipelines against grch38-based references using 150bp paired-end reads with respect to HG002 GIAB v4.2.1 truth variant call sets in high confidence regions. All mapped reads were called using Illuminas Dragen v3.7.5 genotyper. Best values in each column are highlighted in bold text.

Pipeline	Var Type	TP	FN	FP	Recall	Precision	F1
BWA-MEM-hs38d1	INDELS	414,524	2,252	2,264	0.994597	0.99474	0.994668
	SNPS	3,252,342	23,289	23,261	0.992890	0.99290	0.992895
Dragen-hs38d1	INDELS	414,506	2,270	2,187	0.994553	0.994918	0.994735
	SNPS	3,252,795	22,836	18,551	0.993029	0.994330	0.993679
Giraffe-1000GP	INDELS	414,726	2,050	2,220	0.995081	0.994844	0.994963
	SNPS	3,257,070	18,561	17,726	0.994334	0.994588	0.994461
Giraffe-Parent	INDELS	**415,126**	**1,650**	**1,844**	**0.996041**	**0.995718**	**0.995879**
	SNPS	**3,262,094**	**13,537**	**13,267**	**0.995867**	**0.995950**	**0.995909**

Table B.21: **All High Confident Regions** Hap.py performance of linear and graph-based pipelines against grch38-based references using 150bp paired-end reads with respect to HG005 GIAB v4.2.1 truth variant call sets in high confidence regions. All mapped reads were called using Illuminas Dragen v3.7.5 genotyper. Best values in each column are highlighted in bold text.

Pipeline	Var Type	TP	FN	FP	Recall	Precision	F1
BWA-MEM-hs38d1	INDELS	87,787	1,725	1,944	0.980729	0.980132	0.98043
	SNPS	201,964	19,131	21,428	0.913472	0.904275	0.90885
Dragen-hs38d1	INDELS	87,779	1,734	1,827	0.980629	0.981303	0.980966
	SNPS	202,069	19,027	16,531	0.913942	0.924536	0.919209
Giraffe-1000GP	INDELS	87,895	1,617	1,857	0.981935	0.981027	0.981481
	SNPS	204,643	16,461	15,944	0.925551	0.927866	0.926707
Giraffe-Parent	INDELS	**88,276**	**1,236**	**1,506**	**0.986192**	**0.984622**	**0.985406**
	SNPS	**209,377**	**11,727**	**11,984**	**0.946962**	**0.945971**	**0.946466**

Table B.22: **All High Confident Regions, 1000GP excluded** Hap.py performance of linear and graph-based pipelines against grch38-based references using 150bp paired-end reads with respect to HG005 GIAB v4.2.1 truth variant call sets in high confidence regions. All mapped reads were called using Illuminas Dragen v3.7.5 genotyper. Best values in each column are highlighted in bold text.

Pipeline	Var Type	TP	FN	FP	Recall	Precision	F1
BWA-MEM-hs38d1	INDELS	261,098	2,140	2,007	0.991870	0.992747	0.992309
	SNPS	568,001	21,935	20,836	0.962818	0.964646	0.963731
Dragen-hs38d1	INDELS	261,094	2,144	1,846	0.991855	0.993325	0.992589
	SNPS	568,653	21,283	15,358	0.963923	0.973726	0.968800
Giraffe-1000GP	INDELS	261,308	1,930	1,853	0.992668	0.993305	0.992986
	SNPS	572,511	17,425	15,174	0.970463	0.974203	0.972329
Giraffe-Parent	INDELS	**261,702**	**1,536**	**1,507**	**0.994165**	**0.994557**	**0.994361**
	SNPS	**577,501**	**12,435**	**10,969**	**0.978921**	**0.981377**	**0.980147**

Table B.23: **High Confident Low Complexity Regions** Hap.py performance of linear and graph-based pipelines against grch38-based references using 150bp paired-end reads with respect to HG005 GIAB v4.2.1 truth variant call sets in high confidence regions. All mapped reads were called using Illuminas Dragen v3.7.5 genotyper. Best values in each column are highlighted in bold text.

Pipeline	Var Type	TP	FN	FP	Recall	Precision	F1
BWA-MEM-hs38d1	INDELS	14,902	1,432	1,345	0.912330	0.918222	0.915267
	SNPS	238,355	21,231	20,193	0.918212	0.921897	0.920051
Dragen-hs38d1	INDELS	14,913	1,421	1,118	0.913004	0.931115	0.921970
	SNPS	238,956	20,630	14,639	0.920527	0.942273	0.931273
Giraffe-1000GP	INDELS	15,195	1,139	1,077	0.930268	0.934596	0.932427
	SNPS	242,776	16,810	14,608	0.935243	0.943244	0.939227
Giraffe-Parent	INDELS	**15,499**	**835**	**773**	**0.948880**	**0.953066**	**0.950968**
	SNPS	**247,692**	**11,894**	**10,551**	**0.954181**	**0.959143**	**0.956656**

Table B.24: **High Confident Low Mappability Regions** Hap.py performance of linear and graph-based pipelines against grch38-based references using 150bp paired-end reads with respect to HG005 GIAB v4.2.1 truth variant call sets in high confidence regions. All mapped reads were called using Illuminas Dragen v3.7.5 genotyper. Best values in each column are highlighted in bold text.

Pipeline	Var Type	TP	FN	FP	Recall	Precision	F1
BWA-MEM-hs38d1	INDELS	1,594	104	74	0.938751	0.960215	0.949362
	SNPS	17,848	1037	319	0.945089	0.982277	0.963324
Dragen-hs38d1	INDELS	1,597	101	81	0.940518	0.956708	0.948544
	SNPS	17,911	974	298	0.948425	0.983483	0.965636
Giraffe-1000GP	INDELS	1,607	91	60	0.946408	0.967759	0.956964
	SNPS	18,062	823	206	0.956420	0.988620	0.972254
Giraffe-Parent	INDELS	**1,613**	**85**	**43**	**0.949941**	**0.976894**	**0.963229**
	SNPS	**18,098**	**787**	**72**	**0.958327**	**0.995998**	**0.976799**

Table B.25: **High Confident MHC Regions** Hap.py performance of linear and graph-based pipelines against grch38-based references using 150bp paired-end reads with respect to HG005 GIAB v4.2.1 truth variant call sets in high confidence regions. All mapped reads were called using Illuminas Dragen v3.7.5 genotyper. Best values in each column are highlighted in bold text.

Pipeline	TP	FP	FN	Precision	Sensitivity	F1
Dragen-Graph-hs38d1	**3,713,377**	14,580	**8,552**	0.9961	**0.9977**	0.9969
Giraffe-1000GP	3,708,607	**5,687**	13,934	**0.9985**	0.9963	0.9974
Giraffe-Parent	3,711,135	6,444	11,258	0.9983	0.9970	**0.9976**

(a) HG001

Pipeline	TP	FP	FN	Precision	Sensitivity	F1
Dragen-Graph-hs38d1	**3,878,846**	11,982	**11,592**	0.9969	**0.9970**	0.9970
Giraffe-1000GP	3,871,969	**3,934**	19,021	**0.9990**	0.9951	0.9970
Giraffe-Parent	3,874,697	4,930	16,260	0.9987	0.9958	**0.9973**

(b) HG002

Pipeline	TP	FP	FN	Precision	Sensitivity	F1
Dragen-Graph-hs38d1	**3,679,064**	13,750	**13,171**	0.9963	**0.9964**	0.9964
Giraffe-1000GP	3,672,686	**4,614**	20,010	**0.9987**	0.9946	0.9967
Giraffe-Parent	3,675,759	5,538	16,931	0.9985	0.9954	**0.9970**

(c) HG005

Table B.26: VCFeval performance of the graph-based Illumina Dragen v3.7.5 results `Dragen-Graph-hs38d1` versus the `Giraffe-Parent` and `Giraffe-1000GP` trained DeepTrio results on HG001, HG002, and HG005 GIAB v4.2.1 truth variant call sets. All `Giraffe-Parent` and `Giraffe-1000GP` mapped reads were called using a trained DeepTrio v1.1.0 genotyper. Best values in each column are highlighted in bold text.

Cohort ID	Sample ID	SNPs(INDELs)	MF	CC	MC	DN	HR	HM	XL	CNC	CM
UDP10618	UDP10618	4596326(1662678)	48397	38	21	5	5	0	15	0	13
	UDP11318	4599172(1662619)	47617	11	11	1	1	1	0	0	8
UDP10815	UDP10815	4670495(1695028)	59070	14	14	3	0	2	0	0	9
	UDP11178	4671755(1695378)	56412	20	20	2	5	0	0	0	13
UDP10898	UDP10898	4598520(1706914)	49839	17	15	3	2	2	0	0	10
	UDP11100	4599169(1707215)	49854	26	14	6	1	0	12	0	7
UDP10915	UDP10915	4594489(1729303)	51032	13	13	2	1	2	0	0	8
	UDP11205	4595214(1729292)	49031	23	12	1	0	1	10	0	11
UDP10934	UDP10934	4596460(1792043)	37260	21	13	0	5	1	4	0	11
	UDP11401	4597116(1792526)	35625	12	11	2	2	1	0	0	7
	UDP11402	4597246(1792142)	34354	21	10	6	6	0	6	0	3
UDP11136	UDP11136	4529866(1687948)	44364	54	11	1	0	41	2	0	10
	UDP11452	4531234(1691523)	43478	11	7	0	4	1	1	0	5
UDP11628	UDP11628	4620199(1780410)	37682	17	14	3	5	0	0	0	9
	UDP12046	4620477(1781250)	37079	14	12	2	8	0	0	0	4
	UDP12047	4620963(1781070)	41392	13	12	3	2	0	0	2	6
UDP11732	UDP11732	4801023(1846737)	60615	42	21	6	3	0	18	0	15
	UDP12102	4801929(1846678)	54942	18	16	5	2	0	0	0	11
	UDP12103	4803220(1846798)	60527	23	21	2	2	1	0	0	18
UDP11854	UDP11854	4579136(1731652)	49485	13	13	7	1	0	0	0	5
	UDP12189	4579971(1732111)	49256	26	15	5	3	1	10	0	7
UDP12531	UDP12531	5103899(1518349)	54023	15	14	6	1	1	0	0	7
	UDP12828	5105572(1518403)	58263	36	12	1	8	1	19	0	7
	UDP12829	5106400(1518434)	53520	22	18	3	6	1	0	0	12
UDP12639	UDP12639	4766953(1422540)	59948	23	13	1	3	0	7	0	12
	UDP18612	4770026(1422158)	57176	17	15	2	3	1	2	0	9
UDP12925	UDP12925	4888275(1447275)	43188	26	22	3	12	0	0	0	11
	UDP18398	4891027(1447124)	47376	27	9	0	4	1	18	0	4
	UDP18399	4889817(1447196)	42447	8	8	2	0	0	0	0	6
UDP6540	UDP6540	4561095(1686264)	46285	12	7	0	1	0	5	0	6
	UDP11422	4562810(1686426)	46657	27	17	3	3	0	9	0	12
UDP6603	UDP6603	4804021(1886900)	49991	24	14	1	3	0	6	0	14
	UDP12029	4804566(1887828)	42704	10	10	4	0	0	0	0	6
	UDP12030	4805402(1887866)	43145	11	11	5	1	0	0	0	5
UDP12283	UDP12283	4654603(1770554)	37784	37	13	4	3	23	0	0	7
	UDP12523	4654881(1774679)	39409	12	11	3	2	0	0	0	7
	UDP12524	4657260(1774522)	36491	11	9	2	5	1	0	0	3

Table B.27: **Proband vs Sibling Analysis 15 Cohort Variant Filter Results** Total counts of starting variants down to the filtered candidate list for each of the 15 cohort samples comprising 15 probands and 22 siblings. Bolded cohort IDs are cohort probands where their diagnosis has an associated CLIA-validated variant by the UDP. MF are the total number of variants after passing through Mendelian filters. CC, the number of candidate variants and compound heterozygous pairs produced by the candidate analysis workflow. MC, the total number of candidate variants and compound heterozygous pairs after merging candidates together that share the same gene locus and the X-linked variant counts removed for sex-independent sample comparison. DN, the number of de-novo candidate variants. HR, the number of homozygous recesive variants. HM are the number of hemizygous variants. XL, the number of X-linked variants. CNC, the number of variants that lie within a depleted or haploinsufficient region. CM, the number of compound heterozygous pairs of variants.

Sample ID	Total	SNPs	INDELs	MF	CC	DN	HR	HM	XL	CNC	CM
UDP18482	5994493	4624043	1370450	81201	39	2	2	1	19	0	15
UDP18111	6228191	4807587	1420604	39220	12	1	3	3	0	0	5
UDN714479	6100463	4702966	1397497	30257	23	0	3	0	14	0	6
UDN833679	6194587	4773584	1421003	46306	23	3	2	3	9	0	6

Table B.28: **Candidate Analysis Filter Variant Counts on 4 Additional Cohort Probands** Total counts of starting variants down to the filtered candidate list for 4 probands. Bolded cohort IDs are cohort probands where their disease was diagnosed by the UDP. MF are the total number of variants after passing through Mendelian filters. CC, the number of candidate variants and compound heterozygous pairs produced by the candidate analysis workflow. DN, the number of de-novo candidate variants. HR, the number of homozygous recesive variants. HM are the number of hemizygous variants. XL, the number of X-linked variants. CNC, the number of variants that lie within a depleted or haploinsufficient region. CM, the number of compound heterozygous pairs of variants.

Sample ID	Type	Total	E	NE	CV(P:B:UNK)	PP(D:B:UNK)	HGMD	$CADD_E20$	$CADD_{NE}15$	VMM30
UDP18482	non-CM	24	2	22	NA:NA:24	NA:NA:24	NA	2	16	NA
	CM Pairs	15	11	4	NA:4:11	5:4:6	NA	NA	NA	15
UDP18111	non-CM	7	0	7	NA:NA:7	NA:NA:7	NA	0	6	NA
	CM Pairs	5	3	2	NA:NA:5	NA:1:4	1	NA	NA	5
UDN714479	non-CM	17	7	10	NA:NA:17	3:1:13	1	5	10	NA
	CM Pairs	6	5	1	NA:1:5	3:1:2	2	NA	NA	6
UDN833679	non-CM	17	6	11	NA:2:15	2:2:13	1	6	11	NA
	CM Pairs	6	4	2	NA:NA:6	3:NA:3	NA	NA	NA	6

Table B.29: **Candidate Analysis Candidate Statistic Counts on 4 Additional Cohort Probands** Total count of variants with various annotation stats. `non-CM`, the set of candidate variants that are not compound heterozygous variants. `CM Pairs`, the set of candidate variants that are compound heterozygous variants.

`E`, counts of exonic variants based on the SnpEff annotation list (includes `NON_SYNONYMOUS_CODING`, `FRAME_SHIFT`, `CODON_CHANGE_PLUS_CODON_DELETION`, `STOP_GAINED`, `START_GAINED`).

`NE`, counts of non-exonic variants based on the SnpEff annotation list (includes anything not in the `Exonic` list).

`CV(P)`, ClinVar variant interpretation of either Pathogenic or Likely Pathogenic.

`CV(B)`, ClinVar variant interpretation of either Benign or Likely Benign.

`CV(UNK)`, ClinVar variant interpretation of either not present in the database or categorized as `not_provided`, `Conflicting_interpretations_of_pathogenicity`, `Uncertain_significance`.

`PP(D)`, PolyPhen variant interpretation of either probably damaging or possibly damaging.

`PP(B)`, PolyPhen variant interpretation as benign.

`PP(UNK)`, PolyPhen variant interpretation non existant in current database.

`HGMD`, Human Gene Mutation Database annotation of Disease Mutation `DM` or possible Disease Mutation `DM?`.

$CADD_E20$, number of exonic variants with a Combined Annotation Dependent Depletion Phred score of greater than or equal to 20.

$CADD_{NE}15$, number of non-exonic variants with a Combined Annotation Dependent Depletion Phred score of greater than or equal to 15.

`VMM30`, number of compound heterozygous pairs with a Virtual Mendelian Model Combined score of greater than or equal to 30.

Workflow Run	Total Runtime	Total CPU Hours	Total Cost
HG002 Trio + HG001 VG Pedigree Workflow	37h 2m	7983.73	$92.32
HG002 Trio + HG001 Canidate Analysis Workflow	7h 56m, 8h 35m	195.09, 213.51	$3.68, $4.10
HG005 Trio + HG001 VG Pedigree Workflow	38h 11m	7062.96	$88.22,
HG005 Trio + HG001 Canidate Analysis Workflow	8h 43m, 9h 15m	213.00, 237.20	$4.07, $4.62

Table B.30: Workflow costs for various runs of the VG Pedigree workflow and the Candidate Analysis workflow on the Terra Platform.

Phasing Method	Total (Heterozygous)	Variants Assessed	SE Rate	SE/F Rate	HD[%]
EAGLE2	**75444**(**47568**)	36853	1.53%	1.01%	39.17%
EAGLE2 + WHATSHAP	**75444**(**47568**)	**38843**	0.90%	0.49%	0.50%
WHATSHAP	**75444**(47549)	38774	0.88%	0.49%	0.49%
WHATSHAP + EAGLE2	**75444**(**47568**)	36835	1.55%	1.03%	39.17%
WHATSHAP + SHAPEIT4	71594(44966)	36850	**0.76%**	**0.41%**	**0.41%**

(a) WhatsHap Compare HG002 GRCh37 Chr20 Intersecting variants

Phasing Method	Total (Heterozygous)	Variants Assessed	SE Rate	SE/F Rate	HD[%]
EAGLE2	83331(**52255**)	36288	0.79%	0.59%	49.26%
EAGLE2 + WHATSHAP	83331(**52255**)	**37188**	**0.07%**	**0.03%**	**0.03%**
WHATSHAP	**83332**(52221)	37103	**0.07%**	0.04%	0.04%
WHATSHAP + EAGLE2	83331(**52255**)	36279	0.87%	0.66%	49.26%
WHATSHAP + SHAPEIT4	80951(50862)	36288	0.08%	0.04%	0.04%

(b) WhatsHap Compare HG002 GRCh38 Chr20 Intersecting variants

Table B.31: WhatsHap Compare statistics on Chromosome 20 joint-called and phased HG002 trio data using various phasing methods. (A) Comparison stats of the Giraffe aligned and DeepVariant joint-genotyped VCF on GRCh37-based reference against the GIAB v3.3.2 benchmark trio-phased VCF. (B) Comparison stats of the Giraffe aligned and DeepVariant joint-genotyped VCF on GRCh38-based liftovered reference against the GIAB v4.2.1 benchmark trio-phased VCF. Abbreviations: SE(Switch-error), SE/F(Switch/Flip-error), HD(Hamming Distance).

Phasing Method	Total Phased (SNVs)	Blocks	Median Block Size	Largest Block	Smallest Block
BASELINE	39965(35042)	401	2	39088	2
EAGLE2	52213(46735)	**1**	52213	52213	52213
EAGLE2 + WHATSHAP	**56858**(**48422**)	429	3	**55121**	2
WHATSHAP	56474(48171)	430	3	54733	2
WHATSHAP + EAGLE2	52194(46717)	405	3	49609	2
WHATSHAP + SHAPEIT4	52216(46735)	**1**	**52216**	52216	**52216**

(a) HG002 GRCh37 Chr20

Phasing Method	Total Phased (SNVs)	Blocks	Median Block Size	Largest Block	Smallest Block
EAGLE2	53274(47628)	**1**	**53274**	53274	**53274**
EAGLE2 + WHATSHAP	**58190**(**49355**)	429	3	**56453**	2
WHATSHAP	57843(49150)	430	3	56102	2
WHATSHAP + EAGLE2	53255(47610)	405	3	50725	2
WHATSHAP + SHAPEIT4	53273(47626)	**1**	53273	53273	53273

(b) HG003 GRCh37 Chr20

Phasing Method	Total Phased (SNVs)	Blocks	Median Block Size	Largest Block	Smallest Block
EAGLE2	54298(48484)	**1**	54298	54298	54298
EAGLE2 + WHATSHAP	**59155**(**50105**)	429	3	**57418**	2
WHATSHAP	58788(49882)	430	3	57047	2
WHATSHAP + EAGLE2	54279(48466)	405	3	51730	2
WHATSHAP + SHAPEIT4	54308(48493)	**1**	**54308**	54308	**54308**

(c) HG004 GRCh37 Chr20

Phasing Method	Total Phased (SNVs)	Blocks	Median Block Size	Largest Block	Smallest Block
BASELINE	37814(33217)	**1**	37814	37814	37814
EAGLE2	57780(49484)	**1**	57780	57780	57780
EAGLE2 + WHATSHAP	**61423**(**52374**)	434	**3**	**59681**	2
WHATSHAP	60669(52049)	445	**3**	58886	2
WHATSHAP + EAGLE2	57773(49478)	436	**3**	54601	2
WHATSHAP + SHAPEIT4	57838(49511)	**1**	**57838**	57838	**57838**

(d) HG002 GRCh38 Chr20

Phasing Method	Total Phased (SNVs)	Blocks	Median Block Size	Largest Block	Smallest Block
EAGLE2	58881(50312)	**1**	58881	58881	58881
EAGLE2 + WHATSHAP	**62573**(**53184**)	434	3	**60831**	2
WHATSHAP	61838(52878)	445	3	60055	2
WHATSHAP + EAGLE2	58874(50306)	436	3	55774	2
WHATSHAP + SHAPEIT4	58889(50316)	**1**	**58889**	58889	**58889**

(e) HG003 GRCh38 Chr20

Phasing Method	Total Phased (SNVs)	Blocks	Median Block Size	Largest Block	Smallest Block
EAGLE2	59258(50497)	**1**	59258	59258	59258
EAGLE2 + WHATSHAP	**62289**(**52726**)	434	3	**60547**	2
WHATSHAP	61546(52430)	445	3	59763	2
WHATSHAP + EAGLE2	59251(50491)	436	3	56147	2
WHATSHAP + SHAPEIT4	59300(50523)	**1**	**59300**	59300	**59300**

(f) HG004 GRCh38 Chr20

Table B.32: WhatsHap phasing stats on Chromosome 20 joint-called and phased HG002 trio data using various phasing methods. (a-c) Stats for HG002, HG003, and HG004 respectively, aligned with Giraffe and DeepVariant trio-based joint-genotyped VCF on GRCh37-based reference. (d-f) Stats for HG002, HG003, and HG004 respectively aligned with Giraffe and DeepVariant trio-based joint-genotyped VCF on GRCh38-based liftovered reference.

Bibliography

[1] HUGO—a UN for the human genome. *Nature Genetics*, 34(2):115–116, June 2003.

[2] AlleleSeq: analysis of allele-specific expression and binding in a network framework. *Molecular Systems Biology*, 7(1):522, January 2011.

[3] Picard toolkit. https://broadinstitute.github.io/picard/, 2019.

[4] Haley J. Abel, David E. Larson, Allison A. Regier, Colby Chiang, Indraniel Das, Krishna L. Kanchi, Ryan M. Layer, Benjamin M. Neale, William J. Salerno, Catherine Reeves, Steven Buyske, Goncalo R. Abecasis, Elizabeth Appelbaum, Julie Baker, Eric Banks, Raphael A. Bernier, Toby Bloom, Michael Boehnke, Eric Boerwinkle, Erwin P. Bottinger, Steven R. Brant, Esteban G. Burchard, Carlos D. Bustamante, Lei Chen, Judy H. Cho, Rajiv Chowdhury, Ryan Christ, Lisa Cook, Matthew Cordes, Laura Courtney, Michael J. Cutler, Mark J. Daly, Scott M. Damrauer, Robert B. Darnell, Tracie Deluca, Huyen Dinh, Harsha Doddapaneni, Evan E. Eichler, Patrick T. Ellinor, Andres M. Estrada, Yossi Farjoun, Adam Felsenfeld, Tatiana Foroud, Nelson B. Freimer, Catrina Fronick, Lucinda Fulton, Robert Fulton, Stacy Gabriel, Liron Ganel, Shailu Gargeya, Goren Germer, Daniel H. Geschwind, Richard A. Gibbs, David B. Goldstein, Megan L. Grove, Namrata Gupta, Christopher A. Haiman, Yi Han, Daniel Howrigan, Jianhong Hu, Carolyn Hutter, Ivan Iossifov, Bo Ji, Lynn B. Jorde, Goo Jun, John Kane, Chul Joo Kang, Hyun Min Kang, Sek Kathiresan, Eimear E. Kenny, Lily Khaira, Ziad Khan, Amit Khera, Charles Kooperberg, Olga Krasheninina, William E. Kraus, Subra Kugathasan, Markku Laakso, Tuuli Lappalainen, Adam E. Locke, Ruth J. F. Loos, Amy Ly, Robert Maier, Tom Maniatis, Loic Le Marchand, Gregory M. Marcus, Richard P. Mayeux, Dermot P. B. McGovern, Karla S. Mendoza, Vipin Menon, Ginger A. Metcalf, Zeineen Momin, Guiseppe Narzisi, Joanne Nelson, Caitlin Nessner, Rodney D. Newberry, Kari E. North, Aarno Palotie, Ulrike Peters, Jennifer Ponce, Clive Pullinger, Aaron Quinlan, Daniel J. Rader, Stephen S. Rich, Samuli Ripatti, Dan M. Roden, Veikko Salomaa, Jireh Santibanez, Svati H. Shah, M. Benjamin Shoemaker, Heidi Sofia, Taylorlyn Stephan, Christine Stevens, Stephan R. Targan, Marja-Riitta Taskinen, Kathleen Tibbetts, Charlotte Tolonen, Tychele Turner, Paul De Vries, Jason Waligorski, Kimberly Walker, Vivian Ota Wang, Michael Wigler, Richard K. Wilson, Lara Winterkorn, Genevieve Wojcik, Jinchuan Xing, Erica Young, Bing Yu, Yeting Zhang, Tara C. Matise, Donna M. Muzny, Michael C. Zody, Eric S. Lander, Susan K. Dutcher, Nathan O. Stitziel, Ira M.

Hall, and NHGRI Centers for Common Disease Genomics. Mapping and characterization of structural variation in 17,795 human genomes. *Nature*, 583(7814):83–89, July 2020.

[5] Rocio Acuna-Hidalgo, Joris A. Veltman, and Alexander Hoischen. New insights into the generation and role of de novo mutations in health and disease. *Genome Biology*, 17(1):241, November 2016.

[6] David Altshuler, Peter Donnelly, and The International HapMap Consortium. A haplotype map of the human genome. *Nature*, 437(7063):1299–1320, October 2005.

[7] Joanna Amberger, Carol A. Bocchini, Alan F. Scott, and Ada Hamosh. McKusick's Online Mendelian Inheritance in Man (OMIM®). *Nucleic Acids Research*, 37(suppl_1):D793–D796, October 2008.

[8] Arleen D. Auerbach. Fanconi anemia and its diagnosis. *Mutation Research/Fundamental and Molecular Mechanisms of Mutagenesis*, 668(1):4–10, July 2009.

[9] Adam Auton, Gonçalo R. Abecasis, David M. Altshuler, Richard M. Durbin, Gonçalo R. Abecasis, David R. Bentley, Aravinda Chakravarti, Andrew G. Clark, Peter Donnelly, Evan E. Eichler, Paul Flicek, Stacey B. Gabriel, Richard A. Gibbs, Eric D. Green, Matthew E. Hurles, Bartha M. Knoppers, Jan O. Korbel, Eric S. Lander, Charles Lee, Hans Lehrach, Elaine R. Mardis, Gabor T. Marth, Gil A. McVean, Deborah A. Nickerson, Jeanette P. Schmidt, Stephen T. Sherry, Jun Wang, Richard K. Wilson, Richard A. Gibbs, Eric Boerwinkle, Harsha Doddapaneni, Yi Han, Viktoriya Korchina, Christie Kovar, Sandra Lee, Donna Muzny, Jeffrey G. Reid, Yiming Zhu, Jun Wang, Yuqi Chang, Qiang Feng, Xiaodong Fang, Xiaosen Guo, Min Jian, Hui Jiang, Xin Jin, Tianming Lan, Guoqing Li, Jingxiang Li, Yingrui Li, Shengmao Liu, Xiao Liu, Yao Lu, Xuedi Ma, Meifang Tang, Bo Wang, Guangbiao Wang, Honglong Wu, Renhua Wu, Xun Xu, Ye Yin, Dandan Zhang, Wenwei Zhang, Jiao Zhao, Meiru Zhao, Xiaole Zheng, Eric S. Lander, David M. Altshuler, Stacey B. Gabriel, Namrata Gupta, Neda Gharani, Lorraine H. Toji, Norman P. Gerry, Alissa M. Resch, Paul Flicek, Jonathan Barker, Laura Clarke, Laurent Gil, Sarah E. Hunt, Gavin Kelman, Eugene Kulesha, Rasko Leinonen, William M. McLaren, Rajesh Radhakrishnan, Asier Roa, Dmitriy Smirnov, Richard E. Smith, Ian Streeter, Anja Thormann, Iliana Toneva, Brendan Vaughan, Xiangqun Zheng-Bradley, David R. Bentley, Russell Grocock, Sean Humphray, Terena James, Zoya Kingsbury, Hans Lehrach, Ralf Sudbrak, Marcus W. Albrecht, Vyacheslav S. Amstislavskiy, Tatiana A. Borodina, Matthias Lienhard, Florian Mertes, Marc Sultan, Bernd Timmermann, Marie-Laure Yaspo, Elaine R. Mardis, Richard K. Wilson, Lucinda Fulton, Robert Fulton, Stephen T. Sherry, Victor Ananiev, Zinaida Belaia, Dimitriy Beloslyudtsev, Nathan Bouk, Chao Chen, Deanna Church, Robert Cohen, Charles Cook, John Garner, Timothy Hefferon, Mikhail Kimelman, Chunlei Liu, John Lopez, Peter Meric, Chris O'Sullivan, Yuri Ostapchuk, Lon Phan, Sergiy Ponomarov, Valerie Schneider, Eugene Shekhtman, Karl Sirotkin, Douglas Slotta, Hua Zhang, Gil A. McVean, Richard M. Durbin, Senduran Balasubramaniam, John Burton, Petr Danecek,

Thomas M. Keane, Anja Kolb-Kokocinski, Shane McCarthy, James Stalker, Michael Quail, Jeanette P. Schmidt, Christopher J. Davies, Jeremy Gollub, Teresa Webster, Brant Wong, Yiping Zhan, Adam Auton, Christopher L. Campbell, Yu Kong, Anthony Marcketta, Richard A. Gibbs, Fuli Yu, Lilian Antunes, Matthew Bainbridge, Donna Muzny, Aniko Sabo, Zhuoyi Huang, Jun Wang, Lachlan J. M. Coin, Lin Fang, Xiaosen Guo, Xin Jin, Guoqing Li, Qibin Li, Yingrui Li, Zhenyu Li, Haoxiang Lin, Binghang Liu, Ruibang Luo, Haojing Shao, Yinlong Xie, Chen Ye, Chang Yu, Fan Zhang, Hancheng Zheng, Hongmei Zhu, Can Alkan, Elif Dal, Fatma Kahveci, Gabor T. Marth, Erik P. Garrison, Deniz Kural, Wan-Ping Lee, Wen Fung Leong, Michael Stromberg, Alistair N. Ward, Jiantao Wu, Mengyao Zhang, Mark J. Daly, Mark A. DePristo, Robert E. Handsaker, David M. Altshuler, Eric Banks, Gaurav Bhatia, Guillermo del Angel, Stacey B. Gabriel, Giulio Genovese, Namrata Gupta, Heng Li, Seva Kashin, Eric S. Lander, Steven A. McCarroll, James C. Nemesh, Ryan E. Poplin, Seungtai C. Yoon, Jayon Lihm, Vladimir Makarov, Andrew G. Clark, Srikanth Gottipati, Alon Keinan, Juan L. Rodriguez-Flores, Jan O. Korbel, Tobias Rausch, Markus H. Fritz, Adrian M. Stütz, Paul Flicek, Kathryn Beal, Laura Clarke, Avik Datta, Javier Herrero, William M. McLaren, Graham R. S. Ritchie, Richard E. Smith, Daniel Zerbino, Xiangqun Zheng-Bradley, Pardis C. Sabeti, Ilya Shlyakhter, Stephen F. Schaffner, Joseph Vitti, David N. Cooper, Edward V. Ball, Peter D. Stenson, David R. Bentley, Bret Barnes, Markus Bauer, R. Keira Cheetham, Anthony Cox, Michael Eberle, Sean Humphray, Scott Kahn, Lisa Murray, John Peden, Richard Shaw, Eimear E. Kenny, Mark A. Batzer, Miriam K. Konkel, Jerilyn A. Walker, Daniel G. MacArthur, Monkol Lek, Ralf Sudbrak, Vyacheslav S. Amstislavskiy, Ralf Herwig, Elaine R. Mardis, Li Ding, Daniel C. Koboldt, David Larson, Kai Ye, Simon Gravel, The 1000 Genomes Project Consortium, Corresponding authors, Steering committee, Production group, Baylor College of Medicine, BGI-Shenzhen, Broad Institute of MIT and Harvard, Coriell Institute for Medical Research, European Bioinformatics Institute European Molecular Biology Laboratory, Illumina, Max Planck Institute for Molecular Genetics, McDonnell Genome Institute at Washington University, US National Institutes of Health, University of Oxford, Wellcome Trust Sanger Institute, Analysis group, Affymetrix, Albert Einstein College of Medicine, Bilkent University, Boston College, Cold Spring Harbor Laboratory, Cornell University, European Molecular Biology Laboratory, Harvard University, Human Gene Mutation Database, Icahn School of Medicine at Mount Sinai, Louisiana State University, Massachusetts General Hospital, McGill University, and NIH National Eye Institute. A global reference for human genetic variation. *Nature*, 526(7571):68–74, October 2015.

[10] Geraldine A. Van der Auwera and Brian D. O'Connor. *Genomics in the Cloud: Using Docker, GATK, and WDL in Terra*. O'Reilly Media, Sebastopol, CA, 1st edition edition, June 2020.

[11] Dustin Baldridge, Jennifer Heeley, Marisa Vineyard, Linda Manwaring, Tomi L. Toler, Emily Fassi, Elise Fiala, Sarah Brown, Charles W. Goss, Marcia Willing, Dorothy K. Grange, Beth A. Kozel, and Marwan Shinawi. The Exome Clinic and the role of med-

ical genetics expertise in the interpretation of exome sequencing results. *Genetics in Medicine*, 19(9):1040–1048, September 2017.

[12] Daniel Branton, David W. Deamer, Andre Marziali, Hagan Bayley, Steven A. Benner, Thomas Butler, Massimiliano Di Ventra, Slaven Garaj, Andrew Hibbs, Xiaohua Huang, Stevan B. Jovanovich, Predrag S. Krstic, Stuart Lindsay, Xinsheng Sean Ling, Carlos H. Mastrangelo, Amit Meller, John S. Oliver, Yuriy V. Pershin, J. Michael Ramsey, Robert Riehn, Gautam V. Soni, Vincent Tabard-Cossa, Meni Wanunu, Matthew Wiggin, and Jeffery A. Schloss. The potential and challenges of nanopore sequencing. *Nature Biotechnology*, 26(10):1146–1153, October 2008.

[13] M. Burrows and D. J. Wheeler. A block-sorting lossless data compression algorithm. Technical report, 1994.

[14] Mario PL Calus, Han A. Mulder, and John WM Bastiaansen. Identification of Mendelian inconsistencies between SNP and pedigree information of sibs. *Genetics Selection Evolution*, 43(1):34, October 2011.

[15] James Casaletto, Michael Parsons, Charles Markello, Yusuke Iwasaki, Yukihide Momozawa, Amanda B. Spurdle, and Melissa Cline. Federated analysis of BRCA1 and BRCA2 variation in a Japanese cohort. Technical report, December 2021.

[16] Nae-Chyun Chen, Brad Solomon, Taher Mun, Sheila Iyer, and Ben Langmead. Reference flow: reducing reference bias using multiple population genomes. *Genome Biology*, 22(1):8, January 2021.

[17] Chen-Shan Chin, Justin Wagner, Qiandong Zeng, Erik Garrison, Shilpa Garg, Arkarachai Fungtammasan, Mikko Rautiainen, Sergey Aganezov, Melanie Kirsche, Samantha Zarate, et al. A diploid assembly-based benchmark for variants in the major histocompatibility complex. *Nature Communications*, 11(1):1–9, 2020.

[18] Deanna M. Church, Valerie A. Schneider, Tina Graves, Katherine Auger, Fiona Cunningham, Nathan Bouk, Hsiu-Chuan Chen, Richa Agarwala, William M. McLaren, Graham R. S. Ritchie, Derek Albracht, Milinn Kremitzki, Susan Rock, Holland Kotkiewicz, Colin Kremitzki, Aye Wollam, Lee Trani, Lucinda Fulton, Robert Fulton, Lucy Matthews, Siobhan Whitehead, Will Chow, James Torrance, Matthew Dunn, Glenn Harden, Glen Threadgold, Jonathan Wood, Joanna Collins, Paul Heath, Guy Griffiths, Sarah Pelan, Darren Grafham, Evan E. Eichler, George Weinstock, Elaine R. Mardis, Richard K. Wilson, Kerstin Howe, Paul Flicek, and Tim Hubbard. Modernizing reference genome assemblies. *PLoS biology*, 9(7):e1001091, July 2011.

[19] P. Cingolani, A. Platts, M. Coon, T. Nguyen, L. Wang, S.J. Land, X. Lu, and D.M. Ruden. A program for annotating and predicting the effects of single nucleotide polymorphisms, snpeff: Snps in the genome of drosophila melanogaster strain w1118; iso-2; iso-3. *Fly*, 6(2):80–92, 2012.

[20] Michelle M. Clark, Zornitza Stark, Lauge Farnaes, Tiong Y. Tan, Susan M. White, David Dimmock, and Stephen F. Kingsmore. Meta-analysis of the diagnostic and clinical utility of genome and exome sequencing and chromosomal microarray in children with suspected genetic diseases. *npj Genomic Medicine*, 3(1):1–10, July 2018.

[21] Laura Clarke, Susan Fairley, Xiangqun Zheng-Bradley, Ian Streeter, Emily Perry, Ernesto Lowy, Anne-Marie Tassé, and Paul Flicek. The international Genome sample resource (IGSR): A worldwide collection of genome variation incorporating the 1000 Genomes Project data. *Nucleic Acids Research*, 45(D1):D854–D859, January 2017.

[22] John G Cleary, Ross Braithwaite, Kurt Gaastra, Brian S Hilbush, Stuart Inglis, Sean A Irvine, Alan Jackson, Richard Littin, Mehul Rathod, David Ware, et al. Comparing variant call files for performance benchmarking of next-generation sequencing variant calling pipelines. *bioRxiv*, 2015.

[23] Melissa S. Cline, Rachel G. Liao, Michael T. Parsons, Benedict Paten, Faisal Alquaddoomi, Antonis Antoniou, Samantha Baxter, Larry Brody, Robert Cook-Deegan, Amy Coffin, Fergus J. Couch, Brian Craft, Robert Currie, Chloe C. Dlott, Lena Dolman, Johan T. den Dunnen, Stephanie O. M. Dyke, Susan M. Domchek, Douglas Easton, Zachary Fischmann, William D. Foulkes, Judy Garber, David Goldgar, Mary J. Goldman, Peter Goodhand, Steven Harrison, David Haussler, Kazuto Kato, Bartha Knoppers, Charles Markello, Robert Nussbaum, Kenneth Offit, Sharon E. Plon, Jem Rashbass, Heidi L. Rehm, Mark Robson, Wendy S. Rubinstein, Dominique Stoppa-Lyonnet, Sean Tavtigian, Adrian Thorogood, Can Zhang, Marc Zimmermann, BRCA Challenge Authors, John Burn, Stephen Chanock, Gunnar Rätsch, and Amanda B. Spurdle. BRCA Challenge: BRCA Exchange as a global resource for variants in BRCA1 and BRCA2. *PLOS Genetics*, 14(12):e1007752, December 2018.

[24] Julie M. Collins and Claudine Isaacs. Management of breast cancer risk in BRCA1/2 mutation carriers who are unaffected with cancer. *The Breast Journal*, 26(8):1520–1527, 2020.

[25] Michael R. Crusoe, Sanne Abeln, Alexandru Iosup, Peter Amstutz, John Chilton, Nebojša Tijanić, Hervé Ménager, Stian Soiland-Reyes, Bogdan Gavrilovic, and Carole Goble. Methods Included: Standardizing Computational Reuse and Portability with the Common Workflow Language. *arXiv:2105.07028 [cs]*, August 2021.

[26] Olivier Delaneau, Jean-François Zagury, Matthew R. Robinson, Jonathan L. Marchini, and Emmanouil T. Dermitzakis. Accurate, scalable and integrative haplotype estimation. *Nature Communications*, 10(1):5436, November 2019.

[27] Johan T. den Dunnen, Raymond Dalgleish, Donna R. Maglott, Reece K. Hart, Marc S. Greenblatt, Jean McGowan-Jordan, Anne-Francoise Roux, Timothy Smith, Stylianos E. Antonarakis, and Peter E.M. Taschner. HGVS Recommendations for the Description of Sequence Variants: 2016 Update. *Human Mutation*, 37(6):564–569, 2016.

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

[29] Alexander Dilthey, Charles Cox, Zamin Iqbal, Matthew R. Nelson, and Gil McVean. Improved genome inference in the MHC using a population reference graph. *Nature Genetics*, 47(6):682–688, June 2015.

[30] Richard Durbin. Efficient haplotype matching and storage using the positional Burrows–Wheeler transform (PBWT). *Bioinformatics*, 30(9):1266–1272, May 2014.

[31] Bing-Jian Feng. PERCH: A Unified Framework for Disease Gene Prioritization. *Human Mutation*, 38(3):243–251, 2017.

[32] Paolo Ferragina and Giovanni Manzini. Indexing compressed text. *Journal of the ACM*, 52(4):552–581, July 2005.

[33] Ivo F. A. C. Fokkema, Peter E. M. Taschner, Gerard C. P. Schaafsma, J. Celli, Jeroen F. J. Laros, and Johan T. den Dunnen. LOVD v.2.0: the next generation in gene variant databases. *Human Mutation*, 32(5):557–563, 2011.

[34] Laurent C. Francioli, Paz P. Polak, Amnon Koren, Androniki Menelaou, Sung Chun, Ivo Renkens, Cornelia M. van Duijn, Morris Swertz, Cisca Wijmenga, Gertjan van Ommen, P. Eline Slagboom, Dorret I. Boomsma, Kai Ye, Victor Guryev, Peter F. Arndt, Wigard P. Kloosterman, Paul I. W. de Bakker, and Shamil R. Sunyaev. Genome-wide patterns and properties of de novo mutations in humans. *Nature Genetics*, 47(7):822–826, July 2015.

[35] Carl W. Fuller, Lyle R. Middendorf, Steven A. Benner, George M. Church, Timothy Harris, Xiaohua Huang, Stevan B. Jovanovich, John R. Nelson, Jeffery A. Schloss, David C. Schwartz, and Dmitri V. Vezenov. The challenges of sequencing by synthesis. *Nature Biotechnology*, 27(11):1013–1023, November 2009.

[36] William A. Gahl, Thomas C. Markello, Camilo Toro, Karin Fuentes Fajardo, Murat Sincan, Fred Gill, Hannah Carlson-Donohoe, Andrea Gropman, Tyler Mark Pierson, Gretchen Golas, Lynne Wolfe, Catherine Groden, Rena Godfrey, Michele Nehrebecky, Colleen Wahl, Dennis M. D. Landis, Sandra Yang, Anne Madeo, James C. Mullikin, Cornelius F. Boerkoel, Cynthia J. Tifft, and David Adams. The National Institutes of Health Undiagnosed Diseases Program: insights into rare diseases. *Genetics in Medicine*, 14(1):51–59, January 2012.

[37] William A. Gahl, John J. Mulvihill, Camilo Toro, Thomas C. Markello, Anastasia L. Wise, Rachel B. Ramoni, David R. Adams, and Cynthia J. Tifft. The NIH Undiagnosed Diseases Program and Network: Applications to modern medicine. *Molecular genetics and metabolism*, 117(4):393–400, April 2016.

[38] William A. Gahl and Cynthia J. Tifft. The NIH Undiagnosed Diseases Program: Lessons Learned. *JAMA*, 305(18):1904–1905, May 2011.

[39] William A. Gahl, Anastasia L. Wise, and Euan A. Ashley. The Undiagnosed Diseases Network of the National Institutes of Health: A National Extension. *JAMA*, 314(17):1797–1798, November 2015.

[40] Timothy Gall, Elise Valkanas, Christofer Bello, Thomas Markello, Christopher Adams, William P. Bone, Alexander J. Brandt, Jennifer M. Brazill, Lynn Carmichael, Mariska Davids, Joie Davis, Zoraida Diaz-Perez, David Draper, Jeremy Elson, Elise D. Flynn, Rena Godfrey, Catherine Groden, Cheng-Kang Hsieh, Roxanne Fischer, Gretchen A. Golas, Jessica Guzman, Yan Huang, Megan S. Kane, Elizabeth Lee, Chong Li, Amanda E. Links, Valerie Maduro, May Christine V. Malicdan, Fayeza S. Malik, Michele Nehrebecky, Joun Park, Paul Pemberton, Katherine Schaffer, Dimitre Simeonov, Murat Sincan, Damian Smedley, Zaheer Valivullah, Colleen Wahl, Nicole Washington, Lynne A. Wolfe, Karen Xu, Yi Zhu, William A. Gahl, Cynthia J. Tifft, Camillo Toro, David R. Adams, Miao He, Peter N. Robinson, Melissa A. Haendel, R. Grace Zhai, and Cornelius F. Boerkoel. Defining Disease, Diagnosis, and Translational Medicine within a Homeostatic Perturbation Paradigm: The National Institutes of Health Undiagnosed Diseases Program Experience. *Frontiers in Medicine*, 4:62, 2017.

[41] Shilpa Garg. Computational methods for chromosome-scale haplotype reconstruction. *Genome Biology*, 22(1):101, April 2021.

[42] Shilpa Garg, John Aach, Heng Li, Isaac Sebenius, Richard Durbin, and George Church. A haplotype-aware de novo assembly of related individuals using pedigree sequence graph. *Bioinformatics*, 36(8):2385–2392, December 2019.

[43] Amy S. Gargis, Lisa Kalman, Meredith W. Berry, David P. Bick, David P. Dimmock, Tina Hambuch, Fei Lu, Elaine Lyon, Karl V. Voelkerding, Barbara A. Zehnbauer, Richa Agarwala, Sarah F. Bennett, Bin Chen, Ephrem L. H. Chin, John G. Compton, Soma Das, Daniel H. Farkas, Matthew J. Ferber, Birgit H. Funke, Manohar R. Furtado, Lilia M. Ganova-Raeva, Ute Geigenmüller, Sandra J. Gunselman, Madhuri R. Hegde, Philip L. F. Johnson, Andrew Kasarskis, Shashikant Kulkarni, Thomas Lenk, C. S. Jonathan Liu, Megan Manion, Teri A. Manolio, Elaine R. Mardis, Jason D. Merker, Mangalathu S. Rajeevan, Martin G. Reese, Heidi L. Rehm, Birgitte B. Simen, Joanne M. Yeakley, Justin M. Zook, and Ira M. Lubin. Assuring the quality of next-generation sequencing in clinical laboratory practice. *Nature Biotechnology*, 30(11):1033–1036, November 2012.

[44] Erik Garrison, Jouni Sirén, Adam M Novak, Glenn Hickey, Jordan M Eizenga, Eric T Dawson, William Jones, Shilpa Garg, Charles Markello, Michael F Lin, Benedict Paten, and Richard Durbin. Variation graph toolkit improves read mapping by representing genetic variation in the reference. *Nature Biotechnology*, 36(9):875–879, October 2018.

[45] Fangning Gu, Anchi Wu, M. Grace Gordon, Lukas Vlahos, Shane Macnamara, Elizabeth Burke, May C. Malicdan, David R. Adams, Cynthia J. Tifft, Camilo Toro, William A.

Gahl, and Thomas C. Markello. A suite of automated sequence analyses reduces the number of candidate deleterious variants and reveals a difference between probands and unaffected siblings. *Genetics in Medicine*, 21(8):1772–1780, August 2019.

[46] Verena Heinrich, Jens Stange, Thorsten Dickhaus, Peter Imkeller, Ulrike Krüger, Sebastian Bauer, Stefan Mundlos, Peter N. Robinson, Jochen Hecht, and Peter M. Krawitz. The allele distribution in next-generation sequencing data sets is accurately described as the result of a stochastic branching process. *Nucleic Acids Research*, 40(6):2426–2431, March 2012.

[47] Brenna M. Henn, Laura R. Botigué, Carlos D. Bustamante, Andrew G. Clark, and Simon Gravel. Estimating the mutation load in human genomes. *Nature Reviews Genetics*, 16(6):333–343, June 2015.

[48] Ryan D. Hernandez, Lawrence H. Uricchio, Kevin Hartman, Chun Ye, Andrew Dahl, and Noah Zaitlen. Ultrarare variants drive substantial cis heritability of human gene expression. *Nature Genetics*, 51(9):1349–1355, September 2019.

[49] Bill Howe. Virtual Appliances, Cloud Computing, and Reproducible Research. *Computing in Science Engineering*, 14(4):36–41, July 2012.

[50] John Huddleston, Mark J. P. Chaisson, Karyn Meltz Steinberg, Wes Warren, Kendra Hoekzema, David Gordon, Tina A. Graves-Lindsay, Katherine M. Munson, Zev N. Kronenberg, Laura Vives, Paul Peluso, Matthew Boitano, Chen-Shin Chin, Jonas Korlach, Richard K. Wilson, and Evan E. Eichler. Discovery and genotyping of structural variation from long-read haploid genome sequence data. *Genome Research*, 27(5):677–685, May 2017.

[51] Illumina. Accuracy Improvements in Germline Small Variant Calling with the DRAGEN Platform. https://science-docs.illumina.com/documents/Informatics/dragen-v3-accuracy-appnote-html-970-2019-006/Content/Source/Informatics/Dragen/dragen-v3-accuracy-appnote-970-2019-006/dragen-v3-accuracy-appnote-970-2019-006.html.

[52] 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. *Nature Biotechnology*, 36(4):338–345, April 2018.

[53] Sean P. Kane and Karl Matthias. *Docker: Up & Running: Shipping Reliable Containers in Production*. O'Reilly Media, Sebastopol, CA, 2nd edition edition, October 2018.

[54] Konrad J. Karczewski, Laurent C. Francioli, Grace Tiao, Beryl B. Cummings, Jessica Alföldi, Qingbo Wang, Ryan L. Collins, Kristen M. Laricchia, Andrea Ganna, Daniel P.

Birnbaum, Laura D. Gauthier, Harrison Brand, Matthew Solomonson, Nicholas A. Watts, Daniel Rhodes, Moriel Singer-Berk, Eleina M. England, Eleanor G. Seaby, Jack A. Kosmicki, Raymond K. Walters, Katherine Tashman, Yossi Farjoun, Eric Banks, Timothy Poterba, Arcturus Wang, Cotton Seed, Nicola Whiffin, Jessica X. Chong, Kaitlin E. Samocha, Emma Pierce-Hoffman, Zachary Zappala, Anne H. O'Donnell-Luria, Eric Vallabh Minikel, Ben Weisburd, Monkol Lek, James S. Ware, Christopher Vittal, Irina M. Armean, Louis Bergelson, Kristian Cibulskis, Kristen M. Connolly, Miguel Covarrubias, Stacey Donnelly, Steven Ferriera, Stacey Gabriel, Jeff Gentry, Namrata Gupta, Thibault Jeandet, Diane Kaplan, Christopher Llanwarne, Ruchi Munshi, Sam Novod, Nikelle Petrillo, David Roazen, Valentin Ruano-Rubio, Andrea Saltzman, Molly Schleicher, Jose Soto, Kathleen Tibbetts, Charlotte Tolonen, Gordon Wade, Michael E. Talkowski, Benjamin M. Neale, Mark J. Daly, and Daniel G. MacArthur. The mutational constraint spectrum quantified from variation in 141,456 humans. *Nature*, 581(7809):434–443, May 2020.

[55] Shilpa Nadimpalli Kobren, Dustin Baldridge, Matt Velinder, Joel B. Krier, Kimberly LeBlanc, Cecilia Esteves, Barbara N. Pusey, Stephan Züchner, Elizabeth Blue, Hane Lee, Alden Huang, Lisa Bastarache, Anna Bican, Joy Cogan, Shruti Marwaha, Anna Alkelai, David R. Murdock, Pengfei Liu, Daniel J. Wegner, Alexander J. Paul, Shamil R. Sunyaev, and Isaac S. Kohane. Commonalities across computational workflows for uncovering explanatory variants in undiagnosed cases. *Genetics in Medicine*, 23(6):1075–1085, June 2021.

[56] Alexey Kolesnikov, Sidharth Goel, Maria Nattestad, Taedong Yun, Gunjan Baid, Howard Yang, Cory Y. McLean, Pi-Chuan Chang, and Andrew Carroll. DeepTrio: Variant Calling in Families Using Deep Learning. *bioRxiv*, page 2021.04.05.438434, April 2021.

[57] Augustine Kong, Michael L. Frigge, Gisli Masson, Soren Besenbacher, Patrick Sulem, Gisli Magnusson, Sigurjon A. Gudjonsson, Asgeir Sigurdsson, Aslaug Jonasdottir, Adalbjorg Jonasdottir, Wendy S. W. Wong, Gunnar Sigurdsson, G. Bragi Walters, Stacy Steinberg, Hannes Helgason, Gudmar Thorleifsson, Daniel F. Gudbjartsson, Agnar Helgason, Olafur Th Magnusson, Unnur Thorsteinsdottir, and Kari Stefansson. Rate of de novo mutations and the importance of father's age to disease risk. *Nature*, 488(7412):471–475, August 2012.

[58] Peter Krusche, Len Trigg, Paul C. Boutros, Christopher E. Mason, Francisco M. De La Vega, Benjamin L. Moore, Mar Gonzalez-Porta, Michael A. Eberle, Zivana Tezak, Samir Lababidi, Rebecca Truty, George Asimenos, Birgit Funke, Mark Fleharty, Brad A. Chapman, Marc Salit, and Justin M. Zook. Best practices for benchmarking germline small-variant calls in human genomes. *Nature Biotechnology*, 37(5):555–560, May 2019.

[59] Gregory M. Kurtzer, Vanessa Sochat, and Michael W. Bauer. Singularity: Scientific containers for mobility of compute. *PLOS ONE*, 12(5):e0177459, May 2017.

[60] Melissa J Landrum, Jennifer M Lee, Mark Benson, Garth R Brown, Chen Chao, Shanmuga Chitipiralla, Baoshan Gu, Jennifer Hart, Douglas Hoffman, Wonhee Jang, Karen

Karapetyan, Kenneth Katz, Chunlei Liu, Zenith Maddipatla, Adriana Malheiro, Kurt McDaniel, Michael Ovetsky, George Riley, George Zhou, J Bradley Holmes, Brandi L Kattman, and Donna R Maglott. ClinVar: improving access to variant interpretations and supporting evidence. *Nucleic Acids Research*, 46(Database issue):D1062–D1067, January 2018.

[61] Ben Langmead, Cole Trapnell, Mihai Pop, and Steven L. Salzberg. Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biology*, 10(3):R25, March 2009.

[62] Christopher Lee, Catherine Grasso, and Mark F. Sharlow. Multiple sequence alignment using partial order graphs. *Bioinformatics*, 18(3):452–464, March 2002.

[63] Monkol Lek, Konrad J. Karczewski, Eric V. Minikel, Kaitlin E. Samocha, Eric Banks, Timothy Fennell, Anne H. O'Donnell-Luria, James S. Ware, Andrew J. Hill, Beryl B. Cummings, Taru Tukiainen, Daniel P. Birnbaum, Jack A. Kosmicki, Laramie E. Duncan, Karol Estrada, Fengmei Zhao, James Zou, Emma Pierce-Hoffman, Joanne Berghout, David N. Cooper, Nicole Deflaux, Mark DePristo, Ron Do, Jason Flannick, Menachem Fromer, Laura Gauthier, Jackie Goldstein, Namrata Gupta, Daniel Howrigan, Adam Kiezun, Mitja I. Kurki, Ami Levy Moonshine, Pradeep Natarajan, Lorena Orozco, Gina M. Peloso, Ryan Poplin, Manuel A. Rivas, Valentin Ruano-Rubio, Samuel A. Rose, Douglas M. Ruderfer, Khalid Shakir, Peter D. Stenson, Christine Stevens, Brett P. Thomas, Grace Tiao, Maria T. Tusie-Luna, Ben Weisburd, Hong-Hee Won, Dongmei Yu, David M. Altshuler, Diego Ardissino, Michael Boehnke, John Danesh, Stacey Donnelly, Roberto Elosua, Jose C. Florez, Stacey B. Gabriel, Gad Getz, Stephen J. Glatt, Christina M. Hultman, Sekar Kathiresan, Markku Laakso, Steven McCarroll, Mark I. McCarthy, Dermot McGovern, Ruth McPherson, Benjamin M. Neale, Aarno Palotie, Shaun M. Purcell, Danish Saleheen, Jeremiah M. Scharf, Pamela Sklar, Patrick F. Sullivan, Jaakko Tuomilehto, Ming T. Tsuang, Hugh C. Watkins, James G. Wilson, Mark J. Daly, and Daniel G. MacArthur. Analysis of protein-coding genetic variation in 60,706 humans. *Nature*, 536(7616):285–291, August 2016.

[64] Heng Li. Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM. *arXiv:1303.3997 [q-bio]*, May 2013.

[65] Xin Li, Yungil Kim, Emily K. Tsang, Joe R. Davis, Farhan N. Damani, Colby Chiang, Gaelen T. Hess, Zachary Zappala, Benjamin J. Strober, Alexandra J. Scott, Amy Li, Andrea Ganna, Michael C. Bassik, Jason D. Merker, François Aguet, Kristin G. Ardlie, Beryl B. Cummings, Ellen T. Gelfand, Gad Getz, Kane Hadley, Robert E. Handsaker, Katherine H. Huang, Seva Kashin, Konrad J. Karczewski, Monkol Lek, Xiao Li, Daniel G. MacArthur, Jared L. Nedzel, Duyen T. Nguyen, Michael S. Noble, Ayellet V. Segrè, Casandra A. Trowbridge, Taru Tukiainen, Nathan S. Abell, Brunilda Balliu, Ruth Barshir, Omer Basha, Alexis Battle, Gireesh K. Bogu, Andrew Brown, Christopher D. Brown, Stephane E. Castel, Lin S. Chen, Colby Chiang, Donald F. Conrad, Nancy J. Cox, Farhan N. Damani, Joe R. Davis, Olivier Delaneau, Emmanouil T. Dermitzakis,

Barbara E. Engelhardt, Eleazar Eskin, Pedro G. Ferreira, Laure Frésard, Eric R. Gamazon, Diego Garrido-Martín, Ariel D.H. Gewirtz, Genna Gliner, Michael J. Gloudemans, Roderic Guigo, Ira M. Hall, Buhm Han, Yuan He, Farhad Hormozdiari, Cedric Howald, Hae Kyung Im, Brian Jo, Eun Yong Kang, Yungil Kim, Sarah Kim-Hellmuth, Tuuli Lappalainen, Gen Li, Xin Li, Boxiang Liu, Serghei Mangul, Mark I. McCarthy, Ian C. McDowell, Pejman Mohammadi, Jean Monlong, Stephen B. Montgomery, Manuel Muñoz-Aguirre, Anne W. Ndungu, Dan L. Nicolae, Andrew B. Nobel, Meritxell Oliva, Halit Ongen, John J. Palowitch, Nikolaos Panousis, Panagiotis Papasaikas, YoSon Park, Princy Parsana, Anthony J. Payne, Christine B. Peterson, Jie Quan, Ferran Reverter, Chiara Sabatti, Ashis Saha, Michael Sammeth, Alexandra J. Scott, Andrey A. Shabalin, Reza Sodaei, Matthew Stephens, Barbara E. Stranger, Benjamin J. Strober, Jae Hoon Sul, Emily K. Tsang, Sarah Urbut, Martijn van de Bunt, Gao Wang, Xiaoquan Wen, Fred A. Wright, Hualin S. Xi, Esti Yeger-Lotem, Zachary Zappala, Judith B. Zaugg, Yi-Hui Zhou, Joshua M. Akey, Daniel Bates, Joanne Chan, Lin S. Chen, Melina Claussnitzer, Kathryn Demanelis, Morgan Diegel, Jennifer A. Doherty, Andrew P. Feinberg, Marian S. Fernando, Jessica Halow, Kasper D. Hansen, Eric Haugen, Peter F. Hickey, Lei Hou, Farzana Jasmine, Ruiqi Jian, Lihua Jiang, Audra Johnson, Rajinder Kaul, Manolis Kellis, Muhammad G. Kibriya, Kristen Lee, Jin Billy Li, Qin Li, Xiao Li, Jessica Lin, Shin Lin, Sandra Linder, Caroline Linke, Yaping Liu, Matthew T. Maurano, Benoit Molinie, Stephen B. Montgomery, Jemma Nelson, Fidencio J. Neri, Meritxell Oliva, Yongjin Park, Brandon L. Pierce, Nicola J. Rinaldi, Lindsay F. Rizzardi, Richard Sandstrom, Andrew Skol, Kevin S. Smith, Michael P. Snyder, John Stamatoyannopoulos, Barbara E. Stranger, Hua Tang, Emily K. Tsang, Li Wang, Meng Wang, Nicholas Van Wittenberghe, Fan Wu, Rui Zhang, Concepcion R. Nierras, Philip A. Branton, Latarsha J. Carithers, Ping Guan, Helen M. Moore, Abhi Rao, Jimmie B. Vaught, Sarah E. Gould, Nicole C. Lockart, Casey Martin, Jeffery P. Struewing, Simona Volpi, Anjene M. Addington, Susan E. Koester, A. Roger Little, Lori E. Brigham, Richard Hasz, Marcus Hunter, Christopher Johns, Mark Johnson, Gene Kopen, William F. Leinweber, John T. Lonsdale, Alisa McDonald, Bernadette Mestichelli, Kevin Myer, Brian Roe, Michael Salvatore, Saboor Shad, Jeffrey A. Thomas, Gary Walters, Michael Washington, Joseph Wheeler, Jason Bridge, Barbara A. Foster, Bryan M. Gillard, Ellen Karasik, Rachna Kumar, Mark Miklos, Michael T. Moser, Scott D. Jewell, Robert G. Montroy, Daniel C. Rohrer, Dana R. Valley, David A. Davis, Deborah C. Mash, Anita H. Undale, Anna M. Smith, David E. Tabor, Nancy V. Roche, Jeffrey A. McLean, Negin Vatanian, Karna L. Robinson, Leslie Sobin, Mary E. Barcus, Kimberly M. Valentino, Liqun Qi, Steven Hunter, Pushpa Hariharan, Shilpi Singh, Ki Sung Um, Takunda Matose, Maria M. Tomaszewski, Laura K. Barker, Maghboeba Mosavel, Laura A. Siminoff, Heather M. Traino, Paul Flicek, Thomas Juettemann, Magali Ruffier, Dan Sheppard, Kieron Taylor, Stephen J. Trevanion, Daniel R. Zerbino, Brian Craft, Mary Goldman, Maximilian Haeussler, W. James Kent, Christopher M. Lee, Benedict Paten, Kate R. Rosenbloom, John Vivian, Jingchun Zhu, Ira M. Hall, Alexis Battle, Stephen B. Montgomery, GTEx Consortium, Data Analysis & Coordinating Center (LDACC)—Analysis Working Group Laboratory, Statistical Methods groups—Analysis Working Group, Enhancing GTEx (eGTEx) groups, NIH

Common Fund, NIH/NCI, NIH/NHGRI, NIH/NIMH, NIH/NIDA, Biospecimen Collection Source Site—NDRI, Biospecimen Collection Source Site—RPCI, Biospecimen Core Resource—VARI, Brain Bank Repository—University of Miami Brain Endowment Bank, Leidos Biomedical—Project Management, ELSI Study, Genome Browser Data Integration & Visualization—EBI, and University of California Santa Cruz Genome Browser Data Integration & Visualization—UCSC Genomics Institute. The impact of rare variation on gene expression across tissues. *Nature*, 550(7675):239–243, October 2017.

[66] Po-Ru Loh, Petr Danecek, Pier Francesco Palamara, Christian Fuchsberger, Yakir A Reshef, Hilary K Finucane, Sebastian Schoenherr, Lukas Forer, Shane McCarthy, Goncalo R Abecasis, Richard Durbin, and Alkes L Price. Reference-based phasing using the Haplotype Reference Consortium panel. *Nature Genetics*, 48(11):1443–1448, November 2016.

[67] Swapan Mallick, Heng Li, Mark Lipson, Iain Mathieson, Melissa Gymrek, Fernando Racimo, Mengyao Zhao, Niru Chennagiri, Susanne Nordenfelt, Arti Tandon, Pontus Skoglund, Iosif Lazaridis, Sriram Sankararaman, Qiaomei Fu, Nadin Rohland, Gabriel Renaud, Yaniv Erlich, Thomas Willems, Carla Gallo, Jeffrey P. Spence, Yun S. Song, Giovanni Poletti, Francois Balloux, George van Driem, Peter de Knijff, Irene Gallego Romero, Aashish R. Jha, Doron M. Behar, Claudio M. Bravi, Cristian Capelli, Tor Hervig, Andres Moreno-Estrada, Olga L. Posukh, Elena Balanovska, Oleg Balanovsky, Sena Karachanak-Yankova, Hovhannes Sahakyan, Draga Toncheva, Levon Yepiskoposyan, Chris Tyler-Smith, Yali Xue, M. Syafiq Abdullah, Andres Ruiz-Linares, Cynthia M. Beall, Anna Di Rienzo, Choongwon Jeong, Elena B. Starikovskaya, Ene Metspalu, Jüri Parik, Richard Villems, Brenna M. Henn, Ugur Hodoglugil, Robert Mahley, Antti Sajantila, George Stamatoyannopoulos, Joseph T. S. Wee, Rita Khusainova, Elza Khusnutdinova, Sergey Litvinov, George Ayodo, David Comas, Michael F. Hammer, Toomas Kivisild, William Klitz, Cheryl A. Winkler, Damian Labuda, Michael Bamshad, Lynn B. Jorde, Sarah A. Tishkoff, W. Scott Watkins, Mait Metspalu, Stanislav Dryomov, Rem Sukernik, Lalji Singh, Kumarasamy Thangaraj, Svante Pääbo, Janet Kelso, Nick Patterson, and David Reich. The Simons Genome Diversity Project: 300 genomes from 142 diverse populations. *Nature*, 538(7624):201–206, October 2016.

[68] Jacob Mallott, Antonia Kwan, Joseph Church, Diana Gonzalez-Espinosa, Fred Lorey, Ling Fung Tang, Uma Sunderam, Sadhna Rana, Rajgopal Srinivasan, Steven E. Brenner, and Jennifer Puck. Newborn Screening for SCID Identifies Patients with Ataxia Telangiectasia. *Journal of Clinical Immunology*, 33(3):540–549, April 2013.

[69] Diana Mandelker, Ryan J. Schmidt, Arunkanth Ankala, Kristin McDonald Gibson, Mark Bowser, Himanshu Sharma, Elizabeth Duffy, Madhuri Hegde, Avni Santani, Matthew Lebo, and Birgit Funke. Navigating highly homologous genes in a molecular diagnostic setting: a resource for clinical next-generation sequencing. *Genetics in Medicine*, 18(12):1282–1289, December 2016.

[70] Thomas C. Markello, Ted Han, Hannah Carlson-Donohoe, Chidi Ahaghotu, Ursula Harper, MaryPat Jones, Settara Chandrasekharappa, Yair Anikster, David R. Adams, NISC Comparative Sequencing Program, William A. Gahl, and Cornelius F. Boerkoel. Recombination mapping using Boolean logic and high-density SNP genotyping for exome sequence filtering. *Molecular Genetics and Metabolism*, 105(3):382–389, March 2012.

[71] Marcel Martin, Murray Patterson, Shilpa Garg, Sarah O Fischer, Nadia Pisanti, Gunnar W Klau, Alexander Schöenhuth, and Tobias Marschall. WhatsHap: fast and accurate read-based phasing. *bioRxiv*, page 085050, January 2016.

[72] Aaron McKenna, Matthew Hanna, Eric Banks, Andrey Sivachenko, Kristian Cibulskis, Andrew Kernytsky, Kiran Garimella, David Altshuler, Stacey Gabriel, Mark Daly, and Mark A. DePristo. The Genome Analysis Toolkit: A MapReduce framework for analyzing next-generation DNA sequencing data. *Genome Research*, 20(9):1297–1303, September 2010.

[73] M. Mielczarek and J. Szyda. Review of alignment and SNP calling algorithms for next-generation sequencing data. *Journal of Applied Genetics*, 57(1):71–79, February 2016.

[74] Yukihide Momozawa, Yusuke Iwasaki, Michael T. Parsons, Yoichiro Kamatani, Atsushi Takahashi, Chieko Tamura, Toyomasa Katagiri, Teruhiko Yoshida, Seigo Nakamura, Kokichi Sugano, Yoshio Miki, Makoto Hirata, Koichi Matsuda, Amanda B. Spurdle, and Michiaki Kubo. Germline pathogenic variants of 11 breast cancer genes in 7,051 Japanese patients and 11,241 controls. *Nature Communications*, 9(1):4083, October 2018.

[75] Sowmiya Moorthie, Christopher J. Mattocks, and Caroline F. Wright. Review of massively parallel DNA sequencing technologies. *The HUGO Journal*, 5(1):1–12, December 2011.

[76] Lisle E Mose, Charles M Perou, and Joel S Parker. Improved indel detection in DNA and RNA via realignment with ABRA2. *Bioinformatics*, 35(17):2966–2973, 01 2019.

[77] Lisle E Mose, Charles M Perou, and Joel S Parker. Improved indel detection in DNA and RNA via realignment with ABRA2. *Bioinformatics*, 35(17):2966–2973, September 2019.

[78] Saul B. Needleman and Christian D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48(3):443–453, March 1970.

[79] Adam M. Novak, Erik Garrison, and Benedict Paten. A graph extension of the positional Burrows–Wheeler transform and its applications. *Algorithms for Molecular Biology*, 12(1):18, July 2017.

[80] Sergey Nurk, Sergey Koren, Arang Rhie, Mikko Rautiainen, Andrey V. Bzikadze, Alla Mikheenko, Mitchell R. Vollger, Nicolas Altemose, Lev Uralsky, Ariel Gershman, Sergey Aganezov, Savannah J. Hoyt, Mark Diekhans, Glennis A. Logsdon, Michael Alonge, Stylianos E. Antonarakis, Matthew Borchers, Gerard G. Bouffard, Shelise Y. Brooks, Gina V. Caldas, Haoyu Cheng, Chen-Shan Chin, William Chow, Leonardo G. de Lima, Philip C. Dishuck, Richard Durbin, Tatiana Dvorkina, Ian T. Fiddes, Giulio Formenti, Robert S. Fulton, Arkarachai Fungtammasan, Erik Garrison, Patrick G. S. Grady, Tina A. Graves-Lindsay, Ira M. Hall, Nancy F. Hansen, Gabrielle A. Hartley, Marina Haukness, Kerstin Howe, Michael W. Hunkapiller, Chirag Jain, Miten Jain, Erich D. Jarvis, Peter Kerpedjiev, Melanie Kirsche, Mikhail Kolmogorov, Jonas Korlach, Milinn Kremitzki, Heng Li, Valerie V. Maduro, Tobias Marschall, Ann M. McCartney, Jennifer McDaniel, Danny E. Miller, James C. Mullikin, Eugene W. Myers, Nathan D. Olson, Benedict Paten, Paul Peluso, Pavel A. Pevzner, David Porubsky, Tamara Potapova, Evgeny I. Rogaev, Jeffrey A. Rosenfeld, Steven L. Salzberg, Valerie A. Schneider, Fritz J. Sedlazeck, Kishwar Shafin, Colin J. Shew, Alaina Shumate, Yumi Sims, Arian F. A. Smit, Daniela C. Soto, Ivan Sović, Jessica M. Storer, Aaron Streets, Beth A. Sullivan, Françoise Thibaud-Nissen, James Torrance, Justin Wagner, Brian P. Walenz, Aaron Wenger, Jonathan M. D. Wood, Chunlin Xiao, Stephanie M. Yan, Alice C. Young, Samantha Zarate, Urvashi Surti, Rajiv C. McCoy, Megan Y. Dennis, Ivan A. Alexandrov, Jennifer L. Gerton, Rachel J. O'Neill, Winston Timp, Justin M. Zook, Michael C. Schatz, Evan E. Eichler, Karen H. Miga, and Adam M. Phillippy. The complete sequence of a human genome. Technical report, May 2021.

[81] Brian D. O'Connor, Denis Yuen, Vincent Chung, Andrew G. Duncan, Xiang Kun Liu, Janice Patricia, Benedict Paten, Lincoln Stein, and Vincent Ferretti. The Dockstore: enabling modular, community-focused sharing of Docker-based genomics tools and workflows. *F1000Research*, 6:52, January 2017.

[82] Nathan D. Olson, Justin Wagner, Jennifer McDaniel, Sarah H. Stephens, Samuel T. Westreich, Anish G. Prasanna, Elaine Johanson, Emily Boja, Ezekiel J. Maier, Omar Serang, David Jáspez, José M. Lorenzo-Salazar, Adrián Muñoz-Barrera, Luis A. Rubio-Rodríguez, Carlos Flores, Konstantinos Kyriakidis, Andigoni Malousi, Kishwar Shafin, Trevor Pesout, Miten Jain, Benedict Paten, Pi-Chuan Chang, Alexey Kolesnikov, Maria Nattestad, Gunjan Baid, Sidharth Goel, Howard Yang, Andrew Carroll, Robert Eveleigh, Mathieu Bourgey, Guillaume Bourque, Gen Li, M. A. ChouXian, LinQi Tang, D. U. YuanPing, ShaoWei Zhang, Jordi Morata, Raúl Tonda, Genís Parra, Jean-Rémi Trotta, Christian Brueffer, Sinem Demirkaya-Budak, Duygu Kabakci-Zorlu, Deniz Turgut, Özem Kalay, Gungor Budak, Kübra Narcı, Elif Arslan, Richard Brown, Ivan J. Johnson, Alexey Dolgoborodov, Vladimir Semenyuk, Amit Jain, H. Serhat Tetikol, Varun Jain, Mike Ruehle, Bryan Lajoie, Cooper Roddey, Severine Catreux, Rami Mehio, Mian Umair Ahsan, Qian Liu, Kai Wang, Sayed Mohammad Ebrahim Sahraeian, Li Tai Fang, Marghoob Mohiyuddin, Calvin Hung, Chirag Jain, Hanying Feng, Zhipan Li, Luoqi Chen, Fritz J. Sedlazeck, and Justin M. Zook. precisionFDA Truth Challenge V2:

Calling variants from short- and long-reads in difficult-to-map regions. Technical report, February 2021.

[83] Zubin Patel, Leah Kottyan, Sara Lazaro, Marc Williams, David Ledbetter, Gerard Tromp, Andrew Rupert, Mojtaba Kohram, Michael Wagner, Ammar Husami, Yaping Qian, C. Alexander Valencia, Kejian Zhang, Margaret Hostetter, John Harley, and Kenneth Kaufman. The struggle to find reliable results in exome sequencing data: filtering out Mendelian errors. *Frontiers in Genetics*, 5:16, 2014.

[84] Brent S. Pedersen, Joe M. Brown, Harriet Dashnow, Amelia D. Wallace, Matt Velinder, Martin Tristani-Firouzi, Joshua D. Schiffman, Tatiana Tvrdik, Rong Mao, D. Hunter Best, Pinar Bayrak-Toydemir, and Aaron R. Quinlan. Effective variant filtering and expected candidate variant yield in studies of rare human disease. *npj Genomic Medicine*, 6(1):1–8, July 2021.

[85] Gang Peng, Yu Fan, Timothy B. Palculict, Peidong Shen, E. Cristy Ruteshouser, Aung-Kyaw Chi, Ronald W. Davis, Vicki Huff, Curt Scharfe, and Wenyi Wang. Rare variant detection using family-based sequencing analysis. *Proceedings of the National Academy of Sciences*, 110(10):3985, March 2013.

[86] Ryan Poplin, Pi-Chuan Chang, David Alexander, Scott Schwartz, Thomas Colthurst, Alexander Ku, Dan Newburger, Jojo Dijamco, Nam Nguyen, Pegah T. Afshar, Sam S. Gross, Lizzie Dorfman, Cory Y. McLean, and Mark A. DePristo. A universal SNP and small-indel variant caller using deep neural networks. *Nature Biotechnology*, 36(10):983–987, November 2018.

[87] Ryan Poplin, Valentin Ruano-Rubio, Mark A. DePristo, Tim J. Fennell, Mauricio O. Carneiro, Geraldine A. Van der Auwera, David E. Kling, Laura D. Gauthier, Ami Levy-Moonshine, David Roazen, Khalid Shakir, Joel Thibault, Sheila Chandran, Chris Whelan, Monkol Lek, Stacey Gabriel, Mark J. Daly, Ben Neale, Daniel G. MacArthur, and Eric Banks. Scaling accurate genetic variant discovery to tens of thousands of samples. Technical report, bioRxiv, July 2018.

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

[89] Heidi L. Rehm, Sherri J. Bale, Pinar Bayrak-Toydemir, Jonathan S. Berg, Kerry K. Brown, Joshua L. Deignan, Michael J. Friez, Birgit H. Funke, Madhuri R. Hegde, Elaine Lyon, and Working Group of the American College of Medical Genetics and Genomics Laboratory Quality Assurance Commitee. ACMG clinical laboratory standards for next-generation sequencing. *Genetics in Medicine: Official Journal of the American College of Medical Genetics*, 15(9):733–747, September 2013.

[90] Philipp Rentzsch, Max Schubach, Jay Shendure, and Martin Kircher. CADD-Splice—improving genome-wide variant effect prediction using deep learning-derived splice scores. *Genome Medicine*, 13(1):31, February 2021.

[91] Sue Richards, Nazneen Aziz, Sherri Bale, David Bick, Soma Das, Julie Gastier-Foster, Wayne W. Grody, Madhuri Hegde, Elaine Lyon, Elaine Spector, Karl Voelkerding, and Heidi L. Rehm. Standards and guidelines for the interpretation of sequence variants: a joint consensus recommendation of the American College of Medical Genetics and Genomics and the Association for Molecular Pathology. *Genetics in Medicine*, 17(5):405–423, May 2015.

[92] Jared C. Roach, Gustavo Glusman, Arian F. A. Smit, Chad D. Huff, Robert Hubley, Paul T. Shannon, Lee Rowen, Krishna P. Pant, Nathan Goodman, Michael Bamshad, Jay Shendure, Radoje Drmanac, Lynn B. Jorde, Leroy Hood, and David J. Galas. Analysis of Genetic Inheritance in a Family Quartet by Whole-Genome Sequencing. *Science*, 328(5978):636, April 2010.

[93] Eric E. Schadt, Steve Turner, and Andrew Kasarskis. A window into third-generation sequencing. *Human Molecular Genetics*, 19(R2):R227–R240, October 2010.

[94] Iris Schrijver, Nazneen Aziz, Daniel H. Farkas, Manohar Furtado, Andrea Ferreira Gonzalez, Timothy C. Greiner, Wayne W. Grody, Tina Hambuch, Lisa Kalman, Jeffrey A. Kant, Roger D. Klein, Debra G. B. Leonard, Ira M. Lubin, Rong Mao, Narasimhan Nagan, Victoria M. Pratt, Mark E. Sobel, Karl V. Voelkerding, and Jane S. Gibson. Opportunities and challenges associated with clinical diagnostic genome sequencing: a report of the Association for Molecular Pathology. *The Journal of molecular diagnostics: JMD*, 14(6):525–540, November 2012.

[95] Wade L. Schulz, Thomas Durant, Alexa J. Siddon, and Richard Torres. Use of application containers and workflows for genomic data analysis. *Journal of Pathology Informatics*, 7(1):53, January 2016.

[96] Tae Seok Seo, Xiaopeng Bai, Dae Hyun Kim, Qinglin Meng, Shundi Shi, Hameer Ruparel, Zengmin Li, Nicholas J. Turro, and Jingyue Ju. Four-color DNA sequencing by synthesis on a chip using photocleavable fluorescent nucleotides. *Proceedings of the National Academy of Sciences*, 102(17):5926–5931, April 2005.

[97] Rachel M. Sherman, Juliet Forman, Valentin Antonescu, Daniela Puiu, Michelle Daya, Nicholas Rafaels, Meher Preethi Boorgula, Sameer Chavan, Candelaria Vergara, Victor E. Ortega, Albert M. Levin, Celeste Eng, Maria Yazdanbakhsh, James G. Wilson, Javier Marrugo, Leslie A. Lange, L. Keoki Williams, Harold Watson, Lorraine B. Ware, Christopher O. Olopade, Olufunmilayo Olopade, Ricardo R. Oliveira, Carole Ober, Dan L. Nicolae, Deborah A. Meyers, Alvaro Mayorga, Jennifer Knight-Madden, Tina Hartert, Nadia N. Hansel, Marilyn G. Foreman, Jean G. Ford, Mezbah U. Faruque, Georgia M. Dunston, Luis Caraballo, Esteban G. Burchard, Eugene R. Bleecker, Maria I. Araujo, Edwin F. Herrera-Paz, Monica Campbell, Cassandra Foster, Margaret A. Taub, Terri H. Beaty, Ingo Ruczinski, Rasika A. Mathias, Kathleen C. Barnes, and Steven L. Salzberg. Assembly of a pan-genome from deep sequencing of 910 humans of African descent. *Nature Genetics*, 51(1):30–35, January 2019.

[98] Yin Yao Shugart, Yun Zhu, Wei Guo, and Momiao Xiong. Weighted pedigree-based statistics for testing the association of rare variants. *BMC Genomics*, 13(1):667, November 2012.

[99] Yuval B Simons, Michael C Turchin, Jonathan K Pritchard, and Guy Sella. The deleterious mutation load is insensitive to recent population history. *Nature Genetics*, 46(3):220–224, March 2014.

[100] Jouni Sirén. Indexing Variation Graphs. In *2017 Proceedings of the Meeting on Algorithm Engineering and Experiments (ALENEX)*, Proceedings, pages 13–27. Society for Industrial and Applied Mathematics, January 2017.

[101] Jouni Sirén, Erik Garrison, Adam M Novak, Benedict Paten, and Richard Durbin. Haplotype-aware graph indexes. *Bioinformatics*, 36(2):400–407, January 2020.

[102] Jouni Sirén, Jean Monlong, Xian Chang, Adam M. Novak, Jordan M. Eizenga, Charles Markello, Jonas A. Sibbesen, Glenn Hickey, Pi-Chuan Chang, Andrew Carroll, Namrata Gupta, Stacey Gabriel, Thomas W. Blackwell, Aakrosh Ratan, Kent D. Taylor, Stephen S. Rich, Jerome I. Rotter, David Haussler, Erik Garrison, and Benedict Paten. Pangenomics enables genotyping of known structural variants in 5202 diverse genomes. *Science*, December 2021.

[103] Jouni Sirén, Jean Monlong, Adam M. Novak, Jordan M. Eizenga, Charles Markello, Jonas A. Sibbesen, Glenn Hickey, Pi-Chuan Chang, Andrew Carroll, Namrata Gupta, Stacey Gabriel, Thomas W. Blackwell, Aakrosh Ratan, Kent D. Taylor, Stephen S. Rich, Jerome I. Rotter, David Haussler, Erik Garrison, and Benedict Paten. Software and products for "Pangenomics enables genotyping known structural variants in 5,202 diverse genomes", 2021.

[104] Jouni Sirén, Niko Välimäki, and Veli Mäkinen. Indexing Graphs for Path Queries with Applications in Genome Research. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 11(2):375–388, March 2014.

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

[106] Kimberly Splinter, David R. Adams, Carlos A. Bacino, Hugo J. Bellen, Jonathan A. Bernstein, Alys M. Cheatle-Jarvela, Christine M. Eng, Cecilia Esteves, William A. Gahl, Rizwan Hamid, Howard J. Jacob, Bijal Kikani, David M. Koeller, Isaac S. Kohane, Brendan H. Lee, Joseph Loscalzo, Xi Luo, Alexa T. McCray, Thomas O. Metz, John J. Mulvihill, Stanley F. Nelson, Christina G.S. Palmer, John A. Phillips, Leslie Pick, John H. Postlethwait, Chloe Reuter, Vandana Shashi, David A. Sweetser, Cynthia J. Tifft, Nicole M. Walley, Michael F. Wangler, Monte Westerfield, Matthew T. Wheeler, Anastasia L. Wise, Elizabeth A. Worthey, Shinya Yamamoto, and Euan A. Ashley. Effect of Genetic Diagnosis on Patients with Previously Undiagnosed Disease. *New England Journal of Medicine*, 379(22):2131–2139, November 2018.

[107] Amanda B. Spurdle, Sue Healey, Andrew Devereau, Frans B. L. Hogervorst, Alvaro N. A. Monteiro, Katherine L. Nathanson, Paolo Radice, Dominique Stoppa-Lyonnet, Sean Tavtigian, Barbara Wappenschmidt, Fergus J. Couch, David E. Goldgar, and On Behalf Of Enigma. ENIGMA—Evidence-based network for the interpretation of germline mutant alleles: An international initiative to evaluate risk and clinical significance associated with sequence variation in BRCA1 and BRCA2 genes. *Human Mutation*, 33(1):2–7, 2012.

[108] Peter D. Stenson, Matthew Mort, Edward V. Ball, Molly Chapman, Katy Evans, Luisa Azevedo, Matthew Hayden, Sally Heywood, David S. Millar, Andrew D. Phillips, and David N. Cooper. The Human Gene Mutation Database (HGMD®): optimizing its use in a clinical diagnostic or research setting. *Human Genetics*, 139(10):1197–1207, October 2020.

[109] Jae Hoon Sul, Brian E. Cade, Michael H. Cho, Dandi Qiao, Edwin K. Silverman, Susan Redline, and Shamil Sunyaev. Increasing Generality and Power of Rare-Variant Tests by Utilizing Extended Pedigrees. *The American Journal of Human Genetics*, 99(4):846–859, October 2016.

[110] Csilla Szabo, Anthony Masiello, Joseph F. Ryan, The BIC Consortium, and Lawrence C. Brody. The Breast Cancer Information Core: Database design, structure, and scope. *Human Mutation*, 16(2):123–131, 2000.

[111] THE GLOBAL ALLIANCE FOR GENOMICS AND HEALTH. A federated ecosystem for sharing genomic, clinical data. *Science*, 352(6291):1278–1280, June 2016.

[112] Maxime P. Vallée, Tonya L. Di Sera, David A. Nix, Andrew M. Paquette, Michael T. Parsons, Russel Bell, Andrea Hoffman, Frans B. L. Hogervorst, David E. Goldgar, Amanda B. Spurdle, and Sean V. Tavtigian. Adding In Silico Assessment of Potential Splice Aberration to the Integrated Evaluation of BRCA Gene Unclassified Variants. *Human Mutation*, 37(7):627–639, 2016.

[113] Geraldine A. Van der Auwera, Mauricio O. Carneiro, Christopher Hartl, Ryan Poplin, Guillermo del Angel, Ami Levy-Moonshine, Tadeusz Jordan, Khalid Shakir, David Roazen, Joel Thibault, Eric Banks, Kiran V. Garimella, David Altshuler, Stacey Gabriel, and Mark A. DePristo. From FastQ Data to High-Confidence Variant Calls: The Genome Analysis Toolkit Best Practices Pipeline. *Current Protocols in Bioinformatics*, 43(1):11.10.1–11.10.33, 2013.

[114] John Vivian, Arjun Arkal Rao, Frank Austin Nothaft, Christopher Ketchum, Joel Armstrong, Adam Novak, Jacob Pfeil, Jake Narkizian, Alden D Deran, Audrey Musselman-Brown, Hannes Schmidt, Peter Amstutz, Brian Craft, Mary Goldman, Kate Rosenbloom, Melissa Cline, Brian O'Connor, Megan Hanna, Chet Birger, W James Kent, David A Patterson, Anthony D Joseph, Jingchun Zhu, Sasha Zaranek, Gad Getz, David Haussler, and Benedict Paten. Toil enables reproducible, open source, big biomedical data analyses. *Nature Biotechnology*, 35(4):314–316, April 2017.

[115] Kate Voss, Jeff Gentry, and Geraldine Van der Auwera. Full-stack genomics pipelining with GATK4 + WDL + Cromwell. *F1000Research*, 6, August 2017.

[116] Justin Wagner, Nathan D. Olson, Lindsay Harris, Jennifer McDaniel, Haoyu Cheng, Arkarachai Fungtammasan, Yih-Chii Hwang, Richa Gupta, Aaron M. Wenger, William J. Rowell, Ziad M. Khan, Jesse Farek, Yiming Zhu, Aishwarya Pisupati, Medhat Mahmoud, Chunlin Xiao, Byunggil Yoo, Sayed Mohammad Ebrahim Sahraeian, Danny E. Miller, David Jáspez, José M. Lorenzo-Salazar, Adrián Muñoz-Barrera, Luis A. Rubio-Rodríguez, Carlos Flores, Giuseppe Narzisi, Uday Shanker Evani, Wayne E. Clarke, Joyce Lee, Christopher E. Mason, Stephen E. Lincoln, Karen H. Miga, Mark T. W. Ebbert, Alaina Shumate, Heng Li, Chen-Shan Chin, Justin M. Zook, and Fritz J. Sedlazeck. Towards a Comprehensive Variation Benchmark for Challenging Medically-Relevant Autosomal Genes. Technical report, October 2021.

[117] Justin Wagner, Nathan D. Olson, Lindsay Harris, Jennifer McDaniel, Ziad Khan, Jesse Farek, Medhat Mahmoud, Ana Stankovic, Vladimir Kovacevic, Byunggil Yoo, Neil Miller, Jeffrey A. Rosenfeld, Bohan Ni, Samantha Zarate, Melanie Kirsche, Sergey Aganezov, Michael Schatz, Giuseppe Narzisi, Marta Byrska-Bishop, Wayne Clarke, Uday S. Evani, Charles Markello, Kishwar Shafin, Xin Zhou, Arend Sidow, Vikas Bansal, Peter Ebert, Tobias Marschall, Peter Lansdorp, Vincent Hanlon, Carl-Adam Mattsson, Alvaro Martinez Barrio, Ian T. Fiddes, Chunlin Xiao, Arkarachai Fungtammasan, Chen-Shan Chin, Aaron M. Wenger, William J. Rowell, Fritz J. Sedlazeck, Andrew Carroll, Marc Salit, and Justin M. Zook. Benchmarking challenging small variants with linked and long reads. Technical report, October 2021.

[118] Klaudia Walter, Josine L. Min, Jie Huang, Lucy Crooks, Yasin Memari, Shane McCarthy, John R. B. Perry, ChangJiang Xu, Marta Futema, Daniel Lawson, Valentina Iotchkova, Stephan Schiffels, Audrey E. Hendricks, Petr Danecek, Rui Li, James Floyd, Louise V. Wain, Inês Barroso, Steve E. Humphries, Matthew E. Hurles, Eleftheria Zeggini, Jeffrey C. Barrett, Vincent Plagnol, J. Brent Richards, Celia M. T. Greenwood, Nicholas J. Timpson, Richard Durbin, Nicole Soranzo, Senduran Bala, Peter Clapham, Guy Coates, Tony Cox, Allan Daly, Petr Danecek, Yuanping Du, Richard Durbin, Sarah Edkins, Peter Ellis, Paul Flicek, Xiaosen Guo, Xueqin Guo, Liren Huang, David K. Jackson, Chris Joyce, Thomas Keane, Anja Kolb-Kokocinski, Cordelia Langford, Yingrui Li, Jieqin Liang, Hong Lin, Ryan Liu, John Maslen, Shane McCarthy, Dawn Muddyman, Michael A. Quail, Jim Stalker, Jianping Sun, Jing Tian, Guangbiao Wang, Jun Wang, Yu Wang, Kim Wong, Pingbo Zhang, Inês Barroso, Ewan Birney, Chris Boustred, Lu Chen, Gail Clement, Massimiliano Cocca, Petr Danecek, George Davey Smith, Ian N. M. Day, Aaron Day-Williams, Thomas Down, Ian Dunham, Richard Durbin, David M. Evans, Tom R. Gaunt, Matthias Geihs, Celia M. T. Greenwood, Deborah Hart, Audrey E. Hendricks, Bryan Howie, Jie Huang, Tim Hubbard, Pirro Hysi, Valentina Iotchkova, Yalda Jamshidi, Konrad J. Karczewski, John P. Kemp, Genevieve Lachance, Daniel Lawson, Monkol Lek, Margarida Lopes, Daniel G. MacArthur, Jonathan Marchini, Massimo Mangino, Iain Mathieson, Shane McCarthy,

Yasin Memari, Sarah Metrustry, Josine L. Min, Alireza Moayyeri, Dawn Muddyman, Kate Northstone, Kalliope Panoutsopoulou, Lavinia Paternoster, John R. B. Perry, Lydia Quaye, J. Brent Richards, Susan Ring, Graham R. S. Ritchie, Stephan Schiffels, Hashem A. Shihab, So-Youn Shin, Kerrin S. Small, María Soler Artigas, Nicole Soranzo, Lorraine Southam, Timothy D. Spector, Beate St Pourcain, Gabriela Surdulescu, Ioanna Tachmazidou, Nicholas J. Timpson, Martin D. Tobin, Ana M. Valdes, Peter M. Visscher, Louise V. Wain, Klaudia Walter, Kirsten Ward, Scott G. Wilson, Kim Wong, Jian Yang, Eleftheria Zeggini, Feng Zhang, Hou-Feng Zheng, Richard Anney, Muhammad Ayub, Jeffrey C. Barrett, Douglas Blackwood, Patrick F. Bolton, Gerome Breen, David A. Collier, Nick Craddock, Lucy Crooks, Sarah Curran, David Curtis, Richard Durbin, Louise Gallagher, Daniel Geschwind, Hugh Gurling, Peter Holmans, Irene Lee, Jouko Lönnqvist, Shane McCarthy, Peter McGuffin, Andrew M. McIntosh, Andrew G. McKechanie, Andrew McQuillin, James Morris, Dawn Muddyman, Michael C. O'Donovan, Michael J. Owen, Aarno Palotie, Jeremy R. Parr, Tiina Paunio, Olli Pietilainen, Karola Rehnström, Sally I. Sharp, David Skuse, David St Clair, Jaana Suvisaari, James T. R. Walters, Hywel J. Williams, Inês Barroso, Elena Bochukova, Rebecca Bounds, Anna Dominiczak, Richard Durbin, I. Sadaf Farooqi, Audrey E. Hendricks, Julia Keogh, Gaëlle Marenne, Shane McCarthy, Andrew Morris, Dawn Muddyman, Stephen O'Rahilly, David J. Porteous, Blair H. Smith, Ioanna Tachmazidou, Eleanor Wheeler, Eleftheria Zeggini, Saeed Al Turki, Carl A. Anderson, Dinu Antony, Inês Barroso, Phil Beales, Jamie Bentham, Shoumo Bhattacharya, Mattia Calissano, Keren Carss, Krishna Chatterjee, Sebahattin Cirak, Catherine Cosgrove, Richard Durbin, David R. Fitzpatrick, James Floyd, A. Reghan Foley, Christopher S. Franklin, Marta Futema, Detelina Grozeva, Steve E. Humphries, Matthew E. Hurles, Shane McCarthy, Hannah M. Mitchison, Dawn Muddyman, Francesco Muntoni, Stephen O'Rahilly, Alexandros Onoufriadis, Victoria Parker, Felicity Payne, Vincent Plagnol, F. Lucy Raymond, Nicola Roberts, David B. Savage, Peter Scambler, Miriam Schmidts, Nadia Schoenmakers, Robert K. Semple, Eva Serra, Olivera Spasic-Boskovic, Elizabeth Stevens, Margriet van Kogelenberg, Parthiban Vijayarangakannan, Klaudia Walter, Kathleen A. Williamson, Crispian Wilson, Tamieka Whyte, Antonio Ciampi, Celia M. T. Greenwood, Audrey E. Hendricks, Rui Li, Sarah Metrustry, Karim Oualkacha, Ioanna Tachmazidou, ChangJiang Xu, Eleftheria Zeggini, Martin Bobrow, Patrick F. Bolton, Richard Durbin, David R. Fitzpatrick, Heather Griffin, Matthew E. Hurles, Jane Kaye, Karen Kennedy, Alastair Kent, Dawn Muddyman, Francesco Muntoni, F. Lucy Raymond, Robert K. Semple, Carol Smee, Timothy D. Spector, Nicholas J. Timpson, Ruth Charlton, Rosemary Ekong, Marta Futema, Steve E. Humphries, Farrah Khawaja, Luis R. Lopes, Nicola Migone, Stewart J. Payne, Vincent Plagnol, Rebecca C. Pollitt, Sue Povey, Cheryl K. Ridout, Rachel L. Robinson, Richard H. Scott, Adam Shaw, Petros Syrris, Rohan Taylor, Anthony M. Vandersteen, Jeffrey C. Barrett, Inês Barroso, George Davey Smith, Richard Durbin, I. Sadaf Farooqi, David R. Fitzpatrick, Matthew E. Hurles, Jane Kaye, The UK10K Consortium, Writing group, Production group, Cohorts group, Neurodevelopmental disorders group, Obesity group, Rare disease group, Statistics group, Ethics group, Incidental findings group, and Management committee. The UK10K project identifies rare variants in health and dis-

ease. *Nature*, 526(7571):82–90, October 2015.

[119] Joachim Weischenfeldt, Orsolya Symmons, François Spitz, and Jan O. Korbel. Phenotypic impact of genomic structural variation: insights from and for human disease. *Nature Reviews Genetics*, 14(2):125–138, February 2013.

[120] Aaron M. Wenger, Paul Peluso, William J. Rowell, Pi-Chuan Chang, Richard J. Hall, Gregory T. Concepcion, Jana Ebler, Arkarachai Fungtammasan, Alexey Kolesnikov, Nathan D. Olson, Armin Töpfer, Michael Alonge, Medhat Mahmoud, Yufeng Qian, Chen-Shan Chin, Adam M. Phillippy, Michael C. Schatz, Gene Myers, Mark A. DePristo, Jue Ruan, Tobias Marschall, Fritz J. Sedlazeck, Justin M. Zook, Heng Li, Sergey Koren, Andrew Carroll, David R. Rank, and Michael W. Hunkapiller. Accurate circular consensus long-read sequencing improves variant detection and assembly of a human genome. *Nature Biotechnology*, 37(10):1155–1162, October 2019.

[121] Andy B. Yoo, Morris A. Jette, and Mark Grondona. SLURM: Simple Linux Utility for Resource Management. In Dror Feitelson, Larry Rudolph, and Uwe Schwiegelshohn, editors, *Job Scheduling Strategies for Parallel Processing*, Lecture Notes in Computer Science, pages 44–60, Berlin, Heidelberg, 2003. Springer.

[122] Taedong Yun, Helen Li, Pi-Chuan Chang, Michael F Lin, Andrew Carroll, and Cory Y McLean. Accurate, scalable cohort variant calls using DeepVariant and GLnexus. *Bioinformatics*, 36(24):5582–5589, December 2020.