Title

Read Mapping, Variant Calling, and Copy Number Variation Detection in Segmental Duplications

Author

Prodanov, Timofey

Publication Date

2022

UNIVERSITY OF CALIFORNIA SAN DIEGO

Read Mapping, Variant Calling, and Copy Number Variation Detection in Segmental Duplications

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

Doctor of Philosophy

in

Bioinformatics and Systems Biology

by

Timofey Prodanov

Committee in charge:

Professor Vikas Bansal, Chair
Professor Melissa Gymrek, Co-Chair
Professor Vineet Bafna
Professor Siavash Mir Arabbaygi
Professor Pavel Pevzner

2022

The Dissertation of Timofey Prodanov is approved, and it is acceptable in quality and form for publication on microfilm and electronically.

University of California San Diego

2022

DEDICATION

This work is dedicated

to my parents Zhanna Etsina and Petr Prodanov,

to my brother Daniil Prodanov,

and to my dearest wife Polina Lipaeva.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I would like to thank my advisor, Prof. Vikas Bansal for supervising me through my Ph.D. program. Vikas is an outstanding mentor and scientist, and he has helped me greatly to grow as a scientist and to overcome any difficulties I encountered during my studies.

I would like to thank Prof. Pavel Pevzner and Prof. Yana Safonova for encouraging me to apply to the University of California San Diego and for helping me during the start of my Ph.D. program. I would like to thank Prof. Vineet Bafna, Prof. Melissa Gymrek and Prof. Siavash Mir Arabbaygi for their assistance and crucial feedback.

I would like to thank my wife Polina Lipaeva for her invaluable help and support and would like to thank all my friends across the world for being there for me.

Chapter 2, in full, is a reformatted reprint of "Sensitive alignment using paralogous sequence variants improves long-read mapping and variant calling in segmental duplications" as it appears in *Nucleic Acids Research* (2020) by Timofey Prodanov and Vikas Bansal. The dissertation author was the primary author of this paper.

Chapter 3, in full, is a reformatted reprint of "Robust and accurate estimation of paralog-specific copy number for duplicated genes using whole-genome sequencing" as it appears in *Nature Communications* (2022) by Timofey Prodanov and Vikas Bansal. The dissertation author was the primary author of this paper.

Chapter 4, in part, is currently being prepared for submission for publication of the paper by Timofey Prodanov and Vikas Bansal. The dissertation author was the primary author of this paper.

VITA

2011–2015 Bachelor of Science,
Saint Petersburg University, Saint Petersburg, Russia.

2015–2017 Master of Science,
Saint Petersburg Academic University, Saint Petersburg, Russia.

2017–2022 Doctor of Philosophy in Bioinformatics & Systems Biology,
University of California San Diego, La Jolla, CA, USA.

PUBLICATIONS

Prodanov, T. & Bansal, V. Robust and accurate estimation of paralog-specific copy number for duplicated genes using whole-genome sequencing.
Nature Communications **13**, 3221 (2022), doi.org/10.1038/s41467-022-30930-3

Prodanov, T. & Bansal, V. Sensitive alignment using paralogous sequence variants improves long-read mapping and variant calling in segmental duplications.
Nucleic Acids Research **48**, e114 (2020), doi.org/10.1093/nar/gkaa829

Ivanov, M., Matsvay, A., Glazova, O., Krasovskiy, S., Usacheva, M., Amelina, E., Chernyak, A., Ivanov, M., Musienko, S., Prodanov, T., Kovalenko, S., Baranova, A. & Khafizov, K. Targeted sequencing reveals complex, phenotype-correlated genotypes in cystic fibrosis.
BMC Medical Genomics **11**, 13 (2018), doi.org/10.1186/s12920-018-0328-z

Prodanov, T. Adaptive randomized algorithms for community detection in graphs.
Stochastic Optimization in Informatics **11**, 29–54 (2015).

ABSTRACT OF THE DISSERTATION

Read Mapping, Variant Calling, and Copy Number Variation Detection in Segmental Duplications

by

Timofey Prodanov

Doctor of Philosophy in Bioinformatics and Systems Biology

University of California San Diego, 2022

Professor Vikas Bansal, Chair
Professor Melissa Gymrek, Co-Chair

Segmental duplications or low-copy repeats (LCRs) are long segments of duplicated DNA that cover more than 5% of the human genome and overlap more than 600 protein-coding genes. Copy number and sequence variants in over 150 such duplicated genes (e.g. *SMN1/2*, *STRC*, *NCF1*) are associated with risk for rare and complex human diseases. Paralogous sequence variants (PSVs) are short differences between homologous sequences within duplicated loci. It has been shown that many PSVs are not fixed in the population, which reduces their potential to differentiate paralogous regions. Moreover, segmental duplications exhibit extensive copy number variation, and are characterized by poor read mappability even for long-read data. All these factors lead to diminished accuracy of existing bioinformatical tools for short- and long-read data in duplicated regions. This dissertation presents three novel computational methods that solve classical bioinformatical problems (read mapping, variant calling and copy number variation detection) in LCR regions. In contrast to existing tools, three proposed methods examine PSV genotypes in order to distinguish sets of reliable and unreliable PSVs, and use reliable PSVs to achieve higher accuracy than state-of-the-art methods in the field.

First, we describe a probabilistic method, DuploMap, designed to improve the accuracy of long-read mapping within LCR regions. It iteratively genotypes PSVs and leverages reliable PSVs to distinguish between candidate read locations. This allows for high accuracy variant calling in segmental duplications using long reads. Next, we present the first toolkit for LCR regions, Parascopy. Parascopy uses short-read whole-genome sequencing to estimate total copy number as well as paralog-specific copy number for duplicated genes. Parascopy analyzes reads mapped to different repeat copies and utilizes multiple samples to mitigate sequencing bias and identify reliable PSVs. Accurate copy number estimation facilitates discovery of pathogenic copy number changes in duplicated genes. A novel variant caller, ParascopyVC, builds upon copy number variation detection and uses short-read data to call pooled and locus-specific variants within segmental duplications. ParascopyVC uses population allele frequencies and pooled genotypes to select informative PSVs. Finally, the tool uses informative PSVs to identify additional locus-specific variants, enabling the discovery of novel disease-causing variants in duplicated genes.

CHAPTER 1

General Introduction and Scope of the Thesis

1.1 The human genome

The human genome is a collection of 23 pairs of chromosomes encoded as Deoxyribonucleic acid (DNA) within cell nuclei. DNA consists of two strands made of multiple small subunits called nucleotides: adenine (A), cytosine (C), guanine (G) and thymin (T). Each of the nucleotides forms a *base pair* with a complement nucleotide (A with T; C with G) on the opposite strand. Two strands of DNA are called *sense* ('+' strand) and *antisense* ('-' strand), and strand sequences are reverse-complement to each other.

Human reference genome is a set consisting of multiple sequences over an alphabet 'A,C,G,T', where each sequence encodes a contiguous stretch of human DNA. Ideally, each reference sequence would store a full chromosome, however, most modern human reference genomes contain gaps within chromosome sequences (denoted by letter 'N' and other symbols), and include short "contig" strings in addition to the chromosome sequences.

During the last decades, significant efforts have been undertaken to create a reference human genome: the Genome Reference Consortium Human Build 37 (GRCh37) and 38 (GRCh38) were published in February 2009 and December 2013, respectively. The two most recent reference genome patches GRCh38.p13 and GRCh38.p14 were published in February 2019 and May 2022, while a number of alternative human reference genomes emerged, such as T2T-CHM13 assembly [1]. In total, human reference genomes contain ≈ 3.1 billion nucleotides. Traditionally, human reference genomes are assembled using a single individual (or a small group of individuals) and contain a haploid collection of chromosomes — each autosomal chromosome appears in the reference genome once.

Most human cells contain a diploid set of chromosomes — 22 autosomal chromosomes are present in pairs of *homologous* chromosomes, while sex chromosomes appear either as a pair of X chromosomes, or as individual X and Y chromosomes. Each individual human genome is different from the reference genome, and all pairs of homologous chromosomes differ from each other. In general, an individual human genome has approximately 4–5 million sequence variants [2]. Such variants represent short (one or several nucleotides) substitutions, insertions and deletions compared to the reference genome sequence. Possible sequences of a variant are called *alleles*. Variants with the same allele on both homologous chromosomes are called *homozygous* and variants with different alleles are called *heterozygous*.

1.2 Segmental duplications and structural variations

The reference genome contains a large number of segmental duplications (also known as low-copy repeats, LCRs) — long segments of DNA that appear in the genome multiple times with high sequence similarity [3–5]. Copies of a segmental duplication are often called *repeat copies* or *homologous regions* of the duplication. Typically, segmental duplications are by definition longer than 1 kilobase (kb) and have sequence similarity over 90%. Segmental duplications can appear on different chromosomes (interchromosomal), or on the same chromosome (intrachromosomal). In some duplications, repeat copies follow each other with only

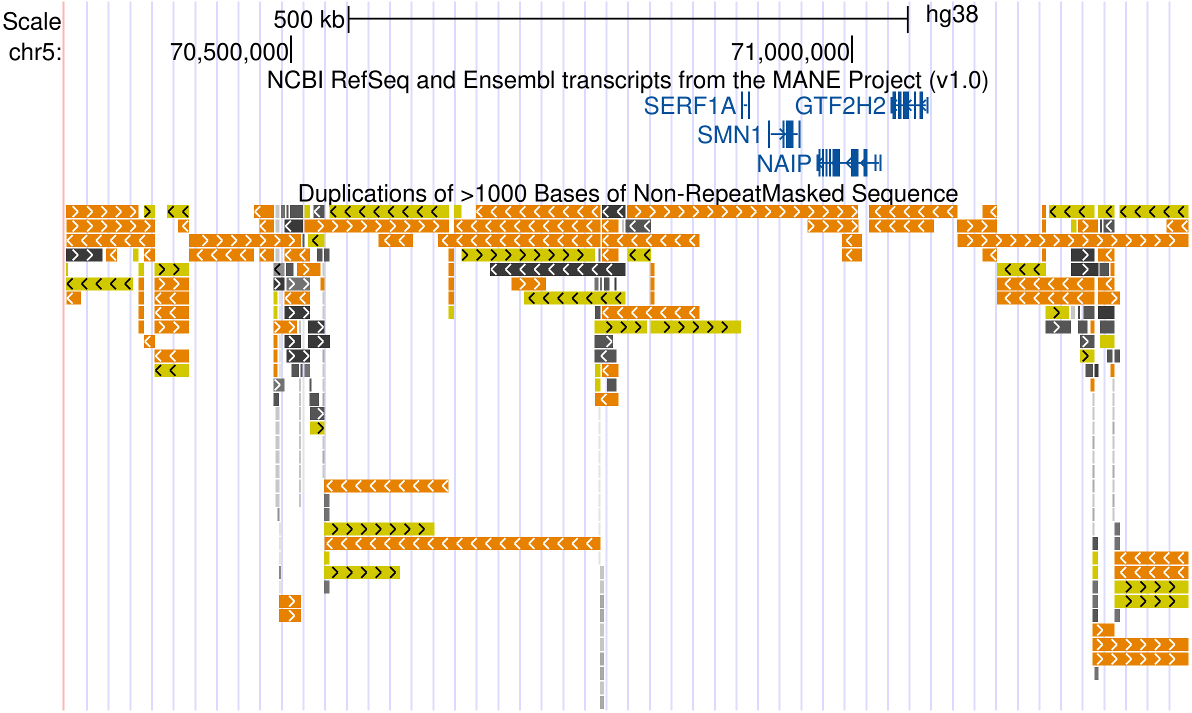

Scale
chr5:
500 kb
hg38
70,500,000
71,000,000
NCBI RefSeq and Ensembl transcripts from the MANE Project (v1.0)
SERF1A
GTF2H2
SMN1
NAIP
Duplications of >1000 Bases of Non-RepeatMasked Sequence

Figure 1.1. **Genome browser view of a complex set of segmental duplications.** UCSC genome browser [6] shows a 1 Mb locus (chr5:70,300,001-71,300,000) containing duplicated genes *SMN1*, *SERF1A*, *NAIP* and *GTF2H2*, shown with blue lines. Each colored bar on the bottom of the plot denotes a duplicated region longer than 1 kb. Color of the bar indicates sequence similarity of the duplication: 90 to 98% shown with black and gray; 98 to 99% shown with yellow; over 99% shown with orange.

a small stretch of non-duplicated DNA in between (tandem segmental duplications). Repeat copies can appear on the same or on the opposite DNA strands.

Even though repeat copies of a segmental duplication can be completely identical, often there exist at least several paralogous sequence variants (PSVs). Each PSV represents a small difference between the sequences of the repeat copies, such as a short substitution, insertion or deletion. Note, that in a duplication with more than two repeat copies, it is not guaranteed that a PSV would have a separate allele for each repeat copy.

In addition to sequence variants, an individual human genome differs from the reference genome by approximately 1.5% when considering structural variants (SVs) [7, 8]. Structural variants include deletions, duplications, insertions, inversions and translocations, all of which can vary in size from 50 base pairs (bp) to over a million base pairs (megabase, Mb). Structural variants of different types are commonly associated with segmental duplications [9], in particular, segmental duplications are prone to extensive copy number variations. Consequently, long history of structural variation in a relatively short genomic region leads to complex duplication structures, (See Figure 1.1 for an example).

1.3 Duplicated genes in the human genome

Human genome stores vast amounts of functional and hereditary information. The main functional genomic units are *genes*, which vary in size from a few hundred base pairs to few million base pairs. In total, the human genome contains over 20 thousands genes. Sequence variants and structural variants in genes can have both beneficial and adverse effect on the human health, potentially leading to genetic diseases.

Genes within duplicated regions raise additional interest due to several factors: (i) duplicated genes play a vital role in primate evolution [4]; (ii) structural variants, including partial or full gene deletion and insertions, are more common in the duplicated genes; (iii) sequence variants and structural variants within the duplicated genes are challenging to both detect and to pinpoint to the correct repeat copy due to the repetitive nature of the LCR regions.

1.3.1 *SMN1/2* duplicated locus

One of the most studied duplicated genes is *SMN1*. It is located in a 200 kb-long duplication with sequence similarity over 99.8%. The coding region of the *SMN1* gene is approximately 29 kb long and contains nine exons, designated historically as exons 1, 2a, 2b and 3–8. The second repeat copy of the duplication contains a protein-coding gene *SMN2*, which differs from the *SMN1* gene by just 24 PSVs — less than one PSV per kilobase. Moreover, previous studies have shown that just 8 PSV sites are non-polymorphic in the human population [10].

Sequence variants and copy number changes in the *SMN1/2* genes are associated with *Spinal Muscular Atrophy* (SMA) [10] — serious genetic neuromuscular disorder that results in muscle atrophy with potentially lethal consequences. Paralog-specific copy number of the duplication can modify the disease phenotype: an individual with two copies of the *SMN1* paralog and one copy of the *SMN2* is healthy, while an individual with one *SMN1* copy and two *SMN2* copies is the disease carrier or is affected by the disease.

1.4 DNA sequencing technologies

DNA sequencing became possible with the development of the Sanger method in 1977 [11]. Sanger sequencing generates highly accurate reads with length up to 900 bp, but has several downsides, mainly high sequencing cost and slow and labor-intensive sequencing process. Between 1990 and 2003 twenty laboratories collaborated in order to sequence a single human individual [12] at a cost of approximately 2.7 billion dollars.

In 2005, the 454 GS 20 Roche sequencing platform [13] became available and gave rise to the *Next-Generation Sequencing* (NGS), also called "short-read sequencing". A year later, Solexa launched the Genome Analyzer sequencer, which first utilized Sequencing by Synthesis technology. In 2007, Illumina acquired Solexa and its technology [14], and has developed a number of new sequencing platforms since then. NGS technologies have several advantages compared to the Sanger sequencing: (i) lower sequencing cost; (ii) ability to sequence the whole genome with high coverage; (iii) capability to produce more data with the same amount of input DNA.

In general, Next-Generation Sequencing reads are characterized by smaller read length ($\leq$ 300 bp) and slightly higher error rate (0.1%) compared to the Sanger sequencing. In addition, many Illumina sequencers support Paired-End Sequencing: technology that allows to sequence a DNA fragment from both sides and generate pairs of "linked" reads. For example, Illumina HiSeq 2500 platform allows to sequence pairs of reads with length 2×250 bp and insert size (length of the fragment) up to 800 bp. Various library preparation techniques allow to increase insert size to multiple kilobases.

In late 2000s, a range of new technologies appeared, together known as *Third Generation Sequencing* or Single Molecule Sequencing (SMS) [14, 15]. Two most widely known Third Generation Sequencing providers are Pacific Biosciences (PacBio) with the single molecule real time (SMRT) platform [16] and Oxford Nanopore Technologies (ONT) [17] with MinION, GridION and PromethION platforms. Third Generation Sequencing datasets can be characterized by large read length ($>$ 10 kb) [14] and high error rate (5–10%) [15, 18]. Recently, Oxford Nanopore presented a protocol to generate ultra-long reads with lengths reaching 800 kb (N50 $>$ 100 kb) [19], while Pacific Biosciences presented single molecule high-fidelity (HiFi) technology [20] that generates 10–25 kb reads with error rate $<$ 0.5%.

In 2017, 10X Genomics presented a technology [21–23] that can be characterized as both the Second and the Third Generation Sequencing. The 10X Genomics' Linked-Reads sequencing technology places long DNA molecules (50–150 kb) into separate droplets. The molecules are later split into smaller fragments, barcoded separately for each droplet, and sequenced using Illumina sequencing technologies. Resulting linked-reads benefit from both high accuracy Illumina protocols and long fragment information, and are often used for structural variation detection [21, 22]. Nevertheless, 10X Genomics discontinued the Linked-Reads technology in 2020.

1.5 Challenges of modern bioinformatics

1.5.1 Read mapping

The Second and Third generation DNA sequencing technologies generate reads originating from random genomic locations. Depending on the library preparation protocol, the set of genomic regions can be bounded (for example to cover only exons), but read locations are nevertheless unknown in advance. The problem of placing sequencing reads to their original genomic locations is called *read mapping* or *read alignment*. Several factors increase read mapping complexity: (i) sequencing reads are imperfect and each read sequence may contain one or several substitutions, insertions or deletions compared to the true genomic sequence; (ii) individual genomic sequence is different from the reference genome; (iii) a single read may have several possible genomic locations, which share significant similarity or may be identical; (iv) number of sequencing reads may reach and exceed a billion, therefore read mapping algorithms need to be very time efficient.

Since 1970s there arose multiple efficient algorithms for finding a substring of length m in a longer string of length n: Knuth–Morris–Pratt [24] performs the search in $O(n)$, while the suffix tree [25] and the suffix array [26] algorithms require a preprocessing step (indexing), but perform a faster search in $O(m)$. Burrows-Wheeler transform [27] and FM-index [28] allow to compress and index long strings in $O(n)$, and are actively used in Bowtie [29], Bowtie2 [30] and BWA [31, 32] read mapping tools. Note that all string-searching algorithms need to be applied to the read subsequences (known as seeds) due to the presence of sequencing errors and sequence variants.

In addition to finding an approximate read location, read mapping tools need to construct an alignment between the reference sequence and the read sequence — placing the minimal number of insertions, deletions and substitutions needed to match the two sequences. Several algorithms solve this problem, namely Needleman–Wunsch [33], Smith–Waterman [34]; and Pair Hidden Markov Models [35] (Pair-HMM), all requiring $O(nm)$ running time. In order to tackle the quadratic complexity of the alignment algorithms, multiple optimizations were developed, including parallel processing [30], striped Smith–Waterman algorithm [36] and Wavefront algorithm [37].

Read mapping tools that work with the Third Generation sequencing data encounter even more problems: higher error rate leads to fewer matches between the read and reference sequences, while bigger read length makes it difficult for the tools to keep high processing speed. Nevertheless, the workflow of the most popular long-read mapping tool Minimap2 [38, 39] is reminiscent of the short-read workflows: first, Minimap2 finds matches between the read seeds and the reference, then it chains sets of nearby matches, and, finally, completes the alignment between the seeds in each chain.

1.5.2 Variant calling

The process of identifying sequence variants from the sequencing data is called *variant calling*. Briefly, variant calling algorithms contain two important steps: variant discovery and genotyping. During the variant discovery step, a variant caller analyses read mappings for one or more samples, and finds genomic positions (sites) that have a significant number of reads that support a non-reference allele. Input read alignments are sorted by the genomic coordinate in advance and indexed, which allows for quick access to all reads overlapping a specific site. Due to the low error rate, variant discovery for the Second Generation Sequencing data can often be done by the simple count of the number of reads that support each non-reference allele at each genomic position. Nevertheless, the true complexity of the process can be seen in case of low-complexity variants, as well as long insertions or deletions, as the reads have higher chance to have an error within the variant or to cover the variant only partially. The variant discovery step becomes even more complex in the presence of high error rate or low read depth.

During the variant genotyping step, variant callers assign most probable genotypes to each variant. In a simple diploid case a variant with two alleles has three possible *unphased* genotypes: 0/0, 0/1 and 1/1. Some variant calling tools perform an additional haplotyping step, which produces *phased* genotypes (in this example 0|0, 0|1, 1|0 and 1|1) that link alleles of the nearby variants, indicating that they lie either on the same chromosome or on different homologous chromosomes. Note that the number of possible genotypes raises exponentially with the number of alleles and the ploidy (number of homologous chromosomes). Variant callers assign genotypes based on the reads that overlap the variant, and employ a wide range of computational and statistical techniques, including Multinomial distribution and Bayesian approach in FreeBayes [40]; Pair-HMM in GATK [41] and Longshot [42]; and deep Neural Networks in DeepVariant [43]. In addition to the read–variant observations, variant callers incorporate external information, such as population frequencies of the known variants and pedigree information.

1.5.3 Copy number variation detection

Similarly to variant callers, copy number variation (CNV) detection tools use existing read mappings as input, and search for genomic sites that exhibit non-reference alleles. However, in case of CNVs, genomic sites are longer (hundreds to millions base pairs), while CNV alleles represent the number of times the sequence appears in an individual genome. In contrast to variant calling tools, CNV-detection tools are rarely able to determine the variant allele (locus copy number) for each homologous chromosome, and instead determine sum copy number across the two homologous chromosomes together. In case of the low-copy repeats, two terms can be defined: paralog-specific copy number — copy number value for each repeat copy of the low-copy repeat; and aggregate copy number — sum copy number across all repeat copies. CNV-detection tools can be split into three categories by the set of regions they analyse: (i) whole-genome analysis as in CNVnator [44, 45] and QuicK-mer2 [46]; (ii) targeted analysis as in GenomeSTRIP [47]; and (iii) single gene analysis as in SMNCopyNumberCaller [10].

All CNV-detection tools calculate background read depth at different GC-content values across the whole genome or across a specific set of genomic loci. Then, the tools compare the read depth distribution with the background read depth values in order to discover copy number variations. Specifically, CNVnator [44] and GenomeSTRIP [47] use Gaussian distribution, while QuicK-mer2 [46] uses an unnamed read depth distribution at unique genomic k-mers.

SMNCopyNumberCaller [10] uses Gaussian mixture models to predict aggregate copy number and then uses PSV-allelic read depth to predict *SMN1/2* paralog-specific copy number.

1.6 Scope of the thesis

Despite the vast amount of bioinformatical algorithms and tools that focus on the analysis of the whole-genome sequencing data, there is a lack of tools designed to withstand the difficulties of the low-copy repeats. Segmental duplications are often deliberately removed from the analysis; and, as a direct result, many duplicated genes remain understudied.

Chapter 2 introduces an algorithm, DuploMap, that takes an existing long-read mappings, and refines the alignments that overlap low-copy repeats. Specifically, DuploMap iteratively genotypes PSVs and updates read locations. Refined long-read mappings are more accurate that the original read mappings on the simulated data, and produce variant calls with F_1 scores for the HG002 WGS dataset.

Chapter 3 presents a copy number variation detection tool Parascopy. Parascopy uses short-read WGS data from multiple samples to first find aggregate copy number values, and then searches for a set of reliable PSVs to estimate sample paralog-specific copy number across various repeat copies. Parascopy produces more robust and accurate copy number estimates compared to other CNV-detection methods.

Chapter 4 describes a variant caller ParascopyVC, which is designed to call variants within low-copy repeats. ParascopyVC uses copy number estimates and obtained by Parascopy and calls variant genotypes, pooled across all repeat copies. Next, ParascopyVC finds informative PSVs based on the pooled PSV genotypes and the population allele frequencies, and uses them to call paralog-specific variants.

DuploMap, Parascopy and ParascopyVC are specifically designed for low-copy repeats. In contrast to existing methods, they analyze reads mapped to various repeat copies simultaneously, and estimate PSV genotypes in order to identify subsets of reliable PSVs. All three methods do not require whole-genome remapping and use multiple parallel threads to produce efficient, scalable and accurate solutions to classical bioinformatical problems within low-copy repeats.

CHAPTER 2

Long-Read Mapping in Segmental Duplications

2.1 Introduction

High-throughput short-read sequencing technologies have transformed the study of genetic variation and the discovery of disease-associated variants for human disorders. However, the short read lengths (typically a few hundred bases) of short-read technologies such as Illumina limit the comprehensive detection of genetic variation [48]. The human genome is highly repetitive and contains several types of repetitive sequences including hundreds of long segmental duplications (ranging in length from a few kilobases to hundreds of kilobases) that have greater than 98% sequence similarity to other sequences [3, 49]. Some of these duplicated sequences are perfectly identical to their paralogous sequences over several kilobases. Duplications with length at least 10 kilobases and sequence identity of 98% or greater cover 3.0–3.2% of the human genome and overlap more than 800 protein-coding genes. Variants in many of these genes are implicated in rare Mendelian disorders as well as complex diseases [50]. Some examples of such duplicated genes are *PMS2* in Lynch syndrome [51], *STRC* in hearing loss [52], and *NCF1* [53] in autoimmune diseases. From the perspective of whole-genome sequencing, many of these segmental duplications are partially or completely inaccessible to short reads since the vast majority of reads originating from such regions cannot be unambiguously aligned to the genome [50, 54]. This limits the discovery of disease-associated mutations and our understanding of the function of these genes.

In recent years, two single molecule sequencing (SMS) technologies that can generate reads that are tens to hundreds of kilobases long have become widely available. The Pacific Biosciences (PacBio) SMRT technology can generate reads that are, on average, 10-60 kilobases long [55]. Another long read sequencing technology – Oxford Nanopore (ONT) MinION – can generate long reads with lengths that can even exceed a megabase in length [19]. The availability of these technologies has dramatically altered the ability to assemble bacterial and mammalian genomes since the long read lengths can resolve long repeats present in genomes [56]. The throughput and read lengths for these third-generation sequencing technologies continues to improve, as a result, these technologies are increasingly being used to sequence human genomes [57, 58]. The long read lengths of these technologies provide several advantages for sequencing human genomes compared to short reads. These include the ability to de novo assemble genomes with high contiguity [19, 59], reconstruct haplotypes directly from the sequence reads [42, 60] and increased sensitivity for the detection of structural variants [57, 61].

A key advantage of long SMS reads is their ability to map unambiguously in repetitive regions of the genome that include long segmental duplications with high sequence identity. This can enable accurate variant calling in these regions [42, 62]. However, variant calling using error-prone SMS reads is challenging and short-read variant calling tools do not work well for SMS reads [42, 62]. To address the challenge of variant calling using SMS reads with high error rates, several new methods [42, 43, 62, 63] have been developed. Some of these methods use deep learning based models [43, 63] to overcome the high error rate while others exploit the long-range haplotype information present in SMS reads to enable haplotype-resolved variant calling [42, 62]. Recent work has shown that these variant calling methods achieve high precision and recall for single nucleotide variant (SNVs) calling in unique regions of the human genome that is comparable to that using Illumina WGS [42]. More recently, Circular consensus sequencing (CCS) can generate long reads with high accuracy (99.8%) using multiple passes of the PacBio SMS technology over a single template molecule [59]. The high acccuracy of these HiFi reads enables the accurate detection of both SNVs and short indels in human genomes [59] and also improves the mappability of the genome (97.8% of non-gapped bases) compared to short reads (94.8%).

Nevertheless, many segmental duplications are much longer than the average read length of HiFi reads and remain difficult to map unambiguously [59]. Using simulated PacBio reads, Edge and Bansal [42] found that long-read alignment tools such as Minimap2 [38] and NGMLR [64] result in low recall for variant calling in segmental duplications. Long-read alignment tools typically calculate alignment or similarity scores for each of the possible mapping locations for a read and assign it a high mapping quality if the alignment score of the best location exceeds that of the second best location using some threshold. Long repeated sequences in the human genome result in multiple locations with high scores and pose problems for long-read alignment tools. Recent work has shown that the accuracy of long-read mapping in extra-long tandem repeats in the human genome – typically found in centromeres – can be improved using specialized computational methods [65–67] that are designed to exploit the sequence and structure of long repeats. For example, the Winnowmap algorithm [66] modifies the sequence matching algorithm to avoid filtering out repeated k-mers that are common in tandem repeats [66].

In long segmental duplications with high sequence identity, there is potential to improve alignment accuracy by leveraging prior knowledge about the location and sequence of the duplications. Paralogous sequence variants (PSVs) – differences in sequence between a segmental duplication and its homologous sequences – are the primary source of information for assigning reads to their correct location in such regions. PSVs have previously been used to distinguish paralogous repeat copies and estimate paralog-specific copy number using short reads [68]. More recently, Vollger et al. [69] have developed a computational method for the de novo assembly of segmental duplications that uses PSVs to separate paralog copies. The high error rates of PacBio single-pass and ONT reads make the problem of distinguishing paralogous repeat copies even more difficult. In this chapter, we describe a new probabilistic method for accurate mapping of long reads in segmental duplications that explicitly leverages PSVs to distinguish between repeat copies and assign reads with high confidence. Our method, DuploMap, builds on existing long-read alignment tools and carefully analyzes reads that are mapped to known segmental duplications in the genome. It performs local realignment around PSVs to maximally utilize the information present in noisy SMS reads.

PSVs are defined using a reference genome and it has been shown that a subset of PSVs correspond to polymorphisms in the human population [70, 71]. Using such unreliable or uninformative PSVs for differentiating repeat copies can result in conflicting evidence in support of different alignment locations resulting in reduced sensitivity and specificity of read mapping. To identify and discard uninformative PSVs, DuploMap jointly performs read mapping and PSV genotyping using an iterative algorithm. Only reliable PSVs are used to assign reads to homologous repeat copies. We use simulated data to evaluate the improvement in mappability using DuploMap on alignments generated using existing long-read mapping tools. We also demonstrate the impact of DuploMap on read mappability and variant calling in segmental duplication in the human genome using a number of real datasets generated using the Pacific Biosciences and Oxford Nanopore technologies. DuploMap is open-source software available at https://gitlab.com/tprodanov/duplomap.

2.2 Materials and Methods

Given SMS reads aligned to a reference genome (using a long read aligner such as Minimap2), our objective is to analyze reads that overlap segmental duplications (and their homologous sequences), determine the most likely alignment location for each read and assign a mapping quality to it [72]. We assume that standard alignment tools can correctly align reads in the unique regions of the genome. Therefore, we do not examine reads that do not have a primary alignment overlapping segmental duplications. DuploMap uses prior knowledge about segmental duplications in the human genome to identify clusters of duplicated sequences and pairwise PSVs.

2.2.1 Clustering segmental duplications and identifying PSVs

To identify segmental duplications and PSVs, we used a previously computed database of segmental duplications for the human genome [3, 49]. The database was downloaded from the UCSC table browser [6]. First, we filtered out all pairs of homologous sequences for which the fraction of matching bases was less than 97% or the length of the alignment was less than 5000 bases. Next, we constructed a graph on the segmental duplications where each node was a genomic interval with homology to at least one other interval. This graph had two types of edges: (i) similarity edges between pairs of homologous sequences from the segmental duplication database and (ii) proximity edges between pairs of intervals that are less than 500 bases from each other. The proximity edges were added since reads that overlap intervals close to each other but not homologous have to be analyzed jointly since they affect the PSV genotypes of both components. For the hg38 reference human genome, the segmental duplication graph had 5,818 nodes and 26,301 edges (3,587 similarity and 22,714 proximity edges). We removed 88 of the 256 clusters that did not contain any duplications longer than 10 kilobases and with sequence similarity at least 98%. Most of the remaining clusters were small (less than 3 nodes) but the largest connected component contained 3,301 nodes and 17,752 edges.

To identify PSVs, we used Minimap2 (options `-ax asm20`) to align each pair of homologous sequences. For a pair of aligned sequences $S_1,\ S_2$, we first searched for anchors: k-mers shared between the homologous sequences and represented by k consecutive matches in the pairwise alignment. Each anchor sequence is required to be unique in a window around it (by default, $k = 6$ and window length = 20). This way we get a set of anchor starting positions $\{a_i^{(1)}\},\ \{a_i^{(2)}\}$. If the homologous sequences between two consecutive anchors are different: $S_1[a_i^{(1)} + k \ldots a_{i+1}^{(1)} - 1] \neq S_2[a_i^{(2)} + k \ldots a_{i+1}^{(2)} - 1]$, we define a pairwise PSV as a pair of intervals $(a_i^{(1)} + k, a_{i+1}^{(1)} - 1)$ and $(a_i^{(2)} + k, a_{i+1}^{(2)} - 1)$. As a result of this, adjacent PSVs can be merged into a single PSV. The substrings $S_1[a_i^{(1)} + k \ldots a_{i+1}^{(1)} - 1]$ and $S_2[a_i^{(2)} + k \ldots a_{i+1}^{(2)} - 1]$ define the two alleles for the PSV. To avoid excessive number of PSVs in regions with low sequence similarity, we did not consider regions within the pairwise alignments that had sequence similarity lower than 95% and were longer than 300 bp. Finally, low-complexity PSVs were filtered out and only high-complexity PSVs were retained for genotyping (see Section A.2.1 for details).

2.2.2 DuploMap algorithm

For each cluster in the segmental duplication graph, DuploMap identifies reads that overlap segmental duplications in the cluster and analyzes the reads to determine the alignment location and mapping quality of each read. Unlike existing mapping tools, which map each read independently to the reference genome, DuploMap uses information from all reads jointly to align reads overlapping segmental duplications. This is done by identifying uninformative PSVs jointly with estimating the read alignment locations and mapping qualities. For a cluster of duplications, DuploMap first retrieves all reads for which the primary alignment intersect the genomic intervals contained in the cluster. Next, it performs the following steps on the set of reads:

1. For each read:

    - Find the set of potential alignment locations,
    - Use LCS-based filtering to discard some alignment locations,
    - If the number of alignment locations after filtering is one, assign read to that location with high confidence (mapping quality = 254),
    - Determine the actual alignment for the read and each alignment location using Minimap2.

2. For each PSV, estimate genotype likelihoods using reads aligned with high confidence (mapping quality greater than a threshold), and identify reliable PSVs.

3. For each read with two or more potential alignment locations, calculate location likelihoods using reliable PSVs and estimate mapping quality for best alignment location.

4. Repeat steps 2 and 3 until the read assignments do not change.

After we assign mapping locations and qualities for all reads for a given cluster of segmental duplications, we perform additional post-processing to identify reads that shown high rate of discordance with the genotypes of overlapping PSVs. Reads with high discordance can be result of missing duplicated sequences in the reference genome or due to other reasons such as structural variants. Reads that overlap at least five PSVs and show a high rate of discordance are assigned a low mapping quality (see Section A.2.4 for details). Next, we describe the individual steps (1, 2 and 3) of the algorithm in detail. The procedure for identifying the set of potential alignment locations uses the segmental duplication database (step 1) is described in the Section A.2.3.

2.2.3 Filtering alignment locations using longest common subsequences

For reads overlapping segmental duplications with high sequence identity, comparing the alignment scores for different candidate locations is not very informative of the correct location, particularly for reads with high error rates. We developed a LCS-based strategy that uses k-mers that are unique to a particular alignment location to filter out unlikely locations. The motivation underlying this approach is that the correct alignment location should share unique k-mers with the read, i.e. k-mers that are not present in other locations and the number of such shared unique k-mers should be significantly greater than other locations. This approach allows us to quickly map reads that have some part located outside segmental duplications as well as reads that intersect divergent region(s) within segmental duplications.

We use the LCSk++ algorithm [73] to find the longest common subsequence $\text{LCS}_k(a, b)$ of k-mers shared between a pair of sequences a and b. Function $N(\cdot)$ counts the number of non-overlapping k-mers in a set. Suppose, a read r has n candidate locations $\{l_i\}_{i=1}^n$. For a pair of locations i and j we find three LCS sets: $\text{LCS}(r, l_i)$, $\text{LCS}(r, l_j)$ and $\text{LCS}(l_i, l_j)$. Let $A_{ij} = N\big(\text{LCS}(r, l_i) \setminus \text{LCS}(r, l_j)\big)$ be the k-mers that are present in the LCS between the read and the i-th location, but not in the LCS between the read and the j-th location. Additionally, let $B_{ij} = N\big(k\text{-mers}(l_i) \setminus \text{LCS}(l_i, l_j)\big)$ be the k-mers from the i-th location that are not in $\text{LCS}(l_i, l_j)$. We use the Fisher's Exact Test to calculate the p-value of the contingency table

$$\begin{bmatrix} B_{ij} - A_{ij} & A_{ij} \\ B_{ji} - A_{ji} & A_{ji} \end{bmatrix}.$$

Without loss of generality, suppose that the read is more similar to the location i than to location j. In this case, a low p-value of the test would confirm that the ratio of shared read-location k-mers (A_{ij}) to the number of unique k-mers for location i (B_{ij}) is significantly higher than the corresponding ratio for location j: A_{ji}/B_{ji}. We considered values for k from 9 to 15, which showed similar results on simulated data (data not shown), and used $k = 11$. Since a single base difference can result in potentially k unique k-mers, counts of non-overlapping k-mers in the LCS are used for computing the Fisher's exact test.

For a read with two or more alignment locations, we calculate the LCS-based p-value for each pair of locations. We say that location i dominates location j if $A_{ij}/B_{ij} > A_{ji}/B_{ji}$ and the Fisher Exact Test p-value is less than a threshold (default = 0.0001). Then we select the smallest non-empty subset of locations that dominate all other locations using a directed graph (see Section A.2.3).

2.2.4 Read assignment using PSVs

For reads that have more than one possible alignment location after the LCS-based filtering, we use a PSV-based approach to determine the most likely alignment location (Figure 2.1b). For a long read r and n candidate alignment locations, we consider each pair of alignment locations in turn. For a pair of locations i and j, we use all high-complexity PSVs shared between these locations. For a PSV v we calculate read-PSV alignment probabilities for the two alleles of v. To account for the uncertainty in base-to-base alignment of long error prone reads, we use a small window around the PSV and average over all alignments using a pair-HMM [42]. We denote alignment probabilities $s_v^{(i)} = P(r_v \mid S_v^{(i)})$ and $s_v^{(j)} = P(r_v \mid S_v^{(j)})$, where r_v is read subsequence in a window around the PSV, $S_v^{(i)}$ and $S_v^{(j)}$ are the reference genome subsequences in the same window around the PSV at locations i and j. We cap alignment probabilities $s_v^{(i)} \longleftarrow \max\{\hat{s}, s_v^{(i)}\}$ and $s_v^{(j)} \longleftarrow \max\{\hat{s}, s_v^{(j)}\}$ to reduce impact of a single PSV on read mapping, or a single read on a set of reliable PSVs ($\hat{s} = 10^{-3}$ by default). Using these probabilities we calculate a likelihood for the true location of the read being i (relative to j):

$$P_{ij}(r) = \prod_{v \in V} \left[P(v \text{ is reliable}) \cdot s_v^{(i)} + (1 - P(v \text{ is reliable})) \right].$$

Note that $P(v$ is reliable$)$ is essentially the posterior probability of the reference genotype at both locations defined by a pairwise PSV. When the PSV is unreliable or uninformative, we assume that the PSV should be used to differentiate between the two locations and hence use a constant term (1) in the above equation. Initially, $P(v$ is reliable$)$ is assigned using a constant

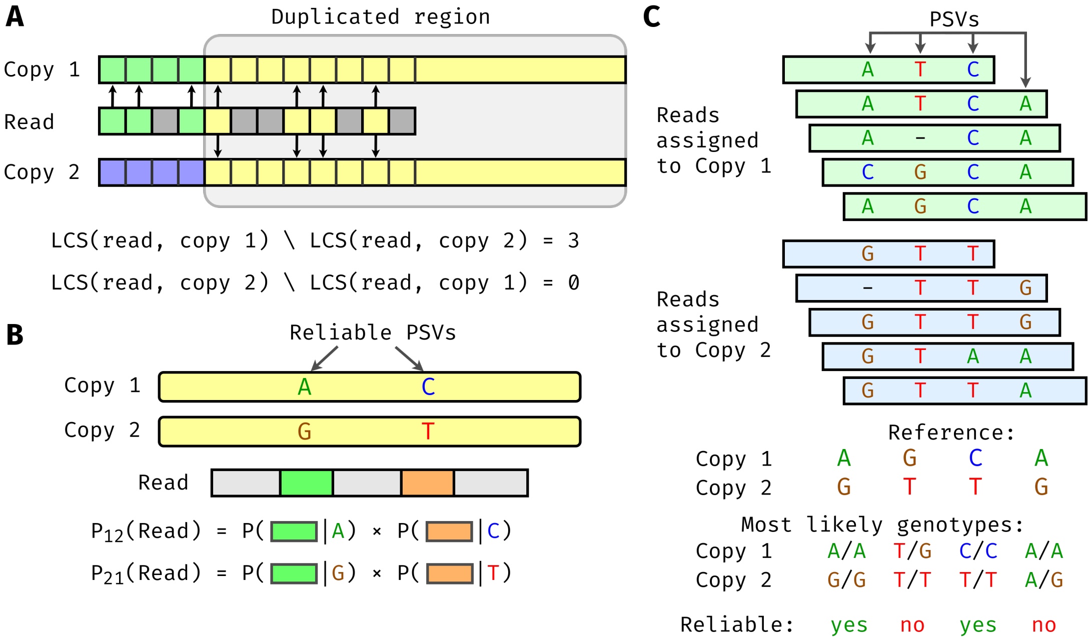


Figure 2.1. **Overview of the DuploMap method. (A) Filtering alignment locations using longest-common subsequences (LCS) of *k*-mers.** A read partially overlaps a segmental duplication and has two possible alignment locations (copy 1 and copy 2). The read and its possible locations are divided into *k*-mers that are shown with different colors. Arrows depict *k*-mers in the LCS between the read and the two copies. In the duplicated region, the read shares four *k*-mers with ‘copy 1’ that are also shared with ‘copy 2’. Outside the duplicated region, the read shares three *k*-mers (shown in green) with the *k*-mers of ‘copy 1’, but not with the *k*-mers of ‘copy 2’. **(B) Calculation of read-location probabilities using PSVs**. The read intersects two reliable PSVs that distinguish the two alignments locations. The probability of each location being correct (relative to the other location) are calculated using the local realignment probabilities between the read and the PSVs. **(C) Identifying reliable PSVs using assigned reads.** Five reads are mapped to ‘copy 1’ and five reads are mapped to ‘copy 2’ with high mapping quality. The genotype likelihoods for each PSV are calculated using these reads. Only two of the four PSVs have the reference genotype as the most likely genotype for both locations of each PSV and are considered reliable.

prior probability and in subsequent iterations it is estimated from the genotype likelihoods using reads assigned with high mapping quality to each location.

For reads with more than two candidate alignment locations, we use the pairwise likelihood to identify the “best” location b such that $P_{bi}(r) \geq P_{ib}(r)$ for all other alignment locations i. We select the second best location $s = \arg\min_i \frac{P_{bi}(r)}{P_{bi}(r)+P_{ib}(r)}$ and assign the mapping quality as $\min\left\{254, -10 \cdot \log_{10}\left(\frac{P_{bs}(r)}{P_{bs}(r)+P_{sb}(r)}\right)\right\}$. If no such location exists, we keep the original alignment of the read and assign it a mapping quality of 0.

2.2.5 Identifying reliable PSVs using assigned reads

PSVs are defined using the reference genome sequence, however, since segmental duplications are difficult to assemble, some PSVs may be assembly artifacts. It is also possible that the analyzed genome has different alleles on homologous chromosomes for some of the PSVs, i.e. the PSV sites overlap with variants. For a PSV v, defined between two locations i and j, we select all reads R_i and R_j that cover the PSV and are assigned to location i and j respectively with high confidence (mapping quality greater or equal to a threshold). We use these reads to calculate the joint likelihoods of the genotype pair (G_v^i, G_v^j) for the two locations. For each location, we consider three possible diploid genotypes defined by the two alleles of the PSV. For location i (j), the '0' allele corresponds to the reference sequence of the PSV at location i (j) and the '1' allele corresponds to the reference sequence of the PSV at location j (i). Hence the three possible genotypes for each location can be represented as $\{0/0,\ 0/1,\ 1/1\}$ and we can estimate the posterior probability of each genotype pair (g_i, g_j) as follows:

$$P(G_v^{(i)} = g_i, G_v^{(j)} = g_j \mid R_i, R_j) = \frac{\prod_{r \in R_i} P(r \mid g_i) \cdot \prod_{r \in R_j} P(r \mid g_j) \cdot p(g_i) p(g_j)}{\sum_{g_i',\, g_j'} \prod_{r \in R_i} P(r \mid g_i') \cdot \prod_{r \in R_j} P(r \mid g_j') \cdot p(g_i') p(g_j')}$$

where $p(g)$ is the prior probability of the genotype g. For most PSV sites, we expect the reference genotype $(0/0)$ to be the correct one. Therefore, we assign a high value for $p(0/0)$ and low probability for non- reference genotypes. For example $p(0/0) = 0.95$, $p(0/1) = p(0/0) \cdot (1 - p(0/0)) = 0.0475$ and $p(1/1) = 1 - p(0/0) - p(1/1) = 0.0025$. Experiments on simulated data using values for $p(0/0)$ ranging from 0.9 to 0.99 gave similar results (data not shown). Therefore, we use the prior value equal to 0.95 as default. $P(r \mid g)$ is a probability of the read subsequence conditional on the genotype g, which is calculated using alignment probabilities:

$$P(r \mid 0/0) = s_v^{(i)}, \quad P(r \mid 0/1) = \tfrac{1}{2} \cdot \left(s_v^{(i)} + s_v^{(j)}\right), \quad P(r \mid 1/1) = s_v^{(j)}.$$

We define the probability $P(v \text{ is reliable})$ as the posterior probability of the genotype being equal to the reference sequence at both locations, i.e. $P(G_v^{(i)} = 0/0, G_v^{(j)} = 0/0 \mid R_i, R_j)$.

2.2.6 Measuring alignment accuracy

To assess the accuracy of read mapping in segmental duplications using simulated data, we used two metrics: (i) recall: the fraction of correctly mapped reads out of all simulated reads with true location overlapping *Long-SegDups*, and (ii) precision: the fraction of correctly mapped reads out of all reads mapped to *Long-SegDups*. *Long-SegDups* refer to the subset of segmental duplications in the genome with length > 5 kb and with sequence similarity at least 97%. A read is considered to be mapped to *Long-SegDups* if its primary alignment overlaps *Long-SegDups* with mapping quality greater or equal than a certain threshold. We say that a read is mapped correctly if it is mapped to *Long-SegDups*, its primary alignment covers the true location by at least 25% (this allows partial alignments, nevertheless the vast majority of the reads overlap the true location by more than 95% or less than 5%, see Figure A.6). Additionally, the alignment should not go out of the true location by more than 100 bp in each direction to remove reads aligned to an incorrect copy in a tandem repeat. Precision and recall values for each mapping quality threshold were calculated by considering only reads with mapping quality greater than or equal to the threshold.

2.2.7 Simulations

We used SimLoRD [74] (`1.0.4`, options `-mp 1`) to generate PacBio SMS reads (median lengths of 8.5 kb, 20 kb and 50 kb) from the reference human genome (hs37d5) using the default error rates of 0.11 for insertion, 0.04 for deletion, and 0.01 for substitution [74]. Reads were forced to only have a single sequencing pass to resemble PacBio CLR reads as opposed to CCS or HiFi reads. We aligned the SMS reads to the human reference (hs37d5) using the long-read alignment tools BLASR (`5.3.3`, options `--hitPolicy allbest --nproc 8`), Minimap2 (`2.17-r941`, options `-t 8 -ax map-pb`) and NGMLR (`0.2.7`, options `-t 8 -x pacbio`). To assess variant calling accuracy, we simulated a diploid genome using the reference human genome sequence with heterozygous SNVs (rate = 0.001) and homozygous SNVs (rate = 0.0005) [42].

We used NanoSim [75] (`2.6.0`) to generate Oxford Nanopore (ONT) SMS reads (mean length of 8.4 kb) using a pre-trained model `human_NA12878_DNA_FAB49712_guppy`. We aligned the simulated ONT reads to the hs37d5 human reference genome using Minimap2 with options `-t 8 -ax map-ont`.

2.2.8 Whole-genome SMS datasets

We used whole-genome SMS datasets for five human individuals generated using different sequencing technologies (PacBio CCS, PacBio CLR and Oxford Nanopore) by the Genome in a Bottle (GIAB) consortium [76]. These datasets were downloaded from the GIAB ftp server: ftp://ftp-trace.ncbi.nlm.nih.gov/giab/ftp/data/ and were aligned to the hg38 reference genome using the tool Minimap2. In addition, Oxford Nanopore reads for NA12878 were obtained from the Nanopore WGS Consortium [19] and aligned to hg38 using minimap2. We also used 10X Genomics datasets (aligned reads and variant calls) for HG001 and HG002 obtained from the GIAB ftp server. We also downloaded a PacBio CLR dataset (SRX8173259) from the SRA and aligned the reads to the mouse reference genome (mm10). Detailed information about the individual datasets is provided in the Section A.2.8.

2.2.9 Variant calling

Variants were called on the HG002 whole-genome PacBio CCS data using the tool Longshot [42] (v0.4.1). Variant calling was done using four different thresholds for the mapping quality (0, 10, 20 and 30). For each threshold, only reads with mapping quality greater than or equal to the threshold were used (-q parameter). Only variants with `PASS` filter and quality value at least 30 were used for analysis. High confidence variant call sets generated by the GIAB consortium were used for assessing accuracy of variant calling [76, 77]. For the HG002 genome, SNVs were compared against the GRCh38 version of the GIAB high-confidence call set (release v3.3.2 and v.4.1). The comparison of variant calls was limited to high-confidence regions (provided in a bed file). Precision and Recall were calculated using RTGtools vcfeval [78, 79] (`v3.11`). Comparison of different sets of variant calls was also done using RTGtools vcfeval.

2.3 Results

2.3.1 Overview of method

DuploMap is a probabilistic method specifically designed to improve the sensitivity and specificity of long-read alignments in segmental duplications in the genome. It starts from an existing set of aligned reads (generated using a long read alignment tool such as Minimap2) and updates the alignments and mapping qualities of reads that are mapped to segmental duplications. It utilizes a pre-computed database of segmental duplications and PSVs for this purpose. In the first step, DuploMap identifies the candidate alignment locations for each read whose initial alignment overlaps segmental duplications and uses an efficient filtering approach based on calculating the Longest Common Subsequence (LCS) to identify the most likely alignment location. This LCS-based filtering approach can identify the correct alignment location for reads that overlap a non-repetitive sequence (Figure 2.1a) and for reads from segmental duplications with moderate sequence identity.

For reads that overlap segmental duplications with very high sequence identity, DuploMap aligns read sequences to possible alignment locations using Minimap2 and performs local realignment in the neighborhood of PSVs that overlap a read. The local realignments are used calculate read-location likelihoods and estimate the most likely location for each read (Figure 2.1b). Since some PSVs may not correspond to fixed differences between homologous sequences, DuploMap uses the reads assigned to each repeat copy (Figure 2.1c) to identify reliable PSVs – PSVs for which the genotype at the homologous positions is consistent with the reference genome. The read-location likelihoods are estimated using only reliable PSVs. Since reliable PSVs are not known in advance, the read assignments and the set of reliable PSVs are inferred using an iterative algorithm (see Methods).

Segmental duplications are defined as sequences with length at least 1 kb and sequence similarity ≥ 90% [49]. However, not all such segmental duplications are challenging for long read alignment. We used simulations to assess the mappability of SMS reads in duplicated sequences as a function of length and sequence similarity (data not shown). Based on these simulations, we constructed a subset of segmental duplications in the human genome with length greater than 5 kb and with sequence similarity at least 97%. These regions cover 86 and 101 megabases of the hg19 and hg38 reference human genomes, respectively. DuploMap only analyzes reads that overlap such segmental duplications (referred to as *Long-SegDups* in this chapter).

2.3.2 Evaluation of mapping accuracy using simulated reads

We simulated single-pass PacBio SMS reads using the SimLORD tool [74] with mean length equal to 8.5 kb and aligned them to the reference human genome using the long read alignment tool, Minimap2 [38]. Alignment tools report a mapping quality for each read which represents the probability that the reported alignment for a read is correct [72]: a mapping quality of 10 (20) corresponds to a probability of 0.9 (0.99). Analysis of the aligned reads showed that 74.9%, 69.0% and 63.1% of the reads that overlap *Long-SegDups* had mapping quality ≥ 10, ≥ 20 and ≥ 30 respectively. Furthermore, for reads completely within segmental duplications with ≥ 99.5% similarity, only 40.7% of reads had mapping quality greater or equal to 10. Although reads that overlap regions that are completely identical between the duplicated sequences cannot be mapped unambiguously, a significant fraction of the reads with low mapping quality overlapped multiple PSVs (illustrated for the *STRC* gene in Figure A.1). Specifically, 70.6% of the reads that had a mapping quality less than 10 overlapped five or more PSVs. Next, we mapped the simulated reads using BLASR [80], a long read alignment tool developed specifically for PacBio reads. BLASR aligned a greater fraction of reads (80.8%) with mapping quality ≥ 20 compared to Minimap2 (68.4%) but was 28 times slower (Table A.1). This increased mappability came at the cost of accuracy: 3.2% of reads with mapping quality ≥ 20 were mapped to the incorrect location. The accuracy of another long read alignment tool, NGMLR [64], was significantly worse compared to Minimap2 and BLASR at all mapping quality thresholds (Figure A.3).

Next, we used DuploMap to post-process the alignments generated using each of the long-read alignment tools separately. For a given mapping quality threshold, we used precision (fraction of correctly aligned reads out of reads *mapped* to *Long-SegDups*) and recall (fraction of correctly aligned reads out of *all simulated reads* in *Long-SegDups*) to assess the accuracy of read mapping. DuploMap improved both the precision and recall of read mapping in segmental duplications for all long-read mapping tools (Figure 2.2). For Minimap2, DuploMap improved the recall from 0.743 to 0.906, at a mapping quality threshold of 10, while maintaining high precision (0.9954, Figure 2.2 and Figure A.2a). The improvement in recall was greater for higher mapping quality thresholds. Even if we consider all aligned reads (mapping quality threshold of 0), re-alignment using DuploMap increased both the precision and recall by 1.2 percentage points. DuploMap also improved both precision and recall for BLASR (Figure 2.2) and NGMLR (Figure A.3). In particular, the precision increased considerably from 0.965 (0.967) to 0.994

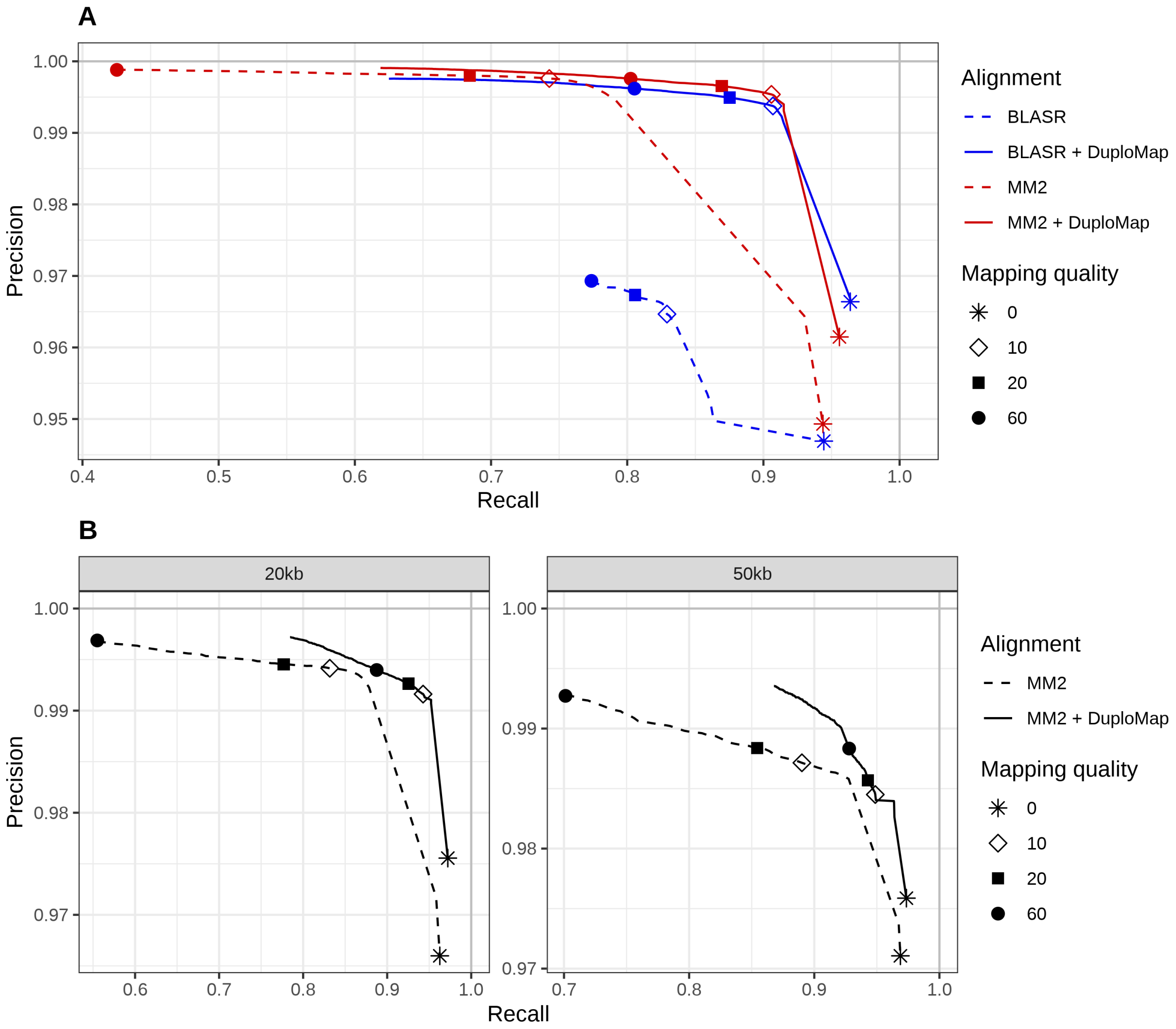


Figure 2.2. **Accuracy of read mapping in segmental duplications using simulated long read data.** Each curve shows the precision and recall of different alignment methods as a function of mapping quality thresholds. Dashed lines correspond to the original alignments while the solid lines show the alignments resulting from realignment using Duplomap. **(A)** Comparison of Minimap2 (MM2), BLASR, Minimap2+DuploMap, and BLASR+DuploMap on simulated reads with mean length 8.5 kb. **(B)** Accuracy of Minimap2 (MM2) and Minimap2+DuploMap on simulated reads with mean lengths 20 kb and 50 kb.

(0.995) while recall improved from 0.829 (0.806) to 0.907 (0.875) at a mapping quality threshold of 10 (20) for BLASR.

Minimap2 uses a minimizer based approach for finding matches between reads and the reference genome [38]. To improve speed, minimizers with a high frequency (top 0.02%) are discarded by default. We evaluated whether discarding a lower fraction (different values of the parameter f) could improve accuracy of read mapping in segmental duplications. Using $f = 0$ (use all minimizers) improved the recall slightly (0.743 to 0.763) but increased the memory usage five-fold (Figure A.2b). Nevertheless, post-processing using DuploMap achieved a higher recall (0.906) while maintaining a high precision (0.9954). We also evaluated Winnowmap [66], a long-read alignment tool that uses a weighted sampling based method for selecting minimizers to improve long-read mapping using Minimap2 in long tandem repeats. However, Winnowmap's recall and precision in *Long-SegDups* regions were lower than those for Minimap2 (Figure A.3).

Next, we evaluated the accuracy of mapping SMS reads in segmental duplications as a function of read length. For this, we simulated PacBio single-pass reads of mean length 20 and 50 kilobases and aligned them to the reference genome using Minimap2. Not surprisingly, the recall for reads (at a fixed mapping quality threshold) increased as the read length increased (Figure 2.2b). Nevertheless, even for 50 kb long reads, recall was only 0.890 at a mapping quality threshold of 10. Re-alignment using DuploMap increased the recall to 0.949 for 50 kb reads, while keeping the precision high (0.985).

We also examined the impact of error rate and sequencing technology on read mapping in segmental duplications. We simulated PacBio single-pass reads with mean length 8.2 kb and high error rates (15%, 9% and 4% for insertion, deletion and substitution, respectively). At a mapping quality threshold of 10 (20), Minimap2 mappings in *Long-SegDups* had a low recall of 0.691 (0.645) with high precision of 0.997 (0.998). DuploMap improved the recall to 0.891 (0.853) while maintaining a high precision of 0.995 (0.997). We used the NanoSim tool [75] to simulate Oxford Nanopore reads and aligned them to the reference genome using Minimap2 (see Materials and Methods). At a mapping quality threshold of 10 (20), post-processing using DuploMap improved the recall from 0.755 (0.661) to 0.882 (0.837) and kept the precision high at 0.994 (0.996).

PSVs or paralogous sequence differences are defined using the reference genome sequence, however, some PSVs overlap with polymorphisms and should not be used to differentiate between the paralogous sequences. To assess whether DuploMap can map reads accurately in the presence of uninformative PSVs, we simulated PacBio reads from two-copy segmental duplications with 0%, 15% and 30% of the PSV genotypes assigned to be non-reference on one of the copies. DuploMap successfully identified uninformative PSVs and achieved high recall and precision for read mapping even with a high fraction of uninformative PSVs (Table A.3).

2.3.3 Improvement in read mapping for diverse SMS technologies

To assess the impact of DuploMap on read mapping in segmental duplications using real SMS data, we analyzed PacBio CCS whole-genome data for an individual, HG002 (NA24385), from the GIAB project [76]. The HiFi reads were initially aligned using Minimap2 to the hg38 reference genome. Post-processing the reads using DuploMap increased the percentage of reads with mapping quality ≥ 10 in segmental duplications from 65.7% to 80.6%, an increase of 15 percentage points (Figure 2.3). DuploMap can change both the alignment location and the mapping quality of reads overlapping segmental duplications. Comparison of the original Minimap2 and the DuploMap alignments showed that 4.8% of reads that had initially very low mapping quality (< 5) were aligned to a different location with mapping quality ≥ 30 (Figure 2.3). Similarly, DuploMap reduced the mapping quality of 1.9% of the reads – that initially had mapping quality ≥ 30 – to less than 10. We observed similar improvements in mappability for several human PacBio HiFi and CLR datasets (Table 2.1 and Figure A.4). The increase in the percentage of reads aligned with a mapping quality greater than a threshold was consistently greater for HiFi reads compared to CLR reads. This was due to the improved ability to correctly allelotype PSVs using the HiFi reads: 1.7% local read-PSV alignments were ambiguous for CCS reads compared to 15.6% for CLR reads from the HG002 genome.

Wenger et al. [59] demonstrated that relative to Illumina reads, PacBio HiFi reads increased the fraction of the genome that is mappable, i.e. covered by at least a certain number of reads with high mapping quality. Nevertheless, several disease-relevant genes such as *SMN1*

Table 2.1. **Improvement in mappability of reads using DuploMap on multiple SMS whole-genome sequence datasets.** The last four columns show the percentage of reads with high mapping quality (≥ 10 and ≥ 20) that overlap *Long-SegDups* regions in the *Minimap2* alignments and the difference between *Minimap2 + DuploMap* alignments and *Minimap2* alignments. CLR = PacBio Contiguous Long Reads, CCS = PacBio Circular Consensus Sequencing, ONT = Oxford Nanopore Technology, MM2 = Minimap2.

Genome	Sequencing	Median	Reads	Read	MM2 (%)		Δ MM2+Duplomap (%)	
	technology	coverage	analyzed	length (N50)	MQ ≥ 10	MQ ≥ 20	MQ ≥ 10	MQ ≥ 20
HG002	CLR	45	878k	11,318	59.4	52.9	+8.4	+10.7
HG003	CLR	20	416k	10,999	59.9	53.5	+9.8	+11.3
HG004	CLR	19	362k	10,946	65.1	58.3	+8.7	+10.5
HG002	CCS	29	300k	13,480	65.7	58.9	+14.9	+19.5
HG005	CCS	32	454k	10,436	64.2	56.6	+15.8	+20.7
HG001	CCS	29	381k	10,004	71.6	63.7	+15.0	+21.2
HG001	ONT	36	535k	13,788	63.5	55.7	+3.9	+7.8
HG002	ONT	58	464k	54,352	64.5	58.0	-1.5	+1.7

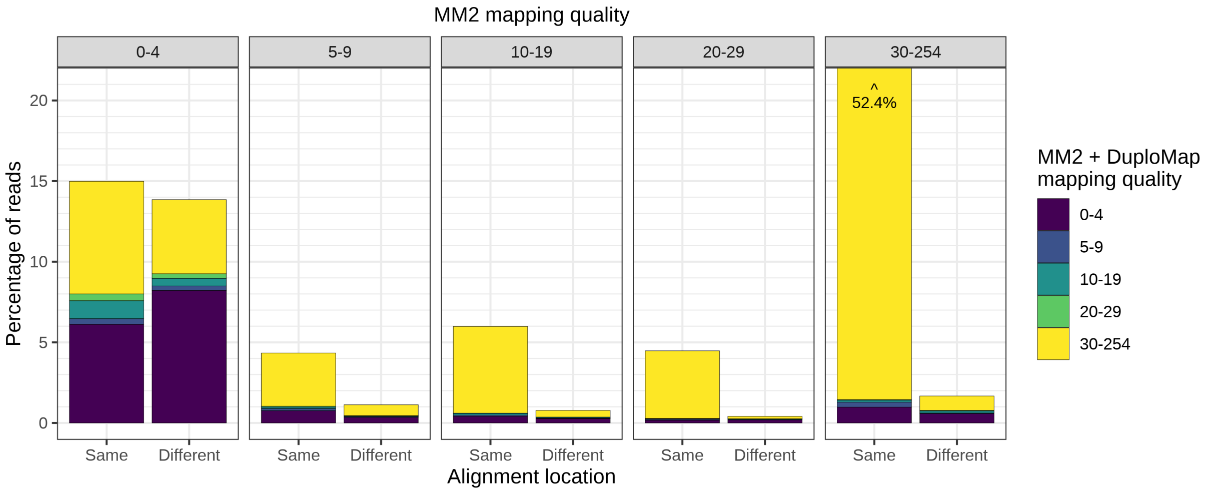


Figure 2.3. **Comparison of mapping qualities and alignment locations for reads aligned with Minimap2 (MM2) and Minimap2+DuploMap on the HG002 CCS dataset.** Five bar-lots corresponding to five bins of mapping quality using Minimap2 are shown. Each bar-plot shows the percentage of reads – color-coded by mapping quality after realignment using Duplomap — that had the same or different alignment location using Minimap2 and Minimap2+Duplomap. One of the bars (30-254 bin) that corresponds to 52.4% of reads is clipped for visual clarity.

were still only partially mappable using HiFi reads. We assessed the impact of realignment using DuploMap on the mappable fraction of the human genome. To enable comparison between

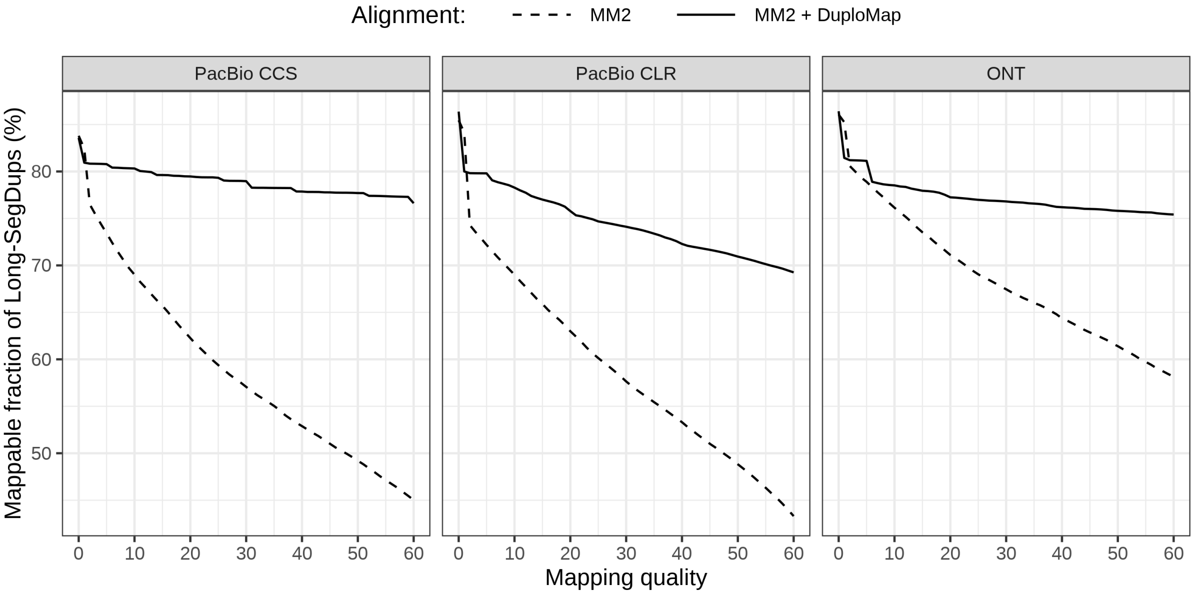


Figure 2.4. **Improvement in mappability of *Long-SegDups* regions using DuploMap and three long-read datasets for the HG002 genome.** Each sub-plot shows the percentage of the *Long-SegDups* regions (78.7 Mb on chromosomes 1-22) that is mappable at different mapping quality thresholds using *Minimap2* and *Minimap2 + Duplomap* alignments. A position is considered mappable if the number of reads covering it is at least 50% of the median coverage for the dataset.

datasets with different sequencing coverages, we defined a genomic position as mappable if the number of reads covering it – with mapping quality greater than a threshold – is at least 50% of the median coverage for the dataset. Relative to Minimap2, realignment using DuploMap increased the fraction of the genome – limited to segmental duplications – that is mappable at all mapping quality thresholds (Figure 2.4). For HiFi reads, at a mapping quality threshold of 10 (20), 80.32% (79.47%) of the *Long-SegDups* regions were mappable relative to 69.01% (62.26%) using Minimap2. This also increased the mappability of 11 (16) of the 193 disease-associated duplicated genes using HiFi reads (Table A.5).

Next, we analyzed a whole-genome human dataset generated using the Oxford Nanopore technology [19, 77]. Similar to PacBio datasets, realignment using DuploMap increased the fraction of reads with mapping quality greater than 10 (20) by 3.9 (7.8) percentage points (Table 2.1). For the ONT dataset with ultra-long reads (mean read length of 54.4 kb), only a minor improvement in the number of reads with high mapping quality was observed (Table 2.1). Nevertheless, at a mapping quality threshold of 10, an additional 1.9 Mb of DNA sequence is mappable using DuploMap aligned reads compared to reads aligned using Minimap2 (Figure 2.4).

DuploMap can post-process long reads for any genome with a reference sequence and a database of segmental duplications. We downloaded a mouse PacBio CLR dataset (median coverage 25) and aligned it to the mm10 reference genome using Minimap2. For running Duplomap, we created a PSV database for the mouse genome using a previously computed database of segmental duplications [81]. Of the 147k reads aligned to long segmental duplications with high sequence identity (mouse *Long-SegDups* regions, total length = 154 Mb), 77.1% (70.3%) were aligned with a mapping quality of 10 (20) or greater. Read re-mapping using DuploMap increased the percentage of reads with mapping quality of 10 (20) or greater to 89.5% (89.1%).

DuploMap is multi-threaded and can use multiple cores to process clusters of segmental duplications in parallel. It required 2-5 hours (using 8 CPU cores) to process simulated and real whole-genome PacBio datasets with 30× coverage (Tables A.1 and A.2). This additional run-time was only 25-30% of the run-time of Minimap2 for generating the initial set of alignments. Since DuploMap infers reliable PSVs jointly using reads mapped to a cluster of segmental duplications, it needs to store all read alignments for a cluster in memory and hence the memory usage increases with increasing coverage (see Table A.2).

2.3.4 Variant calling in segmental duplications using DuploMap alignments

DuploMap increases the fraction of the genome - limited to segmental duplications - that is mappable using SMS reads. This is expected to improve the sensitivity of variant calling in such regions. To assess this, we used the variant calling tool Longshot [42] on simulated PacBio reads (30× coverage). SNVs were called using Longshot for four different mapping quality thresholds (0, 10, 20 and 30), i.e. reads with mapping quality below the threshold were not used for variant calling. We found that the recall for variants called in *Long-SegDups* using reads re-aligned with DuploMap was greater than that obtained using Minimap2-aligned reads at all mapping quality thresholds (Figure A.5). At a mapping quality threshold of 10, the recall increased from 0.833 to 0.945 while the precision was virtually unchanged ($\geq$ 0.999 for both sets of alignments). The recall for Minimap2 was highest (0.898) when using all aligned reads (ignoring mapping quality) but resulted in significantly lower precision 0.904. For both Minimap2 and DuploMap, the best precision-recall tradeoff was observed at a mapping quality threshold of 10. For segmental duplications with 99.9% or greater identity, variants called using Minimap2+DuploMap alignments (mapping quality threshold of 10) had a recall 0.716 and precision equal to 0.998, compared to 0.253 and 0.994 respectively for variants obtained using Minimap2 alignments.

Next, we assessed the impact of the improved read mapping on variant calling using whole-genome PacBio HiFi data for HG002 (29× coverage). A recently developed variant calling tool, DeepVariant, has been shown to achieve very high precision and recall for PacBio HiFi reads [59]. Since a subset of the HG002 HiFi dataset was used for training the DeepVariant model [59], we used Longshot [42] for variant calling. Longshot has been shown to achieve high accuracy (F_1 score of 0.9985) for SNV calling on CCS reads [82]. Across chromosomes 1-22, Longshot called 3,727,419 SNVs using the reads realigned with DuploMap (mapping quality threshold of 10), 18,291 more than using the Minimap2 aligned reads. We used the high-confidence benchmark variant calls from the GIAB consortium (v3.3.2) that cover approximately 2.35 Gb of the GRCh38 version of the reference genome (excluding the X and Y chromosomes) to assess the accuracy of the SNV calls. The precision and recall of SNV calling using the Minimap2-aligned reads and the DuploMap-aligned reads was identical: 0.9963 and 0.990 respectively (see Table A.4). This was not surprising since the GIAB high-confidence benchmark variant calls (v3.3) were primarily generated using short read datasets and exclude the vast majority of repetitive regions in the genome.

The GIAB consortium recently released high-confidence benchmark variant calls (v4.1) for the HG002 genome that cover an additional 6% of the genome compared to the v3.3.2 calls. These benchmark variant calls incorporate information from 10X Genomics linked-read and PacBio HiFi read datasets and include variant calls in some segmental duplications. The precision and recall in the v4.1 regions for DuploMap and Minimap2 alignments were similar, although, SNV calls from DuploMap aligned reads had a higher F_1 score (0.9905) compared to Minimap2 (Table A.4). In the subset of the v4.1 regions that overlap *Long-SegDups* regions, DuploMap based calls had a higher recall compared to Minimap2 but lower precision at all mapping quality thresholds (Figure 2.5). Manual inspection of some of the false positives called using DuploMap aligned reads suggested that these may correspond to missing true positives in the GIAB v4.1 callset (see Figure A.9). Hence, the true precision may be higher. Nevertheless, the F_1 score of the DuploMap-based calls was consistently higher that the F_1 score of the calls using Minimap2 alignments. In addition, the improvement in the F_1 score was not dependent on the variant quality threshold used for Longshot (Table A.4). Visual inspection of the SNVs calls that were called only using the DuploMap alignments and matched the v4.1 benchmark calls showed that the vast majority of these SNVs were not called using Minimap2 alignments due to low mapping quality of the reads (see Figure A.7 for an example). We also identified a number of false positive variants called using the Minimap2 alignments that were corrected by variant calls using DuploMap alignments (see Figure A.8 for an example).

Next, we directly compared the SNVs calls made on the HG002 CCS dataset using Minimap2 aligned reads and reads re-aligned using DuploMap. We utilized 10X Genomics linked-read variant calls for the same individual as an independent source for comparison. In *Long-SegDups* regions on chromosomes 1-22, 83,648 DuploMap-derived SNVs were shared with 10X calls compared to 72,830 for Minimap2. 14,713 SNVs called exclusively using the DuploMap alignments were supported by 10X calls. The vast majority of these SNVs were located outside GIAB 4.1 high confidence regions and had low mappability using Minimap2 alignments (see Figure A.10 for an example of such a region that overlaps the medically-relevant

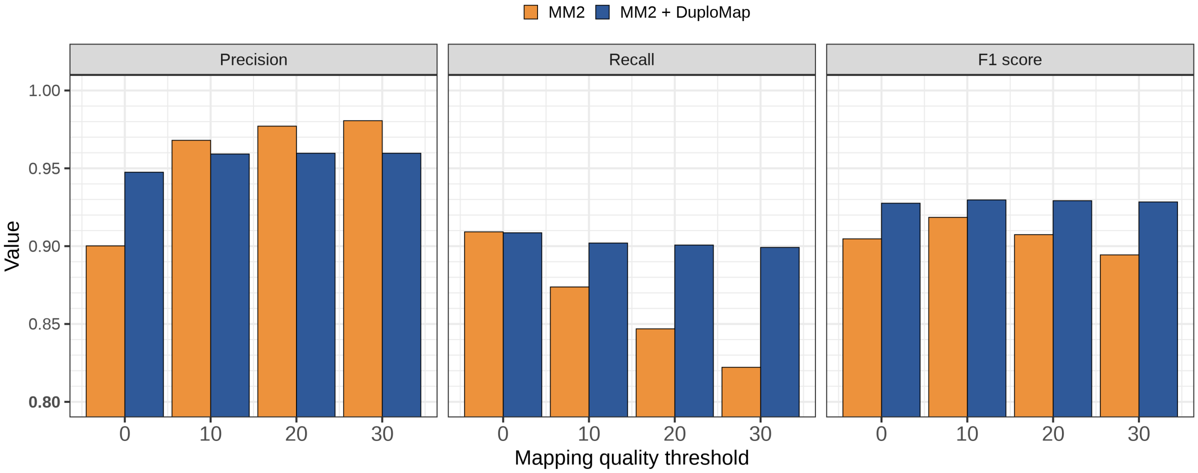


Figure 2.5. **Comparison of variant calling accuracy for HG002 CCS reads using Minimap2 and Minimap2+DuploMap.** SNVs were called using Longshot for different mapping quality thresholds. Precision, recall and F_1 values were calculated by comparison to the GIAB v4.1 benchmark calls in the *Long-SegDups* regions that overlapped with the GIAB high-confidence regions.

gene *GTF2I* [83]). We also identified 211 calls in GIAB high-confidence regions that were shared between the DuploMap calls and 10X calls but were absent in the GIAB v4.1 benchmark calls. Visual inspection of these calls suggested that for many of them, the GIAB benchmark callset is either missing a variant or has the incorrect genotype (see Figure A.9 for an example). Further, 36,021 SNVs were located outside the GIAB v4.1 high-confidence regions and were shared between all three callsets (10X, Minimap2 and DuploMap). These variants are likely to be true positives that are located outside GIAB v4.1 high-confidence regions.

2.3.5 Uninformative PSVs and variant calling using short reads

In addition to aligning reads that overlap segmental duplications, DuploMap also estimates genotypes for PSVs to identify unreliable or uniformative PSVs. Uninformative PSVs are likely to be the result of true variants in segmental duplications and therefore, should also be called as variants using long-read variant calling. Analysis of Longshot variant calls for the HG002 CCS dataset showed that 42.5% of the SNVs called using the DuploMap alignments in

Long-SegDups regions intersected PSV sites such that the variant allele matched the PSV allele at the homologous site. Such non-reference PSVs were not specific to DuploMap alignments; 43.4% of the SNVs called using Minimap2 alignments also intersected PSVs. In both cases, approximately 76% of the variants overlapping PSVs are present in the dbSNP database (build 151) [84].

Next, to assess the impact of uninformative PSVs on short read variant calling, we analyzed PacBio CCS read data for the HG001 (NA12878) genome for which pedigree-derived variant calls have been generated by the Platinum Genomes (PG) Project using whole-genome Illumina sequence data [85]. We focused our analysis of PSVs on two-copy segmental duplications. Of the 14,800 PSVs in two-copy duplications - with high confidence genotypes (QUAL $\geq$ 60) estimated by DuploMap - 16.5% had a genotype of (0/0, 0/1) and 6.0% had a genotype of (0/0, 1/1). A genotype of (0/0, 1/1) for a PSV implies that the genomic sequence at both homologous positions (on both alleles) is identical and hence the PSV cannot differentiate between reads from the two homologous sequences. Such PSVs are expected to cause incorrect read mapping and lead to incorrect variant calls since short read mapping tools rely on PSVs to place reads with high confidence in segmental duplications. For example, if a true variant is present in the region flanking the PSV position with a non-reference genotype, short reads covering the variant and the PSV can be mismapped to the homologous location resulting in a false variant call (see Figure A.11 for an illustration).

To search for false variants resulting from uninformative PSVs, we identified variants in the Platinum Genomes variant calls [85] for HG001 that were located near PSVs. Of the 2,769 variants that were located near uninformative PSVs in the PG calls, we identified 76 variants such that the variant was missing in the CCS variant calls but another variant was present at the homologous position with the same alternate allele. One such example of a false variant due to a uninformative PSV was located at the *PMS2* locus (Figure 2.6). The short-read PG calls report a SNV at chr7:6,752,118 (hg38 reference genome, rs1060836) that was also reported in gnomAD database of human variants [86] with an average allele frequency of 0.16 but with

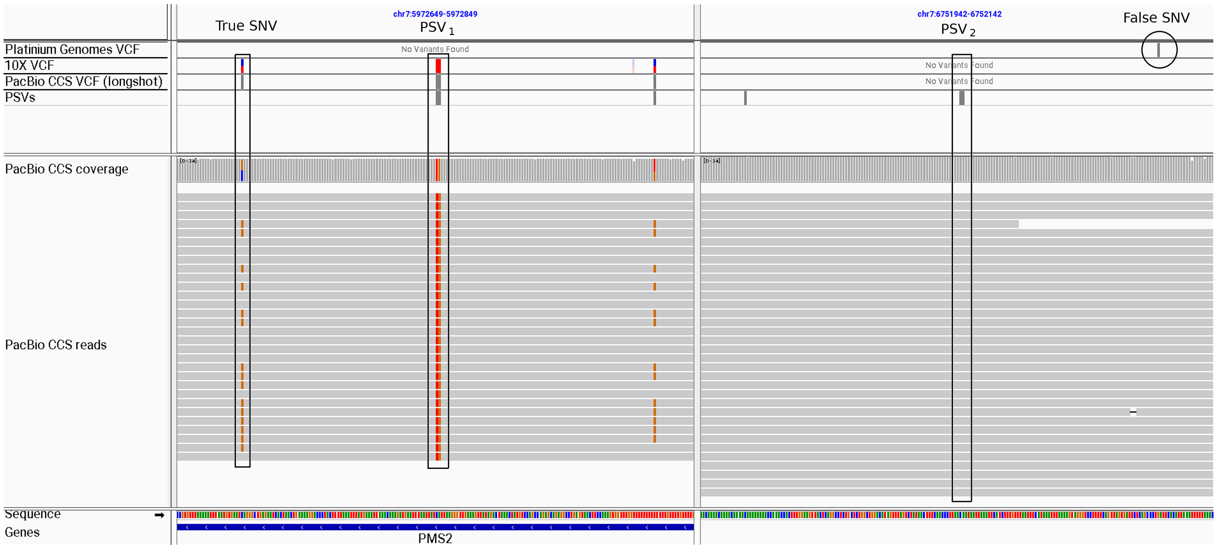


Figure 2.6. **Illustration of how unreliable PSVs adversely impact short-read variant calling in segmental duplications.** An Integrated Genomics Viewer (IGV) view of a duplicated region on chromosome 7 that overlaps the *PMS2* gene is shown. A PSV is located at chr7:5972749 (allele = with the homologous position at chr7:6752042. Variant calling on PacBio CCS reads (aligned with DuploMap) identifies two variants, a homozygous variant at the PSV site (chr7:5972749:CA:TG) and a heterozygous SNV located nearby (chr7:5972674:C:G). Both of these variants are supported by 10X Genomics variant calls but are absent from short-read variant calls for the same individual (Platinum Genomes VCF track). In addition, short-read variant calling results in a false SNV (chr7:6752118:G:C) at the position homologous to chr7:5972674 - a result of short read mismapping due to the unreliable PSV.

7-fold lower homozygotes than expected – indicative of a false variant. This SNV was absent from long read variant calls but a SNV located at chr7:5,972,674 – the position homologous to chr7:6,752,118 – was present in the DuploMap based calls and also in the 10X Genomics variant calls. This SNV was located less than 75 bases from an uninformative PSV chr7:5,972,749 that was actually called as a variant with the variant allele being the same as the allele at the homologous site (Figure 2.6).

2.4 Discussion

In this chapter, we presented DuploMap, a method designed specifically for re-aligning SMS reads that are mapped to segmental duplications by existing long-read alignment tools in order to improve accuracy. A unique feature of DuploMap is that it jointly analyzes reads overlapping segmental duplications and explicitly leverages paralogous sequence variants or PSVs for mapping. Using whole-genome human data generated using multiple SMS technologies, we demonstrate that DuploMap significantly improves the mappability of reads overlapping long segmental duplications in the human genome. DuploMap is not a stand-alone long read alignment tool but complements existing tools such as Minimap2 that tend to be conservative in aligning reads in segmental duplications.

The development of DuploMap is motivated by the goal of using long read sequencing technologies for variant calling in long segmental duplications that are problematic for short-read sequencing. The Genome in a Bottle Consortium has developed high-confidence small variant call sets for reference human genomes [76, 77, 87]. Their first call sets were based on short read sequencing and hence exclude almost all segmental duplications. The GIAB consortium is expanding the small variant calls to repeats including segmental duplications using the PacBio CCS and 10X Genomics linked read data-types. Accurate and sensitive read mapping of long reads is a pre-requisite for accurate and sensitive variant calling in long repeats in the human genome. Variant calling using the DuploMap aligned reads identified 14,713 variants in segmental duplications that were shared with 10X Genomics variant calls but were not called using Minimap2 aligned reads. This indicates that DuploMap can prove useful for variant calling in segmental duplications using PacBio CCS reads.

DuploMap is a robust method that works for multiple long read sequencing technologies (PacBio and ONT), can handle reads with high and low error rates, and can post-process reads aligned with different long-read alignment tools. DuploMap's approach of jointly modeling PSV genotypes and read alignments can potentially be used to improve the mapping of linked-reads in segmental duplications [88–90]. Although we have focused on variant calling, the ability to map long reads to segmental duplications with high sensitivity can benefit other uses of long read sequencing. Oxford Nanopore sequencing enables the detection of DNA methylation directly from the raw base signal [91, 92]. Miga et al. [67] have used a unique k-mer based mapping strategy to improve read mapping to generate base-level DNA methylation maps for the centromere of the X chromosome. DuploMap based alignment can enable the analysis of the methylation levels of duplicated genes that cannot be measured using short-read based methylation assays.

Analysis of PacBio CCS reads for a human genome showed that a significant number of PSVs overlap with variants and hence are uninformative for read mapping in segmental duplications. PSVs are defined based on the reference human genome sequence and a common variant in the human population can be incorrectly considered as a PSV if the variant allele is represented in the reference. In addition, gene conversion is well known to result in overlap between PSVs and variants [93, 94]. We also demonstrated that uninformative PSVs can cause incorrect mapping of short reads to homologous sequences resulting in both false positive and false negative variant calls. This problem can be alleviated by using information about reliable PSVs derived from analysis of long read datasets to inform short read mapping and variant calling in segmental duplications.

DuploMap has several limitations. First, the memory usage for DuploMap scales linearly with increasing number of reads since it stores information about all reads that overlap a single cluster of duplications. This can be reduced by writing some of the mapping information to disk or limiting the re-alignment to segmental duplications with low copy number. Second, DuploMap is not a stand-alone aligner and starts from alignments provided by existing long-read alignment tools. If a read is not aligned or aligned to a location that is not homologous to its correct location, DuploMap cannot find the correct alignment. Third, DuploMap does not currently account for missing sequences or copy number changes. Segmental duplications are well known to be hotspots of copy number variation and large structural variants in the human genome [68, 95]. Copy number information about duplicated sequences can be estimated using tools such as QuicK-mer2 [46] and used to potentially improve long read mapping in segmental duplications.

Finally, DuploMap is a reference-based method that relies on segmental duplications identified from a reference genome. Segmental duplications are problematic not only for read mapping but also for de novo assembly using long reads. The problem of distinguishing reads originating from different paralogs without a reference genome is even more challenging but can allow for assembling segmental duplications that may be collapsed or incorrectly represented in the reference genome. Several novel methods have been designed to specifically assemble segmental duplications that leverage long reads, particularly accurate HiFi reads [69, 96]. The SDip method has been shown to assemble diploid contigs for many duplicated genes such as *SMN1* [96]. As these methods develop further and more complete benchmarks for reference human genomes become available, it would be useful to compare the performance of reference-based and haplotype-aware assembly based methods for segmental duplications.

2.5 Data availability

All datasets analyzed in this chapter have been generated previously and are publicly available (links provided in the Section A.2.8). DuploMap is implemented in the Rust programming language and is freely available for download at https://gitlab.com/tprodanov/duplomap. DuploMap can be used to map reads in individual clusters of segmental duplications or across the entire genome. It is also available via conda (`conda install -c bioconda duplomap`). The repository also contains links to pre-computed PSV databases and BED files with *LongSegDups* for the hg19 and hg38 versions of the human genome.

2.6 Acknowledgements

We thank Justin Zook and Justin Wagner for useful discussions and help with accessing GIAB datasets.

Chapter 2, in full, is a reformatted reprint of “Sensitive alignment using paralogous sequence variants improves long-read mapping and variant calling in segmental duplications”. Timofey Prodanov & Vikas Bansal. *Nucleic Acids Research* **48**, e114 (2020), https://doi.org/10.1093/nar/gkaa829. The dissertation author was the primary author of this paper.

CHAPTER 3

Paralog-Specific Copy Number for Duplicated Genes

3.1 Introduction

Whole-genome sequencing (WGS) has the potential to profile all genetic variants simultaneously in a genome, however, the presence of repetitive sequences in the human genome hinders the ability to achieve this potential. Segmental duplications or low-copy repeats (LCRs) are long segments of repetitive DNA that constitute 5-8% of the human genome [3, 5]. More than 900 genes are known to overlap these segmental duplications and mutations in several such genes are associated with rare and complex human diseases [50]. Genes that overlap segmental duplication or have high sequence homology to other loci in the genome are problematic for short-read sequencing technologies since the reads derived from such genes have ambiguity in their alignment and are difficult to correctly position in the genome [48, 50, 52]. As a result, variants such as SNVs and short indels are difficult to identify in these genes using short reads [54].

Low copy repeats are also highly susceptible to copy number changes including deletions and duplications as well as reciprocal crossover (gene conversion) events that can change paralog-specific copy number. Many of these copy number changes are known to be disease associated [97–101]. For example, copy number of the *SMN1/2* gene can modify phenotype for spinal muscular atrophy (SMA) and copy number changes at the *STRC* locus are known to cause hearing loss [99]. In spite of their relevance for human disease, most duplicated genes are excluded from standard WGS analysis pipelines since the presence of paralogous sequences with high sequence identity and extensive copy number variation makes it difficult to analyze these loci accurately.

To enable the detection of clinically relevant copy number variants in disease-associated duplicated genes, specialized diagnostic assays have been developed that utilize Quantitative real-time PCR (qPCR), paralog ratio tests [102, 103] (PRT) and multiplex ligation-dependent probe amplification [104] (MLPA). Both qPCR and PRT utilize PCR product specificity to distinguish paralogous copies of a gene. However, these methods are labor-intensive and require the design and testing of multiple primers for each locus. Therefore, these methods cannot scale easily for copy number analysis of the hundreds of duplicated genes in the human genome. Array-based methods such as CGH can scale for multiple genes but cannot provide paralog-specific copy number which can be important for disease mapping. For example, at the *SMN1* locus (the two genes *SMN1* and *SMN2* only differ by 5 nucleotides), individuals with two copies of *SMN1* and one copy of *SMN2* are healthy while individuals with one copy of *SMN1* can be affected [105].

Analysis of read depth using WGS data mapped to a reference genome is a widely used approach for identifying copy number changes in the human genome. Over the last decade, a number of statistical methods have been developed for identifying CNVs from WGS and targeted sequencing experiments [44, 45, 47, 106–108]. The vast majority of these methods calculate read-depth in non-overlapping windows of a fixed length across the genome and detect changes in the depth of coverage along chromosomes to identify CNVs. CNV detection from WGS has been shown to be more sensitive than array-CGH based CNV detection [109]. However, CNV detection methods for WGS data are designed to analyze genomic regions independently and either exclude genomic regions with low mappability from consideration or randomly place reads with low mapping quality [44] to avoid false positives. Therefore, such methods tend to have low accuracy for detecting copy number variation in LCRs. One exception is the GenomeSTRiP method that can detect CNVs in both unique and duplicated sequences [47].

Alkan et al. [110] developed a short read mapping algorithm, mrsFAST, that can identify multiple mapping locations for reads and used it to predict copy number in duplicated regions of the human genome. Building on this approach, Sudmant et al. [68] leveraged SUNs — paralogous sequence variants that uniquely tag a repeat copy — to estimate total copy number as well as paralog specific copy number for all duplicated genes in the human genmome. Analysis of WGS data from the initial phases of the 1000 Genomes project showed that almost half (49%) of duplicated genes are copy number invariable while the remaining set of duplicated genes show extensive copy number variation with many copies not represented in the reference human genome [68]. Recently, Shen et al. [46] have developed a computational tool QuicK-mer2 that leverages a similar approach to estimate paralog-specific copy number.

Since WGS is now widely used in the clinical setting for disease diagnostics, there is strong interest in developing computational tools that can detect both copy number and sequence variation in disease-relevant duplicated genes with high accuracy [50]. Several methods — designed specifically for individual genes such as *SMN1*, *STRC*, *PMS2* — have been developed for this purpose [10, 52, 111]. For example, the SMNCopyNumberCaller tool [10] is designed to estimate the copy number of *SMN1*, *SMN2* and a partially deleted version of *SMN2* from WGS data. Similarly, a workflow for detecting variants in the duplicated region of *PMS2* has also been developed [112]. Although these tools are valuable for analyzing duplicated genes, they leverage prior knowledge about individual genes and are not directly applicable to other duplicated genes.

Copy number analysis for duplicated genes requires joint analysis of reads that are mapped to homologous repeat copies [10, 47]. In this chapter, we describe a probabilistic method, Parascopy, for estimating total (and paralog-specific) copy number of low copy repeats (LCRs) in the human genome. Our method leverages a homology database that stores positional information about similar sequences in the human genome as well as the positions at which the paralogous sequences differ (PSVs or paralogous sequence variants). It uses the homology database to extract relevant reads from existing alignments of WGS data. To avoid pitfalls associated with using polymorphic PSVs for differentiating repeat copies, Parascopy jointly estimates paralog-specific copy number and reference allele frequencies for each PSV using WGS data for multiple samples. This also identifies common profiles of copy number variation that can be used to analyze individual WGS datasets. We benchmark Parascopy's accuracy using experimental copy number datasets, Mendelian trio consistency analysis and concordance analysis on replicate WGS datasets.

3.2 Results

3.2.1 Overview of method

Our method, Parascopy, is designed to estimate the aggregate copy number *(AggregateCN)* and paralog-specific copy number *(ParalogCN)* of low-copy repeats or LCRs in the human genome (Figure 3.1a). Even though a large fraction of short reads cannot be mapped unambiguously due to the repetitive nature of such loci, it is feasible to analyze read depth jointly across the different copies of a low-copy repeat and estimate the aggregate number of copies. For a LCR *R*, Parascopy uses a homology table to quickly identify all other regions in the genome that share high sequence similarity or homology with *R*. The homology table — similar to a segmental duplication database — stores all pairs of sequences in the genome (with a minimum length and minimum similarity score) and is precomputed using standard alignment tools (see Methods).

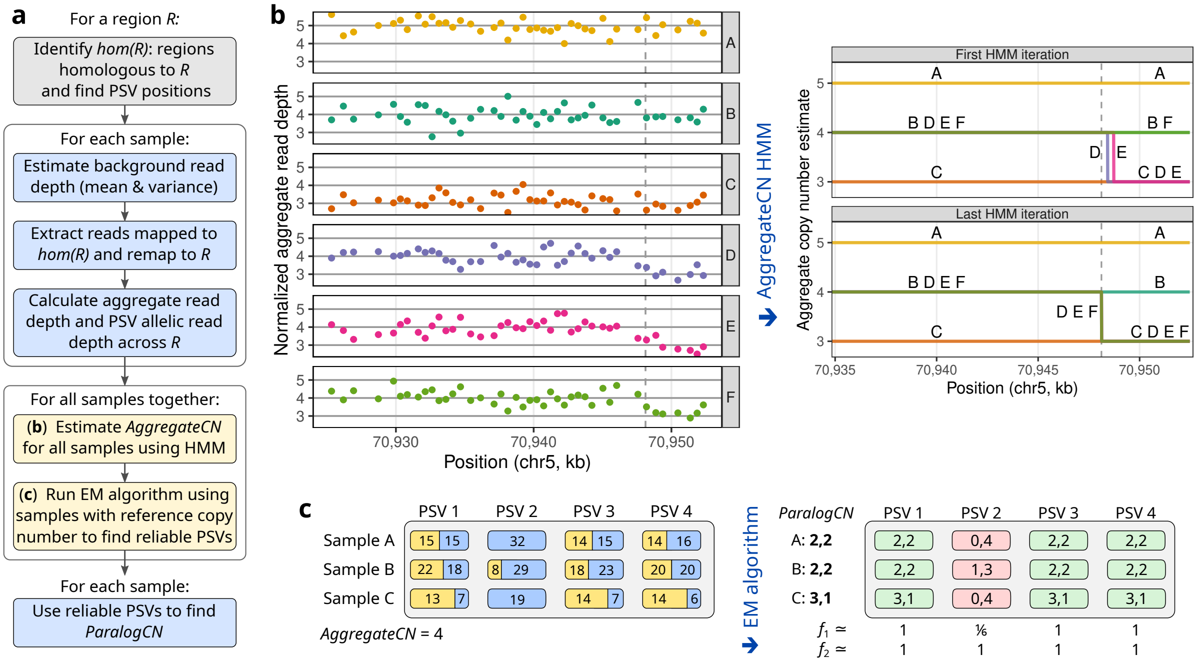

a
For a region R:
Identify hom(R): regions homologous to R and find PSV positions
For each sample:
Estimate background read depth (mean & variance)
Extract reads mapped to hom(R) and remap to R
Calculate aggregate read depth and PSV allelic read depth across R
For all samples together:
(b) Estimate AggregateCN for all samples using HMM
(c) Run EM algorithm using samples with reference copy number to find reliable PSVs
For each sample:
Use reliable PSVs to find ParalogCN
b
Normalized aggregate read depth
A
B
C
D
E
F
70,930
70,940
70,950
Position (chr5, kb)
AggregateCN HMM
First HMM iteration
Last HMM iteration
Aggregate copy number estimate
A
B D E F
C
D
E
B F
C D E
B
D E F
C D E F
70,935
70,940
70,945
70,950
Position (chr5, kb)
c
PSV 1
PSV 2
PSV 3
PSV 4
Sample A
Sample B
Sample C
15 15
32
14 15
14 16
22 18
8 29
18 23
20 20
13 7
19
14 7
14 6
AggregateCN = 4
EM algorithm
ParalogCN
A: 2,2
B: 2,2
C: 3,1
2,2
0,4
2,2
2,2
2,2
1,3
2,2
2,2
3,1
0,4
3,1
3,1
f₁ ≃ 1 ⅙ 1 1
f₂ ≃ 1 1 1 1

Figure 3.1. **Estimation of aggregate and paralog-specific copy number for low-copy repeats using Parascopy. a** Workflow of the method using aligned WGS reads for multiple samples as input to infer aggregate and paralog-specific copy number profiles across a genomic region. **b** Illustration of the iterative Hidden Markov Model (HMM) approach for estimating aggregate copy number *(AggregateCN)* profiles using normalized read depth for multiple samples. Read depth values are shown for six samples (A-F) at the *SMN1/2* locus (aggregated across *SMN1* and *SMN2*). The HMM identifies a partial deletion in samples D and E in the first iteration. Joint update of the HMM parameters results in detection of a common deletion event in the 3 of the 6 samples. **c** Illustration of the Expectation Maximization (EM) algorithm for estimating paralog-specific copy number *(ParalogCN)* and paralogous sequence variant (PSV) reliability. PSV reliability is measured using f-values that correspond to the population frequency of the reference allele for each PSV at each paralogous position.

Subsequently, reads from regions homologous to R are re-mapped to R and the aggregated reads are used to tabulate read depth in non-overlapping windows. A Hidden Markov Model (HMM) is used to segment R into regions of fixed copy number based on the read depth profiles and background read depth distributions. To account for variation in read depth across different genomic regions, the background read depth distributions are estimated for each sample and GC-content value using non-duplicated genomic regions (see Methods). The initial state distribution and transition probabilities of the HMM are estimated jointly across multiple samples enabling high sensitivity for the detection of copy number variants that are present in multiple samples (Figure 3.1b).

Once the aggregrate copy number profile has been estimated for each sample, Parascopy estimates the number of copies of each paralog present in the genome *(ParalogCN)* by analyzing allelic read depth at positions that differ between the homologous sequences, i.e. paralogous sequence variants or PSVs. Since some PSVs are not fixed in the population and correspond to variants, Parascopy jointly models frequency of the reference allele at each homologous position for each PSV and the *ParalogCN* for samples with *AggregateCN* equal to the reference. It considers all possible combinations of *ParalogCN* for each individual sample and uses an EM algorithm to infer maximum likelihood estimates for both sets of variables (Figure 3.1c).

3.2.2 Parascopy estimates copy number accurately and identifies reliable PSVs at the *SMN1/2* locus

The *SMN1/2* locus on chromosome 5 harbors the *SMN1* gene and its paralog *SMN2* in a tandem duplication of length ≈ 100 kilobases and very high sequence identity (99.9%). Mutations — point mutations and copy number changes — in the *SMN1* and *SMN2* genes cause a rare childhood disorder called spinal muscular atrophy (SMA) and *SMN1* is one of the most-studied duplicated genes in the genome. We estimated copy number for all 2504 samples with WGS data from phase 3 of the 1000 Genomes Project (1kGP) [2] using Parascopy (samples for each continental group were analyzed separately). Analysis of the Parascopy copy number profiles across the 1kGP samples identified a known deletion event that spans exon 7-8 (Figure 3.2a) and confirmed the extensive variation in *AggregateCN* (2-6) across human populations [10].

Vijzelaar et al. [113] used multiplex ligation-dependent probe amplification (MLPA) to estimate *AggregateCN* of each exon of the *SMN1/2* gene for 1109 1kGP samples. For the exon 7-8 region, the copy number values for 79 of the 1109 samples were consistent with

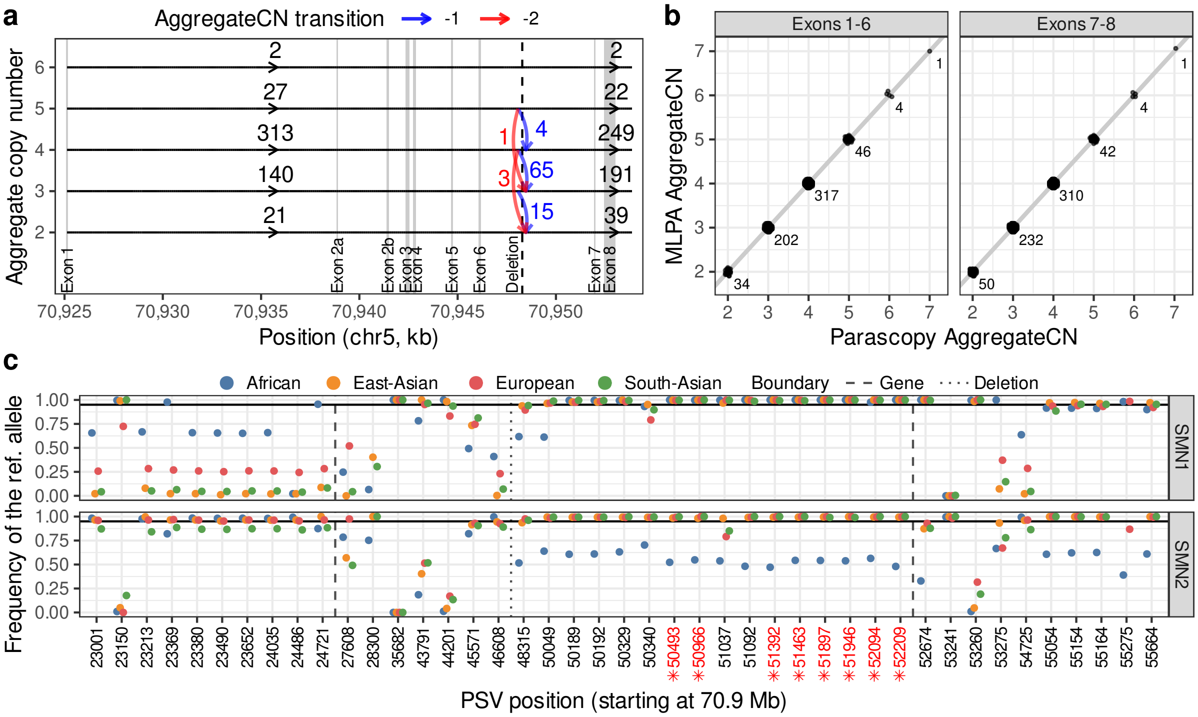


Figure 3.2. **Estimation of aggregate and paralog-specific copy number for the *SMN1/2* locus using Parascopy. a** Output from the Hidden Markov Model estimation of aggregate copy number *(AggregateCN)* profiles for 503 European ancestry samples from 1kGP. The common deletion event at the 3' end of the *SMN1/2* gene is shown using blue and red arrows. **b** Comparison of the Parascopy *AggregateCN* estimates with MLPA based estimates for exons 1-6 and exons 7-8 (with deletion). Labels represent the number of samples with the corresponding copy number estimates. **c** Distribution of the frequencies of the reference alleles (f-values) for 43 paralogous sequence variants (PSVs; 23 within *SMN1/2*) across four different 1kGP continental populations. The 8 PSVs used for estimating paralog-specific copy number by SMNCopyNumberCaller are highlighted in red.

the presence of the common deletion. The *AggregateCN* estimates from Parascopy were perfectly concordant with MLPA values [113] for both exons 1-6 and 7-8 (Figure 3.2b). We also compared Parascopy's accuracy for copy number estimation with three other existing methods: SMNCopyNumberCaller [10] — a method designed specifically to estimate copy number for *SMN1/2*; QuicK-mer2 [46] — an alignment-free approach to estimate *ParalogCN* using *k*-mers unique to paralogous sequences; and CNVnator [44] — a CNV detection algorithm that statistically analyzes read depth from WGS data. Both SMNCopyNumberCaller and QuicK-mer2 showed high accuracy for *AggregateCN* of exons 1-6 but CNVnator had a much lower accuracy

Table 3.1. **Accuracy of aggregate copy number estimation for three different methods across 10 duplicated genes in the human genome.** For each method, accuracy is the percentage of samples with identical WGS-based and experimental copy number values. Percentage of copy number estimates with high quality is shown in parentheses when it is below 100%. The third column in the table shows the mean and standard deviation (SD) of the experimental values. The reference copy number is 4 for all loci except for *SRGAP2* (8) and *AMY1* (6). For the *AMY1* locus, accuracy is estimated by computing mean absolute error (δ) due to high variance in copy number.

Duplicated gene	Sample size	Copy number mean ± SD	CNVnator	QuicK-mer2	Parascopy
SMN1/2	1109	3.7 ± 0.6	68.5 (99.9)	99.5	100.0*
C4A/B	45	3.8 ± 0.6	86.7	75.6	100.0*
FCGR3A/B	51	4.1 ± 0.5	94.1*	94.1*	94.1*
PMS2/CL	140	4.0 ± 0.0	67.7 (92.9)	97.9	100.0*
HYDIN/2	5	4.4 ± 0.9	100.0*	100.0*	100.0*
APOBEC3A/B	179	3.6 ± 0.6	94.4	96.1	96.9* (90.5)
RHD/RHCE	40	3.6 ± 0.8	97.5*	97.5*	97.5*
NPY4R/2	18	4.8 ± 0.8	66.7	77.8*	77.8*
SRGAP2	40	7.8 ± 0.7	82.5	62.5	100.0*
AMY1A/B/C (δ)	225	7.3 ± 2.6	0.887 (99.1)	1.119	0.723* (96.0)

* Highest accuracy for each gene.

equal to 68.5% (Table 3.1). For the exon 7-8 region, only SMNCopyNumberCaller showed high accuracy (sensitivity = 1.00 and specificity = 0.999). while both CNVnator (sensitivity = 0.823 and specificity = 0.849) and QuicK-mer2 (sensitivity = 0.709 and specificity = 0.382) had significantly lower accuracy (Table B.1). We also compared *ParalogCN* estimates from Parascopy with those from SMNCopyNumberCaller, QuicK-mer2 and CNVnator. While SMNCopyNumberCaller's estimates on 855 non-African samples were identical to Parascopy, QuicK-mer2 and CNVnator showed higher mean absolute difference of 0.53 and 0.23 respectively.

Unlike previous methods, Parascopy estimates the population frequency of the reference allele for each PSV (f-values), and only uses *reliable* PSVs — PSVs with $f \geq 0.95$ for all homologous positions — to estimate *ParalogCN*. Estimates of PSV f-values across the different populations showed that 10-19 of the 43 PSVs within and in the vicinity of *SMN1* were reliable for 4 of the 5 continental populations in the 1kGP while none of the PSVs were reliable in the African population samples (Figure 3.2c). This was consistent with the observations of Chen et al. [10] about the lower concordance between *ParalogCN* values at individual PSV sites. Notably, the set of PSVs identified as reliable by Parascopy included all 8 PSVs used for estimating *ParalogCN* by SMNCopyNumberCaller [10].

3.2.3 Parascopy outperforms existing methods for copy number estimation

Next, we benchmarked the accuracy of Parascopy on additional duplicated genes with experimentally determined copy number data. For this, we compiled previously published datasets with experimental copy number data for more than 1100 samples (from the 1kGP) across nine different genes apart from *SMN1/2*. First, we compared the accuracy of *AggregateCN* estimates obtained from Parascopy with CNVnator and QuicK-mer2 (Table 3.1). Across the 9 genes, *AggregateCN* estimates from Parascopy were either more accurate than both methods (*SRGAP2*, *C4A/B*, *PMS2*, *AMY1*) or equally accurate (*FCGR3A/B*, *HYDIN*, *APOBEC3A/B*, *RHD/RHCE*, *NPY4R/2*). For the *AMY1* locus, which has a high variation in total copy number (2–18) in human populations, Parascopy's mean absolute error was 0.72 compared to 0.89 and 1.12 for CNVnator and QuicK-mer2 respectively (Figure B.2). For the *APOBEC3A/B* locus, Parascopy's assigned low quality (< 20) to copy number values for 9.5% of samples due to the small length of the gene. The lowest accuracy (77.8% on 18 samples) for Parascopy was observed for the *NPY4R/2* locus. Visual inspection of read depth profiles at this locus for the 18 samples indicated that Parascopy's estimates are likely to be correct for all samples and were perfectly concordant with QuicK-mer2 estimates (Figure B.1).

Next, we assessed the accuracy of paralog-specific copy number estimation for the three methods across 4 of the 9 genes that had experimental paralog-specific copy number data (Table 3.2). Parascopy's average accuracy (87.58%) was greater than both CNVnator (76.97%) and QuicK-mer2 (66.06%). Estimation of *ParalogCN* depends on PSVs that can differentiate the

Table 3.2. **Accuracy of paralog-specific copy number estimates for three different methods using experimental copy number observations for 4 duplicated genes in the human genome.** The last column shows the number of reliable paralogous sequence variants (PSVs; 1kGP European samples) and the total number of PSVs within the duplicated gene or locus.

Duplicated gene	Sample size	CNVnator	QuicK-mer2	Parascopy	Reliable PSVs
SRGAP2	40	67.5	72.5	97.2*†	1461 / 1940
C4A/B	45	51.1	48.9	66.7*	7 / 50
FCGR3A/B	40	97.5*	47.5	97.5*	120 / 179
RHCE/RHD	40	95.0	97.5*	92.5	897 / 1027

* Highest accuracy for each gene.

† Paralog-specific copy number estimates have low qualities in 4 samples.

repeat copies and all methods had low accuracy for the C4/B locus which had a low number of reliable PSVs (7/50).

Finally, we compared the performance of the different methods for identifying the boundaries of copy number changes within a gene. For this, we analyzed the *PMS2/PMS2CL* locus where 4 of 150 1kGP samples were reported to harbor a partial deletion covering two exons (exons 13 and 14) using LR-PCR sequencing and MLPA [112]. Analysis of the *AggregateCN* profiles estimated by Parascopy's HMM showed that a partial deletion was correctly identified in 4/4 samples albeit with low quality (< 20) in 2 of the 4 samples (Figure B.3). Parascopy did not identify the deletion event in any of the remaining samples (sensitivity = 1.0 and specificity = 1.0). In contrast, QuicK-mer's copy number profiles showed no evidence of the deletion (sensitivity = 0.0 and specificity = 1.0) while CNVnator detected a copy number change in 3/4 samples (sensitivity = 0.75 and specificity = 0.691).

3.2.4 Accuracy of Parascopy copy number estimates across a set of genome-wide low-copy repeats

Next, we evaluated Parascopy's accuracy and robustness for estimating copy number across a larger set of duplicated coding loci in the human genome. For this purpose, we compiled a catalog of 167 low-copy repeat loci — overlapping over 220 protein-coding genes (380 including homologous regions) — using previous analysis of sequence homology of coding regions in the human genome [50] and copy number estimates for genes overlapping segmental duplications [68] (see Methods). These 167 low-copy repeat loci span 12.6 Mb of DNA sequence (including homologous regions) and 65.0 (14.7)% of these loci correspond to two (three) copy duplications.

First, to assess the robustness of the copy number estimates to variation in sequencing bias, we analyzed each of the 167 repeat loci in a set of 90 individuals of Han Chinese ancestry for which WGS data was generated independently by Lan et al. [114] using a PCR-based library preparation protocol. 83 of these 90 individuals also had WGS data available from the 1kGP generated using a PCR-free library preparation protocol. In comparison with the PCR-free data, the PCR-based WGS data exhibited significant greater biases in the distribution of read depth as a function of GC-content (Figures B.4 and B.5). We ran Parascopy, CNVnator and QuicK-mer2 on the two datasets independently and compared the concordance between pairs of replicate samples (across the 167 repeat loci) for each method. Parascopy reported *AggregateCN* estimates (with quality $\geq$ 20) for 94.5% of the pairs and 98.7% of the *AggregateCN* pairs were concordant. In comparison, QuicK-mer2 provided *AggregateCN* values for 100% of the pairs with a concordance rate of 74.9% (Table 3.3). CNVnator's concordance (86.9% with a completeness of 97.0%) was also significantly lower than Parascopy. Notably, Parascopy's concordance without any quality value filter (96.4%) was still 11.4 perecentage points greater than that for CNVnator. These results also showed that Parascopy *AggregateCN* values with quality $<$ 20 are less reliable.

Parascopy does not estimate *ParalogCN* values for loci that have high reference copy number or a low fraction of reliable PSVs (see Methods). As a result, the concordance analysis was limited to the 122 loci that had *ParalogCN* for one or more samples across both replicates. Across these loci, Parascopy's *ParalogCN* had a concordance rate of 99.8% (99.5%) for a quality threshold of 20 (0). Notably, the mean absolute difference between replicates was 0.003. In

Table 3.3. **Concordance of aggregate *(AggregateCN)* and paralog-specific *(ParalogCN)* copy number estimates across 167 duplicated loci between two replicate WGS datasets for 83 Han Chinese samples.**

Data type	Metric	CNVnator		QuicK-mer2	Parascopy	
		Q ≥ 0	Q ≥ 20		Q ≥ 0	Q ≥ 20
AggregateCN (167 loci)	Available estimates (%)	100.0	97.0	100.0	100.0	94.5
	Concordance (%)	85.0	86.9	74.9	96.4	98.7
	Mean absolute difference	0.185	0.157	0.292	0.041	0.014
ParalogCN (167 loci)	Available estimates (%)	100.0	97.0	100.0	72.8	70.1
	Concordance (%)	83.4	85.5	81.1	99.5	99.8
	Mean absolute difference	0.282	0.240	0.299	0.007	0.003
ParalogCN (122 loci)	Available estimates (%)	100.0	97.8	100.0	99.7	96.0
	Concordance (%)	91.4	93.3	85.4	99.5	99.8
	Mean absolute difference	0.129	0.101	0.186	0.007	0.003

Q ≥ 0 — use all copy number estimates; Q ≥ 20 — use only high quality copy number estimates. QuicK-mer2 does not have a quality measures, therefore all copy number estimates were used. 122 loci — a subset of loci where Parascopy estimates *ParalogCN* for at least one sample in both datasets.

comparison, QuicK-mer2 and CNVnator *ParalogCN* estimates were available for all loci and had a concordance rate of 81.1% and 85.5% respectively (Table 3.3). For the smaller set of 122 duplicated loci with *ParalogCN* estimates from Parascopy, QuicK-mer2 and CNVnator average concordance values were 85.4% and 93.3% respectively, higher than those for all loci. At the *SMN1* locus, the PSV f-values, estimated by Parascopy, were highly concordant between the two datasets ($r^2 > 0.92$) and the same set of 20 PSVs were identified as reliable in both datasets (Figure B.6).

Next, we used trio analysis to assess if the *ParalogCN* values estimated by Parascopy are consistent with Mendelian rules of inheritance. For this, we utilized 602 trios with WGS data from the expanded 1kGP dataset [115]. To account the uncertainty in the locus-specific *ParalogCN* values for a trio, we used a probabilistic method to calculate a probability that the trio *ParalogCN* values are concordant with Mendelian inheritance (see Methods). We analyzed trio concordance for 137 of the 167 loci, for which Parascopy could estimate high quality *ParalogCN* values. On average, 99.5% trios were concordant per loci, with 126 loci having at least 99% concordant trios. The concordance rate for the subset of locus-trio pairs for which the predicted *ParalogCN* for the child was greater than 2, was 95.5% (4776/5093).

Parascopy can estimate copy number values for individual samples by utilizing the model parameters (HMM parameters and PSV f-values) inferred from an independent set of samples (see Methods). To assess the accuracy of Parascopy for individual samples, we analyzed 210 samples from two populations in the 1kGP (IBS and CHB) and compared the *AggregateCN* values for each sample obtained by individual estimation (using the model parameters from the other population) with multi-sample estimation (all samples from each population analyzed jointly). Parascopy *AggregateCN* estimates were perfectly concordant (Table B.2) for 165 of the 167 loci. The two remaining loci (*PRAMEF1* and *RHPN2*) had a mean *AggregateCN* > 7 and did not have high quality estimates available for comparison. Similarly, *ParalogCN* estimates showed very high concordance equal to 98.9%.

The accuracy of copy number estimation is expected to improve with increasing read depth. The mean read depth for the 1kGP samples was 33×. To assess the accuracy of Parascopy at lower values of sequence coverage, we sub-sampled WGS data for 107 samples from the IBS population in the 1kGP to one-third and two-thirds of the original read depth, analyzed them using model parameters from a different continental population and compared copy number estimates with those obtained using the full coverage (see Methods). As expected, the percentage of high-quality *AggregateCN* estimates reduced with decreasing read depth: 94% at two-thirds and 88.3% at one-third coverage (Table B.2). Nevertheless, the high-quality *AggregateCN* and *ParalogCN* estimates had high concordance equal to 99.9% and 98.4% respectively at one-third coverage.

Parascopy is multi-threaded and can process multiple loci in parallel. Analyzing 503 European genomes from the 1kGP took 17 hours using 16 cores and required less than 12 Gb of memory. For a single genome with 30× WGS, Parascopy took 16 minutes to analyze 167 duplicated gene loci using 16 threads and required less than 5 Gb of memory. In comparison, CNVnator (QuicK-mer2) took 28 (36) minutes to analyze a single genome using 16 threads and required 12 (40) Gb of memory. We note that a direct comparison of run-time between Parascopy, CNVnator and QuicK-mer2 is difficult since CNVnator and QuicK-mer2 are genome-wide methods while Parascopy is a targeted copy number estimation method. Nevertheless, the low memory requirements and run-time for Parascopy allow it to scale up for analyzing thousands of samples efficienctly.

3.2.5 Analysis of copy number changes and PSVs across 2504 individuals

To explore the diversity of copy number at low copy repeat loci across populations and genes, we estimated copy number at the 167 repeat loci for all 2504 individuals from five continental populations sequenced in the 1kGP. For 151 of the 167 loci, *AggregateCN* values could be estimated with high confidence (quality ≥ 20) for at least 95% of the samples. High average copy number was the main reason for the low quality of the *AggregateCN* estimates at some loci. The mean *AggregateCN* was 4.39 (6.42) for the 151 (16) loci with ≥ 95% (< 95%) of the samples with high confidence *AggregateCN* values. Similarly, for 26 of the 167 loci, *ParalogCN* estimates were not estimated either due to a low number of reliable PSVs (e.g. *CFC1*) or due to the lack of a sufficient number of individuals with reference copy number (see Methods).

Not surprisingly, the most frequent copy number value for the vast majority (88.4%) of loci was equal to the reference copy number. Several disease-associated genes had a low variance in aggregate copy number (e.g. *HYDIN*) while other genes such as *SMN1/2* and *NEB* had a large variance in the copy number. Among 164 loci, 84 loci had 99% or greater of samples with *AggregateCN* equal to the reference (Figure 3.3b). For 15 loci, the *AggregateCN* for more than half of the samples was greater than the reference — likely due to a missing copy in the reference genome (hg38) used for analysis. Notably, the most frequent copy number for the *OTOA* gene locus (*OTOA* + *OTOAP1*) with a reference copy number of 4 was 6 (Figure 3.4a). To investigate this further, we leveraged the recent highly complete human genome assembly from the T2T consortium for the CHM13 cell line [1]. Alignment of the *OTOA* duplicated sequence

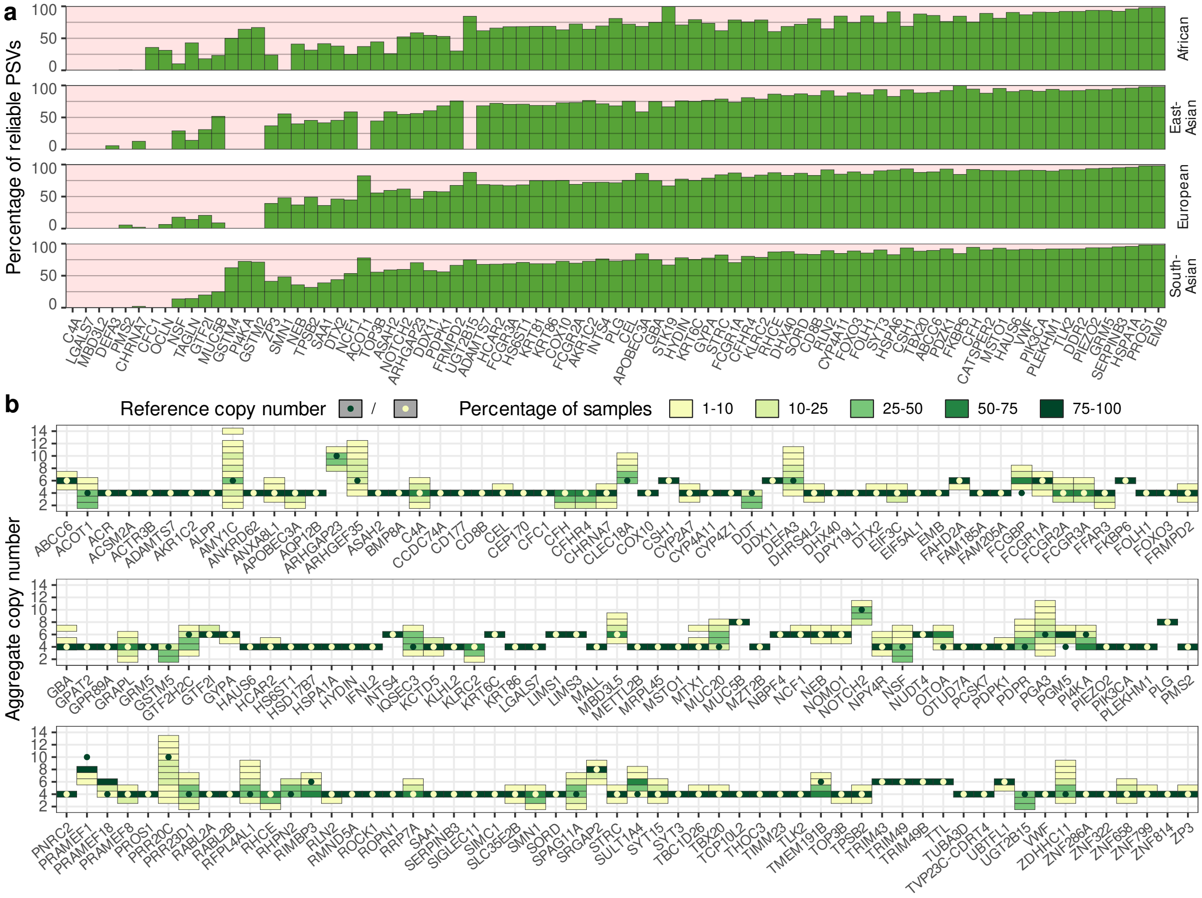


Figure 3.3. **Distribution of the percentage of reliable paralogous sequence variants (PSVs) and aggregate copy number *(AggregateCN)* profiles across duplicated genes.** **a** Percentage of reliable PSVs ($f \geq 0.95$) across 83 disease-associated genes and four continental populations from 1kGP. **b** Distribution of *AggregateCN* for 167 duplicated loci across all populations. Dark/white dots show reference copy number for each locus. Rare events (< 1% samples) are not shown.

to this assembly revealed the presence an additional copy that is not present in the current human reference genome and has sequence similarity ≥ 99.5% to the two other copies. We added this additional copy to the reference genome and re-analyzed the 1kGP samples using Parascopy. The *AggregateCN* estimates were not affected by the presence of the additional copy (concordance = 100%) demonstrating the robustness of Parascopy's *AggregateCN* estimates in the presence of missing repeat copies. In addition, we were able to estimate *ParalogCN* values and identify reliable PSVs using the sequence information from the additional copy. Analysis of

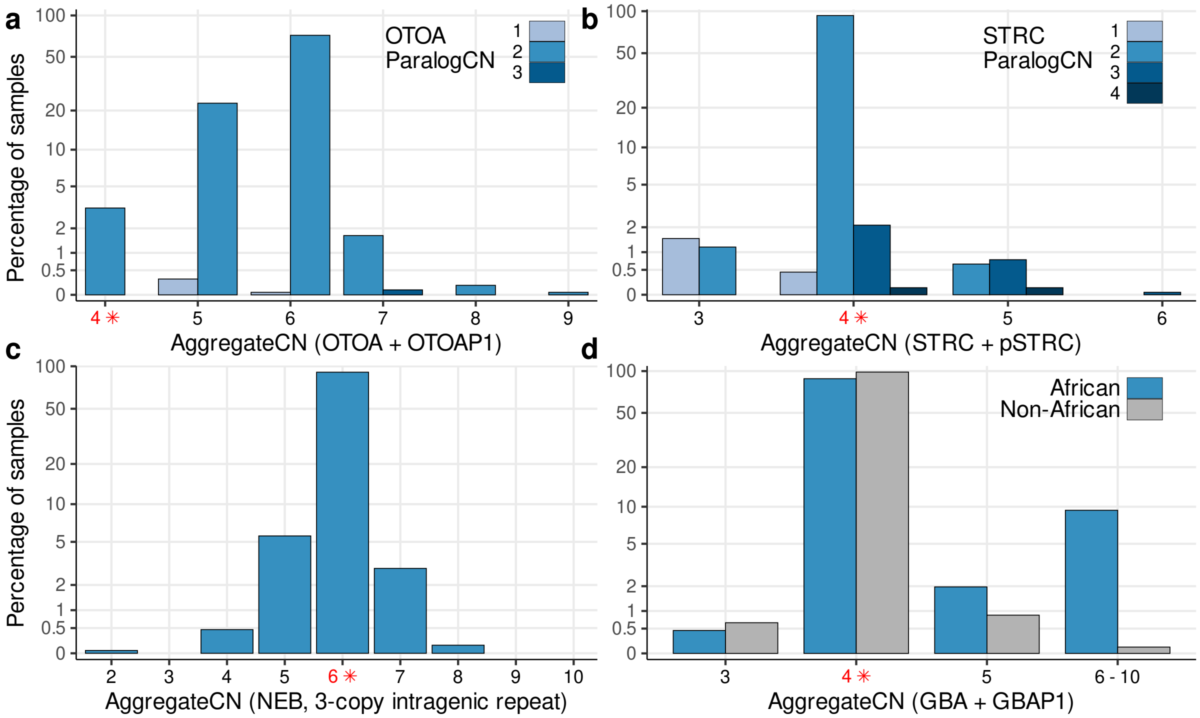


Figure 3.4. **Distribution of aggregate *(AggregateCN)* and paralog-specific *(ParalogCN)* copy number values across 2504 samples from 1kGP for four disease-associated genes.** The reference copy number is shown in red and marked with an asterisk. For the *OTOA* and *STRC* loci (panels **a** and **b**), the *ParalogCN* distribution for each *AggregateCN* bin is also shown. **a** For the *OTOA* gene, the most frequent *AggregateCN* is 6 while the reference copy number is 4, indicating the presence of a missing repeat copy in the reference genome. **b** 1.5% samples exhibit heterozygous deletion of the *STRC* gene while no samples have a homozygous deletion. **c** For the *NEB* gene, *AggregateCN* varies between 2-8 across 1kGP samples while previously reported pathogenic alleles at this locus had copy number ≥ 9. **d** A duplication event that includes both *GBA* and *GBAP1* is frequent in African populations (9.4% of individuals have *AggregateCN* of 6-10) and almost absent in non-African populations. For completeness, the panel includes samples with *AggregateCN* quality less than 20.

the *ParalogCN* values across the 1kGP data showed that *OTOAP1* locus is the most polymorphic in terms of copy number. For example, out of 540 samples with *AggregateCN* = 5, more than 93% samples were missing one copy of *OTOAP1*.

Next, we analyzed the frequency and distribution of copy number changes in individual disease-associated genes and their relationship with known pathogenic variants. For the *STRC* gene, approximately 1.5% of individuals across all continental populations were carriers of a heterozygous deletion of *STRC* while no individual had a bi-allelic deletion (Figure 3.4b).

Bi-allelic deletions in this gene are known to cause hearing loss [52, 116]. At the *GBA* locus — variants in *GBA* are associated with Gaucher disease[117] — we observed that 9.4% of individuals from African populations had an *AggregateCN* of 6 or more while only 2 individuals (0.1%) from non-African populations had such high copy number (Figure 3.4d). *GBA* and *GBAP1* (pseudo-gene) are located in homologous repeats separated by 10 kb on chromosome 1. Further analysis of copy number revealed that the increased copy number is a result of a duplication that includes the last two exons of *GBA*, part of *GBAP1* and the entire region between the two repeats (Figure B.7). For the *NEB* gene locus that harbors an intragenic repeat with three copies, the aggregate copy number varied from a minimum of 2 (one sample) to a maximum of 8 (population allele frequency of 0.12%, Figure 3.4c). Previous analysis of *NEB* copy number in 60 controls using a custom CGH microarray [118] had indicated that copy number gains of 2-4 copies could be pathogenic for nemaline myopathy. Our results on a much larger number of population samples indicate that copy number gains of 2 copies are observed at a low frequency and are unlikely to be pathogenic. Furthermore, the observed frequency of 1-copy gains and losses (3.1% and 5.4%) were consistent with those observed using CGH data (3.9% and 5.4%).

The fraction of reliable PSVs varied significantly across genes with some well-studied disease genes such as *C4A* and *PMS2* having a very low fraction of reliable PSVs while > 90% of the PSVs were reliable for genes such as *VWF* and *ABCC6* (Figure 3.3a). The fraction of reliable PSVs was highly correlated across populations ($r^2 = 90\%$, on average sets of reliable PSVs overlap by more than 95%) except for a few genes such as *SMN1* for which no reliable PSVs were identified for the African population. A high fraction of unreliable PSVs ($f < 0.95$) makes it challenging to estimate *ParalogCN*. Comparison of Parascopy's *ParalogCN* estimates with those estimated by QuicK-mer2 for several disease-associated genes (Figure B.8) showed that while the *AggregateCN* estimates were highly concordant between the two methods, the concordance of the *ParalogCN* estimates was low for genes with a high frequency of unreliable PSVs. For example, the correlation coefficient r^2 between the *ParalogCN* values for Parascopy and QuicK-mer2 (using 503 European samples) was 0.70 for the *FCGR3A* gene (67% reliable

PSVs) but it was only 0.29 for the *SMN1* gene (15% reliable PSVs). Additionally, Figure B.8 shows that when the fraction of reliable PSVs is low, QuicK-mer2 tends to generate ambiguous *ParalogCN* values: 49% of *SMN1 ParalogCN* values obtained using QuicK-mer2 are closer to a half-integer than to any integer.

A high frequency of unreliable PSVs is expected to adversely impact not only *ParalogCN* estimation but also short read mapping and variant calling since short-read mapping tools rely on PSVs to distinguish between different repeat copies. We used simulations to assess the impact of the frequency of reliable PSVs on variant calling accuracy at the *SMN1/2* locus. When all PSVs were reliable, state-of-the-art variant calling tools — GATK HaplotypeCaller [119] and FreeBayes [40] — achieved a recall of 0.52 and 0.55 respectively with a high precision (> 0.96) for variant calling. However, when we incorporated unreliable PSVs (identified from the analysis of 1kGP data, see Methods) in the simulated reads, the precision reduced significantly to 0.56 and 0.59 and the recall decreased to 0.25 and 0.29 for the two methods.

3.3 Discussion

In this chapter, we described a new computational method (and software tool), Parascopy, specifically designed for estimation of copy number for low-copy repeats in the human genome using WGS data. Parascopy leverages WGS data from multiple individuals to automatically account for sequencing biases and estimate aggregate and paralog-specific copy number profiles across specified region(s). Unlike some existing methods that require re-mapping or k-mer analysis of the entire WGS data, Parascopy uses a targeted approach that extracts and analyzes only reads relevant for each repeat loci from existing alignments. This allows it to efficiently estimate copy number for individual repeat loci across thousands of samples. We benchmarked Parascopy's accuracy using experimental copy number data for a number of genes and concordance analysis on replicate samples and it proved to be significantly more accurate than two existing methods — one designed for estimation of paralog-specific copy number (QuicK-mer2) and the second for genome wide copy number variant analysis (CNVnator). Parascopy's estimates of aggregate and paralog-specific copy number are robust to variation in sequencing biases and read depth as well as missing repeat copies in the reference genome.

A number of computational methods have been developed for detecting copy number variants from WGS data by modeling read depth [44, 47, 106, 120]. Most of the methods are designed for analysis of unique regions of the genome and do not focus on repetitive regions of the genome. Parascopy has been developed to fill this gap and uses a two-step approach where it first estimates aggregate copy number (by aggregating reads mapped to homologous regions) and then estimates paralog-specific copy number by careful modeling of PSVs. A similar approach has been used by the SMNCopyNumberCaller method [10] — a method designed for analysis of a single duplicated gene. However, Parascopy's general framework works for any low-copy repeat in the human genome and does not make assumptions about which PSVs can be used to distinguish the paralogous repeat copies. Instead, Parascopy explicitly models and estimates population allele frequencies for each PSV using WGS data for multiple samples and is the first method to do so. Analysis of WGS data at the *SMN1/2* locus demonstrated the ability of Parascopy to correctly identify reliable PSVs and also showed that using a fixed set of PSVs for estimating *ParalogCN* can potentially result in incorrect estimates.

Analysis of PSV allele frequencies using 1000 Genomes data showed that reliable PSVs were highly consistent across populations, however, the frequency of reliable PSVs varied significantly across genes. Information about reliable PSVs is not only useful for estimating paralog-specific copy number but is also relevant for read mapping and variant calling in LCRs. We have previously shown that post-processing of long read alignments using a probabilistic model that models genotypes for PSVs improves read mapping in LCRs [121]. It is well documented that short-read variant calling in LCR regions exhibits a higher rate of false negatives (due to low mappability) and false positives compared to unique regions of the genome [122]. Knowledge about reliable PSVs has the potential to improve short-read mapping and variant calling accuracy in such regions.

Parascopy has several limitations and avenues for further improvement. Parascopy's accuracy is lower for short regions and for regions with very high copy number (> 7). Nevertheless, Parascopy was able to estimate aggregate copy number with greater accuracy for the *AMY1* locus than existing methods. In addition, it cannot estimate *ParalogCN* for loci with high reference copy number (difficult to model large number of possible paralog-specific copy number values) or loci with a very low fraction of reliable PSVs. Parascopy currently works for only WGS data, however, information about allele-specific read depth at PSVs can potentially be used to infer copy number from targeted sequencing assays. Parascopy can estimate copy number for individual genomes using pre-computed model parameters, however, sample-specific sequencing biases may reduce the accuracy of copy number estimation. Parascopy assumes that the paralog-specific copy number for each sample is constant across the analyzed region. However, gene conversion events and hybrid alleles resulting from non-allelic homologous recombination are commonly observed at LCR loci [93, 123] and can result in non-uniform paralog-specific copy number. An HMM based approach can be used to model and detect such events and we plan to explore this in future work.

Unlike variants in unique regions of the genome, small sequence and copy number variants in duplicated genes are rarely analyzed in large-scale human genetic studies. Over the last few years, a number of large-scale WGS datasets for rare and common human diseases have become available [124, 125] and several others are expected to be available soon [126]. We expect Parascopy to be a valuable tool for analyzing such large-scale WGS datasets to identify novel genotype-phenotype associations. In addition, copy number profiles from such datasets will be useful for prioritizing pathogenic copy number changes in duplicated genes in the human genome. Finally, Parascopy can be useful for assessing the completeness and correctness of de novo assemblies at LCRs which can be challenging to assemble correctly even using long reads.

3.4 Methods

Given short sequence reads from WGS aligned to a reference genome for one or more samples, Parascopy jointly analyzes reads aligned to a genomic region *R* and its homologous sequences to estimate aggregate copy number *(AggregateCN)* — number of copies of *R* and its paralogs — as well as paralog-specific copy number *(ParalogCN)* — number of copies of each paralog. The estimation is performed jointly across all samples in two steps: (i) *AggregateCN* profiles are estimated first using read depth in fixed length windows and (ii) *ParalogCN* values are estimated using allele-specific read counts at PSVs and *AggregateCN* profiles. The workflow of the method is presented in Figure 3.1a. Before copy number estimation, background read depth distributions are estimated for each sample using reads mapped to unique regions of the genome.

3.4.1 Construction of homology table

Parascopy uses a precomputed table of homologous regions in the genome (homology table) to identify the paralogous regions for a given genomic region. This homology table stores pairwise duplications in a BED format that allows for indexing and fast retrieval of all duplications overlapping a given genomic region. For each duplication, we store a sequence alignment, length, sequence similarity and other characteristics, which allow for convenient filtering of duplications. Duplications with more than two copies are identified from overlapping pairwise duplications, and PSVs are extracted from the sequence alignments. The homology table is constructed by self-alignment of the reference human genome to itself using BWA [31] (see Section B.2.1). The table is designed to store primarily low-copy repeats; therefore, sequences with too many (> 10) pairwise alignments — that typically correspond to interspersed repeats — are discarded and the regions appropriately flagged in the table.

3.4.2 Normalization of read depth

To infer copy number from read depth, we use information from the observed read depth in a large number of non-repetitive regions of the genome (assumed to have a copy number of 2) for each sample. Briefly, read depth is calculated for windows of fixed length (default 100 bp) selected from unique regions of the genome by assigning mapped reads to the window that contains the center of the first read in each read-pair. Windows that have a high fraction ($\geq$ 10%) of (i) low mapping quality reads ($<$ 10), (ii) reads not mapped in proper pairs, or (iii) soft-clipped reads — are marked as *irregular* and not used for normalization. We considered several distributions to model the read depth distribution and found that the Negative Binomial (NB) distribution provided a better fit for the read depth distribution for PCR-based WGS data compared to the Poisson distribution (see Figure B.4 for an example). Therefore, we use the NB distribution to model the variation in read depth across windows in unique regions of the genome. To account for variation in read depth due to GC-content, we use separate NB parameters for each GC-content value (see Section B.2.2). This procedure is performed independently for each sample and only needs to be done once for each sample independent of the number of repeat loci.

We utilize a set of genomic windows used by the SMNCopyNumberCaller tool [10] for estimating background read depth. We split the set of regions into short windows of length 100 bp. To increase the number of windows with extreme GC-content ($\leq$ 35% or $\geq$ 55%), we select such windows in a 5-kb neighborhood of the original set of genomic regions. Finally, we discard all windows with a distance less than 500 bp to any duplication in the homology table. This procedure yields approximately 90,000 windows for both hg19 and hg38 versions of the human genome.

3.4.3 Identifying homologous regions and calculating aggregate read depth

For a given region R, we find all pairwise homologies overlapping R from the homology table. To reduce complexity, we skip short duplications ($<$ 500 bp). The homologous segments are used to split R into subregions or segments of constant *reference copy number.* Note that the reference copy number of a two-copy duplication is equal to 4 as the human genome is diploid. Next, for each sample, we extract reads aligned to the homologous regions and re-map these reads to the region R. This re-mapping is efficiently done using the precomputed alignments between R and its homologous sequences that are stored in the homology table. Each subregion of constant reference copy number is divided into non-overlapping windows of fixed length and aggregate read depth is computed for each window using the reads from the region R and the reads re-mapped to R. The read assigning procedure is same as for the background read depth analysis. For each window we calculate the fraction of reads with soft-clipping and the fraction of reads not mapped in a proper pair and filter out windows (and one flanking window on either side) if they are *irregular* in more than 10% of the samples.

For some loci, two subregions with the same reference copy number are interrupted by a short region with a different reference copy number (for example by an interspersed repeat). We group such subregions (if they are separated by less than 2000 bp) into *region groups* and all subsequent analysis is performed independently for each region group.

3.4.4 Estimating aggregate copy number profiles jointly for multiple samples

To estimate the aggregate copy number *(AggregateCN)* profiles, we construct a Hidden Markov Model (HMM) [127] for each region group. For a single sample s, the aggregate read depth values in the windows across the region represent the observed values and the *AggregateCN* value for each window forms a set of hidden states. For each window, we consider K possible *AggregateCN* values where K is selected based on the reference copy number for the region and observed aggregate read depth for all samples (see Section B.2.4.1).

To reduce the number of parameters, we define all transition probabilities based on two parameters for each consecutive pair of windows: $\tilde{a}_{\nearrow w},\ \tilde{a}_{\searrow w} \in \left[0, \frac{1}{10}\right]$. We define a jump from *AggregateCN* i to a larger *AggregateCN* j as $a_{ijw} = \tilde{a}_{\nearrow w}^{j-i}$ and to a smaller *AggregateCN* k as $a_{ikw} = \tilde{a}_{\searrow w}^{i-k}$. The transition probability a_{iiw} therefore equals $1 - \sum_{j \neq i} a_{ijw}$. By default, both $\tilde{a}_{\nearrow w}$ and $\tilde{a}_{\searrow w}$ are set to 10^{-5} on the first iteration, and the initial state distribution is set to $\pi_{\neg\text{ref}} = \frac{1}{|S|}$ for all non-reference copy number states and $\pi_{\text{ref}} = 1 - \frac{K}{|S|}$ for the reference copy number.

Let $o_w^{(s)}$ be the aggregate read depth for sample s in window w and let $n_w^{(s)}$ and $p_w^{(s)}$ be the parameters of the Negative Binomial (NB) distribution corresponding to the GC-content of window w (estimated separately for each sample). The emission probability for copy number c (hidden state) and window w is defined as: $b_w^{(s)}(c) = P_{\text{NB}}\left(o_w^{(s)};\ n_w^{(s)} \cdot c/2,\ p_w^{(s)}\right)$.

For each sample, we use the Forward-Backward algorithm [128] to obtain $\gamma_{c,w}^{(s)}$ — the probability that sample s has copy number c in window w. Next, HMM parameters — emission and transition probabilities as well as the initial state distribution — are updated iteratively using $\gamma_{c,w}^{(s)}$. The emission probabilities are updated using a scale parameter m_w for each window w that models window-specific biases in sequencing read depth that are not captured by GC-content based modeling [107]. In the first iteration, m_w is initialized to 1 for all windows. This parameter scales the expected read depth for each window (equally for all samples) to be higher or lower than the default value and is estimated using a maximum likelihood procedure and the $\gamma_{c,w}^{(s)}$ estimates (see Section B.2.4.3 and Figure B.5).

To update initial and transition probabilities, we use a procedure similar to Baum-Welch algorithm [127] (see Section B.2.4.4). The intuition underlying these updates is that the initial state probabilities correspond to population frequencies of the aggregate copy number values at the start of the region, while shared deletion or duplication events result in increased transition probabilities between the states of adjacent windows where the events happen. This iterative procedure — Forward-Backward algorithm for each sample followed by joint update of the HMM parameters — is run until the log-likelihood of the data (summed over all samples) converges (see Section B.2.4.5). Finally, we run the Viterbi algorithm [129] to find most probable *AggregateCN* profile for each sample.

3.4.5 Estimating paralog-specific copy number using PSVs

Once the *AggregateCN* profiles are estimated for each sample, we estimate paralog-specific copy number *(ParalogCN)* values using the allele-specific read counts at PSV sites across the region R. For a region with reference copy number c_r and a sample s with *AggregateCN* c_s, the paralog-specific copy number is defined as an integer tuple of length $c_r/2$ that sums up to c_s. Each tuple element represents the copy number of a specific copy of the duplication and order of the copies is the same for all samples. Note that there are $\binom{3/2 \cdot c_r - 1}{c_r}$ possible paralog-specific copy number tuples for each sample. We assume that the *ParalogCN* does not change in a region with constant *AggregateCN*.

PSVs are defined based on the reference genome assembly and only those PSVs for which the allele on a specific paralog of a duplication is invariant in the population are useful for estimating *ParalogCN*. Since some PSVs are known to correspond to polymorphisms, we model the frequency of the reference allele at a PSV site v and paralog k as f_{vk} where $f_{vk} \in [0, 1]$. We call a PSV v *reliable* if all its f-values are close to 1: $\min_{k=1}^{c_r/2} f_{vk} \geq 0.95$. Such PSVs can be used as markers of each paralog and are useful for estimating *ParalogCN*.

Given sequence data from multiple samples, we want to estimate two sets of variables: (i) *ParalogCN* for each sample, and (ii) PSV frequency matrix f where f_{vk} is the frequency of the reference allele for PSV v on the k-th copy. We use an Expectation-Maximization (EM) algorithm to solve this problem where sample *ParalogCN* values are hidden variables and the matrix f is an unknown parameter (see Section B.2.5.1 for details). In order to reduce computational complexity, we apply the EM algorithm only to those samples for which *AggregateCN* is equal to the reference copy number c_r (minimum of 50 samples). Once the PSV f-values are determined, we calculate the *ParalogCN* for all samples individually. For this, we run the E-step of the

EM algorithm using only reliable PSVs. Parascopy does not estimate *ParalogCN* for loci with very high *AggregateCN* or reference copy number (> 8) and for loci with very high number of possible *ParalogCN* tuples (> 500) to limit run-time.

3.4.6 Estimating copy number values for a single individual

Parascopy is designed to estimate *AggregateCN* and *ParalogCN* profiles using data for multiple samples, therefore, analyzing a single sample or a small number of samples may not produce very accurate results, particularly for *ParalogCN*. To enable the analysis of individual samples, we can use model parameters estimated from a population of samples, e.g. 1000 Genomes Project. Model parameters include initial and transition probabilities of the aggregate copy number HMM, as well as a set of window-specific scale parameters $\{m_w\}$ and PSV frequency matrix f. This allows us to quickly analyze individual samples using precomputed model parameters for multiple duplicated loci.

3.4.7 Genome-wide set of duplicated gene loci

To obtain a set of duplicated gene loci, we started with a set of 1168 duplicated genes that were reported previously [50] as having at least one exon that is difficult to map using short reads due to high sequence similarity to one or more other loci. From these, we removed 124 genes that are known to vary extensively in copy number [68] and 88 genes that are missing from the GENCODE annotation v37 [130]. Additionally, we discarded 257 genes that did not overlap any duplication longer than 2 kilobases in the homology table. The remaining genes were merged into 564 loci, which were then manually filtered in order to remove high copy number regions and regions with complex duplication structures. For some loci, we included additional flanking sequence to provide more useful information for copy number detection. The final set of duplicated gene loci set contained 167 regions, with all homologous regions covering 12.6 Mb.

3.4.8 Analysis of 1000 Genomes samples at 167 repeat loci using Parascopy

We used high-coverage (30×) whole-genome sequence data for 2,504 samples from the 1000 Genomes Project (1kGP) generated by the New York Genome Center [2]. All samples were sequenced using a PCR-free library preparation protocol and cram files aligned to the reference human genome hg38 were used for analysis. The 2,504 samples were divided into 5 groups based according to their continental population and each group of samples was analyzed together in a single run of Parascopy. To assess copy number concordance in trios, we analyzed WGS data from additional 698 related samples from the 1kGP resource. The 698 samples were analyzed independently of the 2,504 samples in a single run of Parascopy.

3.4.9 Copy number benchmarking using experimental data

For a given duplicated locus, Parascopy outputs integer *AggregateCN* and *ParalogCN* estimates for various subregions of the locus. In contrast, QuicK-mer2 and CNVnator output fractional *ParalogCN* values for various subregions throughout the whole genome and do not output *AggregateCN* directly. In order to facilitate a direct comparison, we extract Parascopy, QuicK-mer2 and CNVnator copy number estimates that overlap single positions within multiple copies of 167 duplicated loci. As CNVnator does not output *ParalogCN* estimates in the absence of deletions and duplications, we treat missing CNVnator values as *ParalogCN* = 2. Next, for each locus, we sum fractional *ParalogCN* values and round the sum to the nearest integer to obtain *AggregateCN* estimate; additionally, we round every fractional *ParalogCN* value to obtain integer *ParalogCN* estimates.

Throughout the chapter, we consider Parascopy copy number values to have high quality if their Phred-score is at least 20. Likewise, we convert CNVnator E-values into Phred-scores and say that the *AggregateCN* and *ParalogCN* estimates have high quality if the scores are ≥ 20 across all copies. We assume that all QuicK-mer2 copy number estimates have high quality, as the method does not output any quality measures.

To measure copy number estimation accuracy, we compare Parascopy, QuicK-mer2 and CNVnator *AggregateCN* and *ParalogCN* values derived from WGS data against corresponding copy number estimates based on experimental observations (Tables 3.1, 3.2 & B.1) for the same locus and the same individuals. If the experimental copy number observation is a fractional number, we round it to the nearest integer. Copy number estimate is *correct* if it matches completely with the experimental observation for the same sample. In Tables 3.1 and 3.2, QuicK-mer2 *AggregateCN* and *ParalogCN* values were aggregated across the duplicated genes, and median copy number value was selected. This procedure improved QuicK-mer2 accuracy; however, we did not perform it for all 167 duplicated loci, as it requires a careful case-by-case approach, especially in complex duplications.

SMNCopyNumberCaller [10] `v1.1.1`, QuicK-mer2 [46] `build 2021` and CNVnator [44] `v0.4.1` were run on the WGS datasets using default parameters. Additionally, QuicK-mer2 copy number estimates for 2457 1kGP samples were downloaded from https://github.com/KiddLab/kmer_1KG.

3.4.10 Assessing consistency of copy number estimates

To evaluate Parascopy, QuicK-mer2 and CNVnator robustness, we compare *AggregateCN* and *ParalogCN* estimates obtained for the same individuals based on two independent WGS datasets for 83 Han Chinese samples: PCR-free IGSR [115] dataset and PCR-based BGI [114] dataset. Copy number estimates were selected based on a set of positions within 167 duplicated loci in both datasets; in this way each sample is associated with 167 pairs of *AggregateCN* and *ParalogCN* values. A pair of copy number estimates is considered *available*, if the corresponding copy number estimates have high quality in both datasets. Accordingly, a pair of copy number estimates is *concordant*, if it is available and the corresponding copy number estimates match completely.

To assess robustness of Parascopy copy number estimates to variation in read depth and model parameters, we create two sets of Parascopy model parameters using 503 and 504 samples from the European and East-Asian continental populations, respectively. We analyze the same set of samples (103 Han Chinese samples and 107 Iberian samples) using both sets of model parameters and evaluate the concordance of resulting *AggregateCN* and *ParalogCN* values. Additionally, we subsample the 107 Iberian samples to one-third and two-third coverage, analyze subsampled datasets using East-Asian model parameters, and compare resulting copy number estimates against those obtained using full-coverage dataset and European model parameters.

3.4.11 Paralog-specific copy number validation using trios

In order to assess the accuracy of paralog-specific copy number estimates using Parascopy, we analyzed 602 trios with WGS data from the extended 1kGP [115]. The child in each trio was analyzed independently from the two parents to avoid any bias (except for 9 trios that consisted entirely of IGSR relatives). To assess consistency of *ParalogCN* values in trios, we modeled the population frequencies of paralog-specific copy number for a single chromosome using the observed diploid observations (see Section B.2.8). For each trio with high quality (≥ 20) *ParalogCN* estimates, we calculated the probability of observing the child's *ParalogCN* given the *ParalogCN* estimates of both parents. A trio was considered *discordant* if this probability was less than 0.01.

3.4.12 Measuring the effect of unreliable PSVs on variant calling

In order to evaluate the consequences of unreliable PSVs ($f < 0.95$) on variant calling, we used the NEAT short-read simulation tool [131] `v3.0` to generate a baseline set of single nucleotide variants (SNVs, ≈ 1 SNV per 1 kb) and high-coverage (30×) WGS data with and without unreliable PSVs. To introduce unreliable PSVs we randomly replaced PSV reference alleles with alleles from another copy of the duplication according to the frequencies of PSV reference alleles (f-values) in *SMN1/2* locus in 503 European samples from the 1kGP. This procedure yielded 31 homozygous and 33 heterozygous SNVs corresponding to unreliable PSVs. Next, we called variants in both copies of the duplication (total length 111 kb) using GATK HaplotypeCaller [119] `v4.2.2` and FreeBayes [40] `v1.3.5` and compared results with baseline sets of SNVs using RTG tools [78, 79] `v3.12.1`.

3.4.13 Data Availability

The analyses presented in this chapter are based on the high-coverage whole genome sequencing data of 1000 Genomes Project samples that was generated at the New York Genome Center with funds provided by NHGRI Grant 3UM1HG008901-03S1. This sequencing data is available via ENA Study PRJEB31736 and PRJEB36890. The whole-genome sequence data for 90 Han Chinese samples is available from ENA Study PRJEB11005. For this dataset, we used aligned reads downloaded from https://ftp.1000genomes.ebi.ac.uk/vol1/ftp/data_collections/han_chinese_high_coverage for analysis.

3.4.14 Code Availability

Parascopy is implemented in the Python programming language and is freely available for download at https://github.com/tprodanov/parascopy. It is also available via conda (`conda install -c bioconda parascopy`). Parascopy requires BAM/CRAM files for one or more samples, a reference genome sequence and a homology table (provided for human reference genomes hg19 and hg38) as input.

3.5 Acknowledgements

We thank Vineet Bafna and Melissa Gymrek for helpful discussions and feedback on the computational methods described here. We would also like to thank Raymon Vijzelaar for sharing updated MLPA copy number data for the *SMN1* locus. This work was supported by a grant from the National Human Genome Research Institute (NHGRI; R01 HG010149).

Chapter 3, in full, is a reformatted reprint of “Robust and accurate estimation of paralog-specific copy number for duplicated genes using whole-genome sequencing”. Timofey Prodanov & Vikas Bansal. *Nature Communications* **13**, 3221 (2022), https://doi.org/10.1038/s41467-022-30930-3. The dissertation author was the primary author of this paper.

CHAPTER 4

Variant Calling in Segmental Duplications

4.1 Introduction

Segmental duplications — also known as low copy repeats — constitute ~5% of the human genome and overlap more than 900 protein coding genes [49]. Genes that have been recently duplicated or have high sequence homology to other loci are problematic for NGS since short reads derived from such genes have ambiguity in their alignment and are difficult to correctly position in the genome. Short Illumina reads that originate from duplicated genes with high sequence homology align to multiple locations in the genome and are assigned low mapping quality scores [31, 38]. Such reads are typically discarded during variant calling by existing state-of-the-art variant calling tools such as GATK and FreeBayes to avoid false positive variant calls [40, 119]. A recent analysis of sequence homology for coding regions in the human genome identified 7,691 exons in 1,168 genes which have partial or complete sequence homology (> 98%) to one or more loci [50]. ~1,500 of these exons are problematic for variant detection even using Sanger sequencing [50].

Paralogous sequence variants (PSVs) refer to nucleotide differences (single base changes or small insertion/deletions) between duplicated loci in the genome [132]. Although PSVs can be used to place reads in duplicated genes, some PSVs are polymorphic and correspond to variants in the population rather than to the fixed differences between the duplicated copies [68, 100]. Proximally located duplicated genes frequently exchange short DNA segments via intra-locus gene conversion (IGC) [93] which can introduce polymorphic variants at PSV sites. Therefore, relying on PSVs to map reads can lead to incorrect read mapping within duplicated genes, which in turn can lead to inaccurate variant calls. Consequently, it is important to determine PSV genotypes before using PSV sites for read placement in duplicated genes.

193 of the 1,168 genes with high sequence homology are associated with rare Mendelian disorders, inherited cancers and complex diseases. For example, mutations in *PKD1* account for 85% of cases of autosomal dominant polycystic kidney disease (ADPKD1), one of the most frequent monogenic disorders with a prevalence of 1 per 1,000 individuals [133, 134]. The American College of Medical Genetics (ACMG) recommends carrier screening for many genes (e.g. *PMS2*, *STRC*, *SMN1*, *CYP21A2*) that are duplicated [135]. To enable mutation detection in these genes, specialized and labor intensive diagnostic assays have been developed that typically utilize long-range PCR in combination with Sanger sequencing [52, 136, 137]. Information about variants in duplicated genes is present in NGS read data but current state-of-the-art variant calling methods [119, 138] that rely on read mapping qualities are not well-suited for extracting this information [139].

We describe a probabilistic variant calling method for duplicated genes that jointly analyzes reads aligned to a duplicated gene and its repeat copies and does not rely on read mapping quality. This method can detect variants in regions that are not uniquely mappable even with Sanger reads. Identifying such variants with positional ambiguity is useful since if the variant is predicted to impact gene function (e.g., loss-of-function variant), the correct location of the variant can be determined using targeted approaches. Our method complements existing variant calling tools for NGS and enable the detection of variants in more than 600 duplicated genes in the human genome.

4.2 Results

4.2.1 Overview of method

ParascopyVC is a variant calling method for short-read whole-genome sequence (WGS) data that is designed specifically for low-copy repeats. Unlike standard variant calling methods such as GATK [41] and Freebayes [40], it performs variant calling and genotyping jointly across repeat copies for a duplicated region and utilizes reads with low mapping quality — even those with multiple equally likely mappings — for variant calling. ParascopyVC performs variant calling uses a two-step approach: (i) pooled variant calling and (ii) paralog-specific genotyping. First, reads aligned to a duplicated region *R* and its repeat copies are extracted from existing alignments and re-mapped to the duplicated region *R*. "Pooled variant calling" is performed on the re-mapped reads using a state-of-the-art variant calling method, Freebayes (see Figure 4.1). Aggregate copy number for the region *R* is estimated using the Parascopy copy number tool [140] and used as ploidy for the variant calling. During the second step, ParascopyVC attempts to estimate paralog-specific genotypes for each variant identified from the pooled variant calling. For this purpose, paralog-specific copy number and population reference allele frequencies, estimated using Parascopy, are used to identify informative Paralogous sequence variants (PSVs) that can differentiate the repeat copies for variant calling. A likelihood model is used to estimate paralog-specific genotypes for each variant using paired-end reads that overlap informative PSVs (Figure 4.1). Note that paralog-specific genotypes cannot be estimated for all variants and hence some variants have "positional ambiguity". Nevertheless, knowledge about such variants is useful for downstream analysis and is reported in the pooled variant call output file.

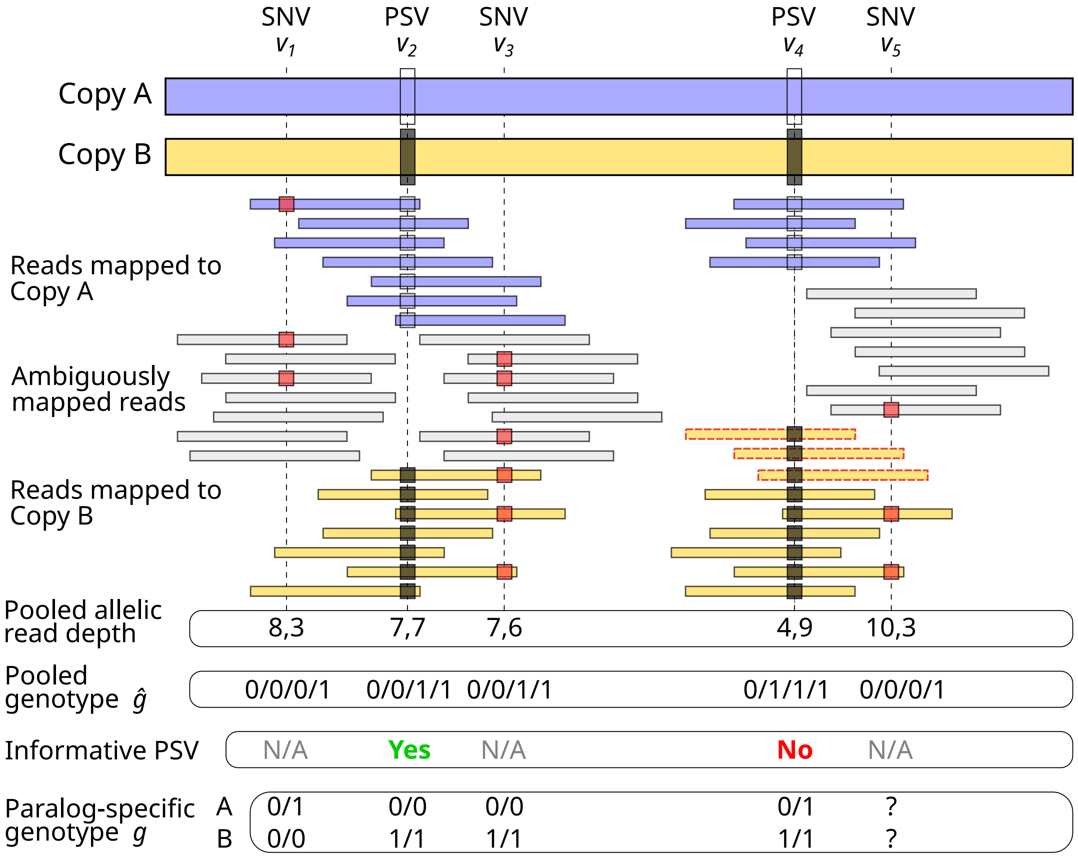


Figure 4.1. **Approach for variant detection and genotyping in duplicated genes.** Reads aligned to repeat copies 'A' and 'B' are shown in blue and yellow, respectively. Gray reads do not overlap any paralogous sequence variants (PSVs), and cannot be mapped unambiguously. Five variant sites were identified from the pooled reads: two PSVs and three single nucleotide variants (SNVs). Using genotype information from population WGS data, we can infer that variant v_2 is a non-polymorphic PSV site. Therefore, v_2 paralog-specific genotypes are $g_{v_2}^{(A)} = 0/0$ and $g_{v_2}^{(B)} = 1/1$. We can infer paralog-specific genotypes of SNVs v_1 and v_3 based on the reads that overlap both the variants and the *informative* PSV v_2. Pooled genotype of the PSV v_4 is not consistent with the reference, therefore the PSV v_4 is not informative and we cannot infer paralog-specific genotype of the nearby SNV v_5. Three reads — denoted by the red dashed border — originated from the copy 'A', but exhibit the copy 'B' allele at the v_4 site, and, as a result, are incorrectly mapped to the copy 'B'.

4.2.2 Evaluating the accuracy of paralog-specific variant calling on simulated data

To evaluate the accuracy of locus-specific variant calling, we generated two WGS variant call sets SIM-R and SIM-U. Both simulated call sets contain diploid artificial variants, however, while the SIM-R dataset does not contain any variants overlapping PSV sites, SIM-U dataset contains polymorphic PSV sites according to the PSV allele frequencies in the European population. For both simulated call sets, we generated four sequencing datasets (see Section 4.3.7) for a total of eight simulated WGS datasets. Finally, artificial reads were mapped to the GRCh38 reference genome using BWA-MEM [31, 32] and sequence variants were called using two state-of-the-art variant callers: FreeBayes [40] and GATK HaplotypeCaller [41].

In order to evaluate locus-specific variant calling within and in the vicinity of the duplicated genes, we utilized 167 low-copy repeat loci, compiled in Prodanov & Bansal (2022) [140]. Considering all repeat copies, the 167 loci span 12.6 Mb and cover 380 protein-coding genes. For each of the 167 duplicated loci, we obtained aggregate and paralog-specific copy number estimates using Parascopy [140] and called paralog-specific and pooled variants using ParascopyVC.

Considering the simulated sequencing datasets with 150 bp paired-reads and 30× coverage, benchmarking regions for the SIM-R and SIM-U datasets (see Section 4.3.8) cover 9.86 Mb and 9.28 Mb, respectively, and the corresponding benchmarking variant call sets contain 9,257 and 19,502 ground truth variants. On the SIM-R dataset (no polymorphic PSV sites), GATK, FreeBayes and ParascopyVC show similar recall (quality threhsold = 20): 0.8719, 0.8483 and 0.8814, respectively, as well as very high precision > 0.996 (see Figure 4.2). Note, that even though ParascopyVC uses FreeBayes to call pooled variants, ParascopyVC recall for paralog-specific variants is 3.3% higher than that of FreeBayes.

Nevertheless, once the genome contains polymorphic PSV sites, standard variant calling accuracy falls drastically. On the SIM-U dataset GATK obtains F_1 score of 0.7654 (recall = 0.7082, precision = 0.8327), FreeBayes shows a slightly higher F_1 score of 0.7889 (recall = 0.6961, precision = 0.9103). In contrast, ParascopyVC achieves F_1 score of 0.8937 with $> 10\%$ higher recall (0.8099) and almost perfect precision (0.9969) (see Figure 4.2). Across the benchmarking regions ParascopyVC incorrectly calls only 48 variants, while FreeBayes and GATK incorrectly identify 1295 and 2738 variants, respectively.

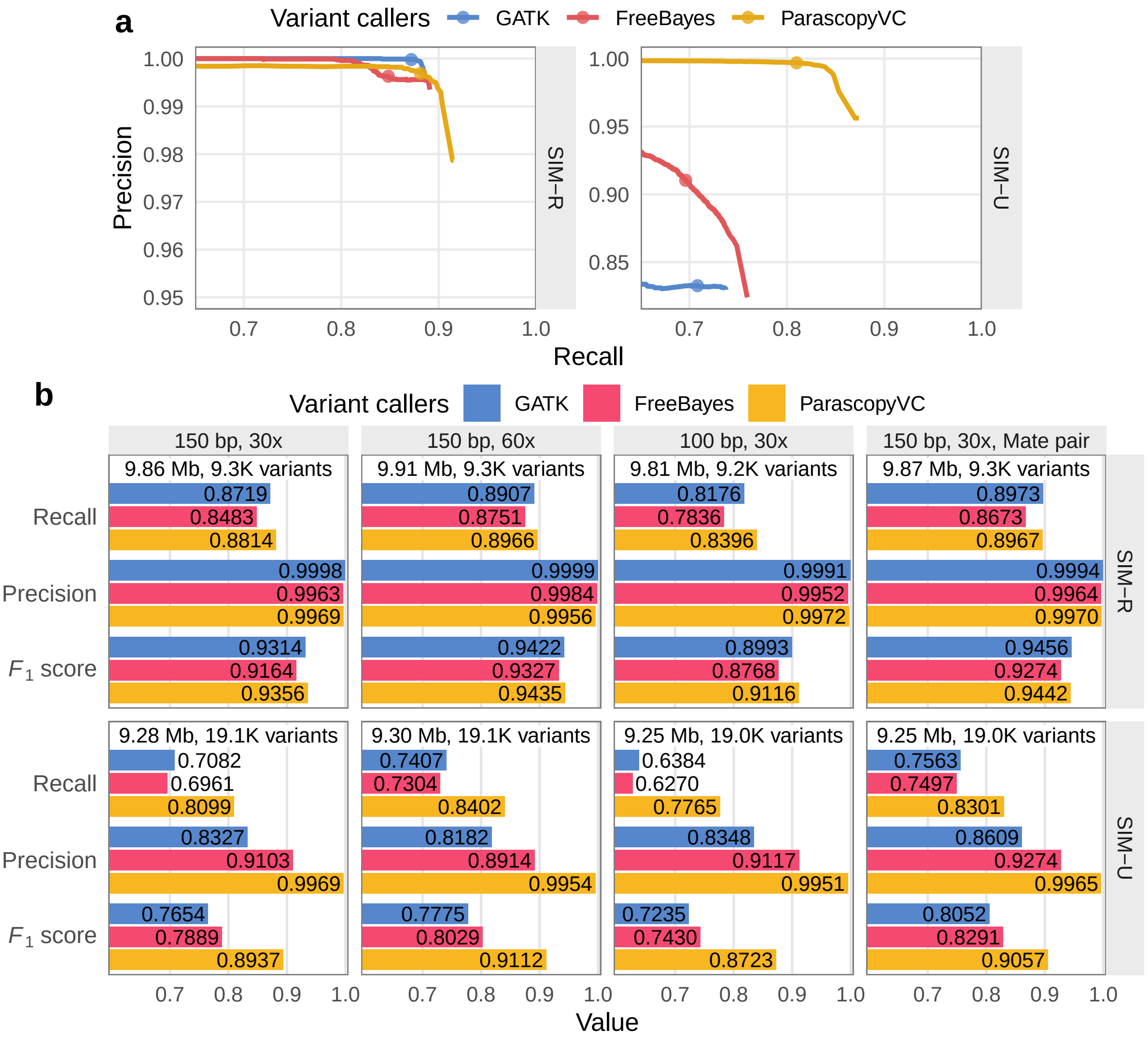


Figure 4.2. **Precision and recall of paralog-specific variant calling on two simulated WGS samples.** Precision and recall are shown for three variant callers: GATK (blue), FreeBayes (red) and ParascopyVC (yellow); and for two simulated variant sets: SIM-R (no polymorphic PSVs) and SIM-U (some polymorphic PSVs). **(a)** Precision-recall curves for two simulated variant sets, artificially sequenced using 150 bp paired reads with 30× coverage. Dots show precision-recall values at the quality threshold = 20. **(b)** Two rows show recall, precision, and F_1 scores for two simulated variant sets SIM-R and SIM-U. Four columns show artificial sequencing datasets with various sequencing features (left to right): 150 bp paired reads with 30× and 60× coverage; 100 bp paired reads with 30× coverage; and 150 bp mate pair reads with 30× coverage. The total length of the benchmarking regions and the number of baseline variants are shown on the top of each barplot.

To measure the effect of read depth, read length and fragment size, we compared called variants across six more sequencing datasets (three for SIM-R and for SIM-U). Raised read depth

(30× → 60×) improves recall by 1.5–2.7% on the SIM-U dataset, and by 3.0–3.5% on the SIM-R dataset at the expense of diminished precision (-1.5%) for FreeBayes and GATK (see Figure 4.2b, second column). Variant calling on shorter simulated reads (100 bp instead of 150 bp) produces similar precision compared to variants obtained using 150 bp reads (see Figure 4.2b, third column). However, variant calling recall drops significantly, with the decrease ranging from 3.3% (ParascopyVC on SIM-U) to 6.9% (FreeBayes, GATK on SIM-U). Finally, we simulated mate-pair reads with larger mean fragment size (2500 bp instead of 500 bp). Variant calling using mate-pair reads produced higher recall than using regular paired-reads by approximately 2% for all variant callers (see Figure 4.2b, fourth column). Additionally, variant calling precision improved by 1.7% and 2.8% for FreeBayes and GATK on SIM-U dataset, and did not improve in other cases, as the precision was already high in the initial sequencing datasets. Note, that ParascopyVC produces higher precision, recall and F_1 scores than FreeBayes and GATK across all simulated sequencing datasets on the SIM-U variant call set. Additionally, ParascopyVC produces higher or equal precision, recall and F_1 scores on the SIM-R variant call set.

4.2.3 Evaluating the accuracy of locus-specific variant calling on seven WGS datasets

The Genome in a Bottle Consortium (GIAB) compiled seven high-confidence variant call sets [76, 87] for individuals HG001–HG007, obtained by careful aggregation of variant call sets based on multiple variant calling tools and multiple sequencing technologies. We used available Illumina WGS data for each of the seven individuals and called variants using FreeBayes and GATK.

Benchmarking regions for the HG002 WGS dataset cover 5.96 Mb (see Section 4.3.8) and contain 7,645 ground truth variants. At a quality threshold = 20 GATK correctly calls 6,490 variants (recall = 0.8489), while also calling 802 false variants (precision = 0.8900), achieving a combined F_1 score of 0.8690. At the same quality threshold FreeBayes calls 6,470 true variants

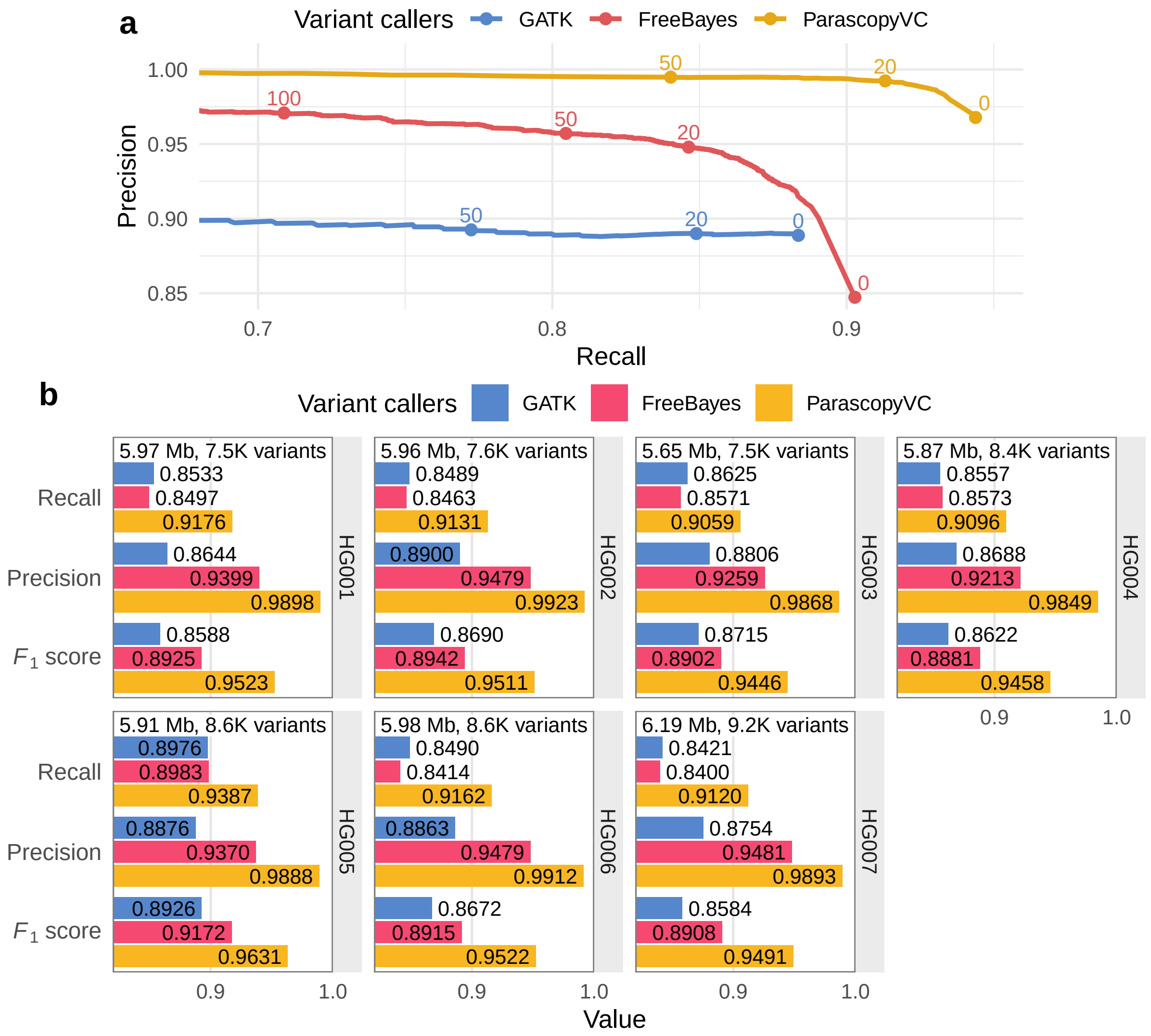


Figure 4.3. **Precision and recall of paralog-specific variant calling on the GIAB benchmark datasets.** Precision and recall are shown for three variant callers: GATK (blue), FreeBayes (red) and ParascopyVC (yellow). **(a)** The lines show variant calling precision-recall curves for the HG002 WGS dataset. Labels show specific quality thresholds (0, 20, 50 & 100). This chapter uses quality = 20 as the default threshold. **(b)** Seven barplots show recall, precision and F_1 score for seven GIAB benchmark datasets. The total length of the benchmarking regions and the number of baseline variants are shown on the top of each barplot.

(recall = 0.8463) and 346 false variants (precision = 0.9479), achieving a higher F_1 score of 0.8942. In contrast, ParascopyVC correctly calls 6,981 variants (recall = 0.9131) and incorrectly calls only 53 variants (precision = 0.9923), resulting in a high F_1 score of 0.9511 (see Figure 4.3a). Even at a quality threshold = 0 (comparing all called variants irrespective of their quality), ParascopyVC has higher precision (0.9678, recall = 0.9438) than the highest GATK precision (quality = 78, precision = 0.9002, recall = 0.6607), while FreeBayes achieves this precision only at quality = 87 (precision = 0.9678, recall = 0.7349).

Benchmarking regions for the remaining six GIAB variant call sets (HG001, HG003–HG007) cover on average 5.93 Mb of the LCR regions and contain on average 8,296 baseline variants (see Figure 4.3b). On all six datasets ParascopyVC shows higher precision and recall than both FreeBayes and GATK. Across all six datasets ParascopyVC precision does not fall below 0.9849, while the top precision another variant caller reaches is 0.9481 — approximately 3.7% smaller. On average, ParascopyVC achieves F_1 score = 0.9512 with standard deviation = 0.0061, while the variant callers FreeBayes and GATK reach F_1 scores 0.8949 ± 0.0100 and 0.8685 ± 0.0117, respectively.

4.2.4 ParascopyVC provides accurately finds pooled variant genotypes

In contrast to other variant calling tools, ParascopyVC finds variants pooled across multiple repeat copies. This may be beneficial in cases when repeat copies are almost indistinguishable from each other, and, as a result, paralog-specific genotypes cannot be identified. We grouped pooled benchmarking regions (see Section 4.3.9) by the aggregate copy number (4, 6 and 8), and compared ground truth variants with ParascopyVC pooled variant calls.

In total, benchmarking regions for the simulated SIM-R dataset cover 4.83 Mb and 76,668 baseline variants. Even though ParascopyVC precision remains very high ($\geq$ 0.9985) at all copy number values, recall drops as the copy number raises: 0.9847, 0.9119 and 0.7214 in case of the aggregate copy numbers 4, 6 and 8, respectively. Benchmarking regions for the SIM-U dataset cover 4.83 Mb and 74,683 baseline variants. The number of baseline variants is smaller for the SIM-U dataset for the following reason: all PSVs in the SIM-R dataset are represented by pooled variants, however, some homozygous unreliable PSVs in the SIM-U dataset will be absent from the list of pooled variants. ParascopyVC shows high equally high precision ($\geq$ 0.9989) even in

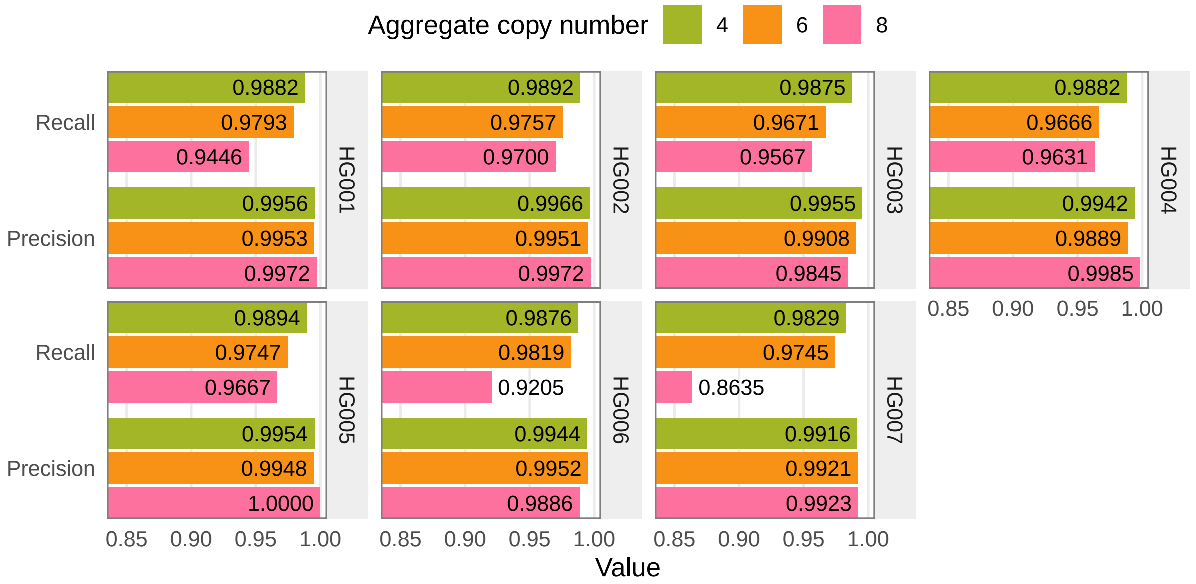


Figure 4.4. **Accuracy of ParascopyVC pooled variant calling on seven WGS datasets.** Seven barplots show recall and precision of ParascopyVC at separate aggregate copy number values: 4 (green), 6 (orange) and 8 (pink).

the presence of unreliable PSVs, coupled with diminished recall: 0.9746, 0.8443 and 0.5938 at copy numbers 4, 6 and 8, respectively.

Pooled variant calling finds more variants compared to paralog-specific variant calling. In two-copy duplications ParascopyVC identified 7,705 true positive non-PSV variants with variant quality ≥ 20 on the SIM-U dataset. At the same time, only 5,063 of these variants (72.7%) were identified by ParascopyVC as paralog-specific variant calls on any of the repeat copies with variant quality ≥ 20. Similarly, only 1,323 (487) variants were identified as both pooled and paralog-specific calls out of the total 2,779 (1,797) true positive pooled variants in three-copy (four-copy) duplications.

ParascopyVC shows much high precision (> 0.995) and recall (≥ 0.97) on the HG002 WGS dataset at all aggregate copy number values (see Figure 4.4), where benchmarking regions cover 2.37 Mb and 36.180 high-confidence pooled variants in total. Across all seven GIAB WGS datasets, benchmarking regions cover on average 2.47 Mb and 37900 high-confidence variants. Across all aggregate copy numbers and all WGS datasets, ParascopyVC shows very high precision (mean 0.9940 ± standard deviation 0.0036). As expected, recall drops with rising aggregate copy number: 0.9876 ± 0.0022 at two-copy duplications, 0.9743 ± 0.0057 at three-copy duplications and 0.9407 ± 0.0380 at four-copy duplications.

4.3 Methods

ParascopyVC variant calling runs on top of existing copy number analysis performed by Parascopy [140] for a single or multiple samples within a set of duplicated loci. Parascopy output includes (i) reads pooled from various repeat copies; (ii) reference allele frequencies (f-values) for paralogous sequence variants (PSVs); and (iii) aggregate and paralog-specific copy number estimates. ParascopyVC utilizes a modified FreeBayes [40] variant calling tool to call polyploid sequence variants based reads pooled from all repeat copies. As input, FreeBayes is provided with pooled reads for a single or multiple samples and a ploidy map that stores aggregate copy number for each sample and for various subregions of the duplicated locus. FreeBayes is modified solely to output more information: in addition to identifying potential variants, modified FreeBayes outputs detailed read-variant observations: read name hash (see Section C.1.1), observed allele, and average base quality of the nucleotides in the observed allele in the read.

Next, each sample is analyzed independently (see Figure 4.1). First, ParascopyVC finds most probable pooled genotypes for all PSVs and variants and selects a set of PSVs with paralog-specific genotypes that are consistent with the sample paralog-specific copy number. Resulting set of PSVs can be used to pinpoint read pairs to a specific repeat copy or to a subset of repeat copies. Finally, ParascopyVC uses read-variant observations to find PSV and variant paralog-specific genotypes when possible.

4.3.1 Pooled and paralog-specific genotypes

In a duplicated region with n repeat copies, paralog-specific copy number of a sample s is a tuple $\left(c_{si}\right)_{i=1}^{n}$, $c_{si} \in \mathbb{N}_{\geq 0}$. Each element of the tuple c_{si} stands for the number of times the corresponding repeat copy i appears in the genome sequence of the sample s. Due to partial and full repeat copy deletions and duplications, copy number c_{si} can be both lower and higher than the reference paralog-specific copy number $c_i^{(\text{ref})} = 2$. Aggregate copy number represents the sum copy number across all repeat copies: $\hat{c}_s = \sum_{i=1}^{n} c_{si}$, while the reference aggregate copy number is $\hat{c}^{(\text{ref})} = 2n$.

Duplicated regions harbor both sequence variants and paralogous sequence variants (PSVs). We examine variants jointly across all repeat copies; as a result, both variants and PSVs are characterized by a set of n genomic positions (one in each duplicated copy) and an allele set A. In the reference genome, variant v exhibits allele $a_{vi}^{(\text{ref})}$ on i-th repeat copy of the duplication. Note, that it is possible that various repeat copies of a duplication lie on opposite strands; in such cases we consider reverse-complement sequences of 'minus'-strand repeat copies.

Pooled genotype $\hat{g}_{vs}$ of a variant v is a multiset of $\hat{c}_s$ alleles — such genotype collects variant alleles over all repeat copies without any specific order and can contain the same allele many times. A paralog-specific genotype g_{vs} is a tuple of n allele multisets, where the i-th multiset contains c_{si} alleles. As an example, consider a two-duplication and a sample s with aggregate copy number $\hat{c}_s = 5$ and paralog-specific copy number $c_s = (3, 2)$. One possible pooled genotype of a variant v with two alleles $A_v = \{0, 1\}$ would be $\hat{g}_{vs} = 0/0/0/1/1$ and one possible paralog-specific genotype would be $g_{vs} = (0/0/0, 1/1)$. Each paralog-specific genotype is associated with a single pooled genotype, obtained by combining all multisets g_{vsi}. Note that a pooled genotype can be associated with multiple paralog-specific genotypes.

We say that a paralog-specific genotype g_{vs} is *reference-compatible*, if, for all i, multiset g_{vsi} consists of the reference allele $a_{vi}^{(\text{ref})}$ taken c_{si} times. Consequently, we say that a pooled genotype $\hat{g}_{vs}$ is reference-compatible if it is associated with a reference-compatible paralog-specific genotype. Assuming that all repeat copies are present in a sample ($c_{si} > 0\ \forall i$), a reference-compatible pooled genotype would contains at least two different alleles in case of paralogous sequence variants (PSVs); and exactly one allele in case of variants that do not overlap PSVs.

It is possible that sample paralog-specific copy number is not fully known: for example in a three-copy duplication sample aggregate copy number was identified as $\hat{c}_s = 6$ and paralog-specific copy number as $c_s = (2, ?, ?)$. In such cases we virtually combine repeat copies with unknown paralog-specific copy numbers into a new, *extended* repeat copy. It is possible that several paralog-specific and several pooled genotypes are reference-compatible if a PSV v has different reference alleles within a single extended repeat copy.

4.3.2 Calculating pooled genotype probabilities

ParascopyVC starts by calculating pooled genotype probabilities $P(\hat{g}_{vs} \mid X)$ based on the allelic read depth X_{vs}. ParascopyVC uses multinomial distribution (MN) consistent with FreeBayes polyploid genotype likelihood calculation [40]. In order to improve genotyping accuracy, the method discards allele obsevations with base qualities less than 10 and all partial allele observations. As longer alleles are less likely to be completely covered by a read, we scale allele probabilities in the multinomial distribution by the factor $\overline{|r|} - |a_i| - 1$, where $\overline{|r|}$ is the average read length and $|a_i|$ is the length of the i-th allele. Probability of observing allelic read depth X_{vs} given the pooled genotype $\hat{g}_{vs}$ can be calculated in the following way:

$$P(X \mid \hat{g}_{vs}) = P_{\text{MN}}\big(X_{vs};\ \mathfrak{p}(a_1,\ \hat{g}_{vs}), \ldots, \mathfrak{p}(a_{|A_v|},\ \hat{g}_{vs})\big),$$

$$\text{where } \mathfrak{p}(a,\ \hat{g}) = \frac{\big(\overline{|r|} - |a| - 1\big) \cdot \max\{e \cdot \hat{c}_s,\ \mu(a \in \hat{g})\}}{\sum_{a' \in A_v} \big(\overline{|r|} - |a'| - 1\big) \cdot \max\{e \cdot \hat{c}_s,\ \mu(a' \in \hat{g})\}}$$

and where e is the error rate, $\hat{c}_s$ is the aggregate copy number of the sample s, A_v is the full set of alleles, and $\mu(z \in Z)$ is the number of occurences (multiplicity) of an element z in the multiset Z. Finally, we use Bayes' theorem to get the pooled genotype probability

$$P(\hat{g}_{vs} \mid X_{vs}) = \frac{p(\hat{g}_{vs}) \cdot P(X_{vs} \mid \hat{g}_{vs})}{\sum_{\hat{g}' \in \hat{G}_{vs}} p(\hat{g}') \cdot P(X_{vs} \mid \hat{g}')}.$$

Note that the set of pooled genotypes $\hat{G}_{vs}$ depends on the set of alleles A_v and aggregate copy number $\hat{c}_s$. Additionally, we select equal pooled genotype priors $p(\hat{g}) = 1 \,/\, |\hat{G}_{vs}|$.

To reduce the number of false positive variants, we measure the size of the strand bias effect on each allele [141]. To do that, we analyze 2×2 contingency table with the number of forward and reverse reads that support the allele in question against the forward and reverse reads that support any other allele. If Fisher's exact test [142] p-value is less than 0.01 for one of the alleles, we mark the variant as the potential false-positive.

4.3.3 Selecting informative paralogous sequence variants

Paralogous sequence variants (PSVs) are short differences between the sequences of the repeat copies in the reference genome. For each PSV v, Parascopy [140] defines a vector f_v, where f_{vi} is the population frequency of the reference allele $a_{vi}^{(\text{ref})}$ on the repeat copy i. Additionally, Parascopy names a PSV v *reliable* if all its f-values are over 0.95. We say that a PSV is *informative*, if its *reference-compatible* paralog-specific genotypes have sufficiently high probabilities. In such cases a read containing an allele a is expected to originate on one of the repeat copies that harbor the allele a. Additionally, we discard a PSV if one of its alleles appears in the reference sequences of all extended repeat copies.

Due to high frequencies of the reference alleles f_v, most reliable PSVs are informative. Nevertheless, some unreliable PSVs may be informative too: suppose a PSV v in a two-copy duplication has alleles $a_{v1}^{(\text{ref})}$ = '0' and $a_{v2}^{(\text{ref})}$ = '1' and the reference allele frequencies are $f_{v1} = 0.99$ and $f_{v2} = 0.60$. Suppose the sample s is estimated to have paralog-specific copy number $c_s = (2, 2)$ and pooled PSV genotype $\hat{g}_{vs} = 0/0/1/1$. Then, based on PSV f-values, it is possible to calculate various paralog-specific genotype likelihoods, which would show that the paralog-specific genotype $(0/0, 1/1)$ is much more likely than other paralog-specific genotypes, for example $(0/1, 0/1)$. Therefore, in this example, PSV v is *informative* and can be used to differentiate between repeat copies as the most probable paralog-specific genotype is reference-compatible.

We calculate PSV paralog-specific genotype probabilities based on based on PSV f-values and pooled genotype probabilities $P(\hat{g}_{vs} \mid X_{vs})$ in the following way:

$$P(g_{vs} \mid X_{vs}; f_v) = \frac{P(\hat{g}_{vs} \mid X_{vs}) \cdot P(g_{vs};\ f_v)}{\sum_{\hat{g}' \in \hat{G}_{vs}} P(\hat{g}' \mid X_{vs}) \cdot \sum_{g' \in G(\hat{g}')} P(g';\ f_v)}.$$

where $\hat{G}_{vs}$ is the full set of pooled genotypes and $G(\hat{g})$ is the full set of paralog-specific genotypes associated with the pooled genotype $\hat{g}$. Finally, we calculate the probability of the paralog-specific genotype as

$$P(g_{vs};\ f_v) = \prod_{i=1}^{n} f_{vi}^{\mu\left(a_{vi}^{(\mathrm{ref})} \in g_{vsi}\right)} \times (1 - f_{vi})^{c_{si} - \mu\left(a_{vi}^{(\mathrm{ref})} \in g_{vsi}\right)}.$$

In other words, probability of the paralog-specific genotype g is a product of either f_{vi} or $(1 - f_{vi})$ depending on the match between the genotype alleles and the reference alleles of the PSV v. Finally, we say that the PSV v is informative if the sum probability of all reference-compatible paralog-specific genotypes exceeds 99%.

4.3.4 Discarding conflicting PSVs

In rare cases, a PSV is predicted to have a reference-compatible genotype and is deemed *informative*, but in reality possesses a reference-incompatible genotype. For example, a PSV with relatively high frequencies f_v and the pooled genotype $\hat{g}_{vs} = 0/0/1/1$ would be predicted a paralog-specific genotype $g_{vs} = (0/0, 1/1)$, however, the true genotype may be $(1/1, 0/0)$.

To find and to discard such PSVs, we search for pairs of closeby PSVs that are covered by the same read pair with conflicting read-allele observations. To reduce the running time, we only consider a pair of PSVs if there are less than 10 other informative PSVs between them. For each pair of PSVs we run a one-tailed Binomial test, where the number of trials is the total number of read pairs that cover both PSVs, a success represents a read pair with conflicting allele observations and the probability of a success is $2e - e^2$, where e denotes the error rate. We say that the two PSVs are in conflict, if the test p-value falls below 10^{-3}.

We construct an undirected graph of conflicts, where each node represent an informative PSV, and two PSVs are connected if and only if they are in conflict. Next, we aim to keep a subset of PSVs V such that the remaining graph contains no edges and $\sum_{v \in V} \tilde{f}_v$ is maximal, where $\tilde{f}_v$ is the minimal frequency f of the PSV v across all repeat copies. Unfortunately, this problem definition is equivalent to the weighted maximum clique problem, which is NP-complete [143]. Because of that, we employ the following greedy heuristic: we iteratively remove the PSV v with the maximal remaining number of edges multiplied by $\sqrt{1 - \tilde{f}_v}$, until the graph is edgeless.

4.3.5 Finding likely paralog-specific genotypes for single nucleotide variants

In contrast to paralogous sequence variants, population frequencies of various variant alleles are unknown. Therefore, we utilize read pairs that overlap both a variant and one or more informative PSVs to identify the likelihood of various variant paralog-specific genotypes. For each read pair we estimate possible read location and probabilities (see Section C.1.2 for details). First, we say that an individual read mate is mapped uniquely if its original mapping quality is over 50 and if it overlaps non-duplicated region in the genome by at least 15 bp. Second, for all individual read mates that are not mapped uniquely, we observe all read–informative PSV interactions and calculate the probabilities $p_r(i)$ for various possible read r locations. For each informative PSV, probabilities of the read locations that are inconsistent with the observed allele are multiplied by the error rate e ($e = 0.01$ by default), while probabilities of all consistent locations are multiplied by $1 - e$. On the next step, read mate r' location probabilities are taken into account: $p_r(i) \leftarrow p_r(i) \cdot \max\{p_{r'}(i),\ 10^{-5}\}$; paired read location probabilities are multiplied by each other, while non-matching paired read locations are penalized. Finally, all location probabilities are normalized by the paralog-specific copy number and other location probabilities:

$$p_r(i) \leftarrow \frac{c_{si} \cdot p_r(i)}{\sum_{j=1}^{n} c_{sj} \cdot p_r(j)}.$$

For each potential variant, called using FreeBayes [40], we calculate its pooled genotype probabilities using allelic read depth according to the multinomial distribution. Similarly to PSVs, we discard allele observations with low base qualities (< 10). In order to reduce computational complexity, we discard all unlikely pooled genotypes (probability $< 10^{-5}$), and store the sum probability of the discarded genotypes, which will be used later to calculate pooled and paralog-specific genotype qualities. Additionally, we do not use the probabilties of the remaining pooled genotypes in the subsequent calculations, as it would break the independence of the paralog-specific genotype likelihood calculation.

Next, for each variant and all applicable paralog-specific genotypes, we calculate paralog-specific genotype priors $p(g_{vs})$ in the following way: each repeat copy genotype g_{vsi} is penalized by the mutation rate (10^{-3}) if non-reference allele is present (homozygous and heterozygous genotypes are penalized equally). For example, possible pooled genotypes $0/0/0/0$ and $0/0/1/1$ produce 4 possible paralog-specific genotypes: $(0/0, 0/0)$, $(0/0, 1/1)$, $(1/1, 0/0)$ and $(0/1, 0/1)$ with the priors approximately equal to $1,\ 10^{-3},\ 10^{-3}$ and 10^{-6}, respectively. Paralog-specific genotype priors for variants that overlap PSVs are calculated according to the PSV f-values as described above: $p(g_{vs}) = P(g_{vs}; f_v)$.

Finally, paralog-specific genotype probabilities are updated according to the read location probabilities and read–variant allelic observations. Assuming that the read r supports variant allele a_r and has probability $p_r(i)$ to originate on the repeat copy i, probability of the paralog-specific genotype can be described in the following way:

$$P(g_{vs} \mid r) = p(g_{vs}) \sum_{i=1}^{n} \frac{p_r(i)}{c_{si}} \Big((1-e) \cdot \mu\left(a_r \in g_{vsi}\right) + e \cdot \left[c_{si} - \mu\left(a_r \in g_{vsi}\right) \right] \Big).$$

Finally, the full paralog-specific genotype probabilities are calculated across all reads R_{vs} that overlap the variant v in the sample s:

$$P(g_{vs} \mid R_{vs}) = \frac{\prod_{r \in R_{vs}} P(g_{vs} \mid r)}{\sum_{g'} \prod_{r \in R_{vs}} P(g' \mid r)}.$$

4.3.6 Output files and quality scores

ParascopyVC generates two output variant call format (VCF) files: pooled and paralog-specific. In a pooled VCF file, reference allele of a variant or PSV is the corresponding reference sequence in the first repeat copy of the duplication. In case of PSVs, the reference alleles from all other repeat copies are stored as alternative alleles. For each sample, ParascopyVC provides the most probable pooled genotype $\hat{g}_{vs}$ and its Phred quality score, calculated as $Q(\hat{g}_{vs}) = -10 \cdot \log_{10}\left(1 - P(\hat{g}_{vs})\right)$.

In a paralog-specifc VCF file, ParascopyVC outputs variants once for each repeat copy. For each sample and each repeat copy i, ParascopyVC finds the most probable marginal paralog-specific genotype $g_{vs}^{(i)}$, where $P(g_{vs}^{(i)}) = \sum_{g' \text{ s.t. } g'_i = g_{vs}^{(i)}} P(g')$. Phred quality scores for the marginal paralog-specific genotypes are calculated in a similar manner to the pooled genotype qualities: $Q(g_{vs}^{(i)}) = -10 \cdot \log_{10}\left(1 - P(g_{vs}^{(i)})\right)$.

Each variant in a variant call VCF is characterized by its variant quality, which is a different metric compared to the genotype qualities. Traditionally, variant quality encodes a probability that the variant genotype contains a non-reference allele for at least one of the samples. In a pooled output VCF file, ParascopyVC sets PSV qualities to a high constant value based on the following two cases: (i) if a pooled PSV genotype is *reference-compatible*, then it contains a non-reference allele compared to the first repeat copy; (ii) if a pooled PSV genotype is not reference-compatible, then it contains a non-reference allele in one of the marginal paralog-specific genotypes. For all variants that do not overlap PSVs, ParascopyVC uses underlying FreeBayes quality scores as the variant qualities in the pooled VCF file.

In the paralog-specific output VCF file, ParascopyVC uses the same formula to calculate variant qualities for both variants and PSVs: $Q^{(i)}(v) = -10 \sum_{s \in S} \log_{10} P\left(g_{vs}^{(i)\ (\text{ref})}\right)$, where $g_{vs}^{(i)\ (\text{ref})}$ is the reference paralog-specific genotype on the repeat copy i (consisting entirely of the reference allele $a_{vi}^{(\text{ref})}$). In other words, ParascopyVC calculates the Phred quality score of the probability that all samples exhibit the reference paralog-specific genotype on the repeat copy i.

4.3.7 Simulated WGS datasets

To assess the accuracy of the ParascopyVC variant calling, we generated two artificial variant sets, with and without unreliable PSVs, which we will call SIM-R and SIM-U, respectively. In case of the SIM-R variant set, we simulated a diploid genome by adding artificial sequence variants such that each genomic coordinate is mutated with probability 0.001 (approximately one variant per kilobase). Rates of different variant types (80% substitutions, 10% insertions and 10% deletions; 61.5% heterozygous variants) were selected to be similar to the variant rates in the Genome in a Bottle (GIAB) high-confidence variant calls for the HG002 individual (85.5% substitutions, 7.25% insertions and 7.25% deletions; 61.5% heterozyous variants) [76, 87]. All artificial variants overlapping PSVs were discarded. The final baseline variant set SIM-R contains $2.74 \cdot 10^6$ sequence variants on the chromosomes 1–22.

In case of the SIM-U variant set, we generated heterozygous and homozygous variants overlapping PSVs according to the frequencies of the reference PSV alleles (f_v) in the 503 European ancestry samples from the 1000 Genomes Project (1kGP) [2], calculated by Parascopy `v1.7` [140]. Next, we combined the resulting variant set with the SIM-R variant set to obtain a bigger set of $2.75 \cdot 10^6$ variants.

For each simulated variant set, we generated four artificial sequencing datasets using ART Illumina [144] read simulator tool `v2016-06-05`. As a default sequencing dataset, we generated 150 bp reads modeling Illumina HiSeq 2500 protocol at 30× read depth. Paired-end simulation was performed with 500 bp mean fragment size and 20 bp standard deviation (`-l 150 -ss HS25 -f 30 -m 500 -s 20`). Additionally, we generated three sequencing datasets, each altering one sequencing feature compared to the default sequencing dataset: (i) 60× read depth instead of 30×; (ii) 100 bp reads instead of 150 bp; (iii) mate-pair simulation with 2500 bp mean insert size and 50 bp standard deviation (`-mp -m 2500 -s 50`). Finally, we mapped artificial reads to the reference genome using BWA-MEM `v0.7.17` [31, 32].

4.3.8 Evaluating paralog-specific variant calling

In addition to eight simulated datasets (four sequencing datasets for two variant sets SIM-R & SIM-U), we utilize seven sets of high-confidence variant calls (HG001–HG007) constructed by the GIAB Consortium [76, 87]. For each dataset, we mapped the corresponding Illumina reads to the GRCh38 reference genome using BWA-MEM. As WGS datasets HG005, HG006 and HG007 have very high coverage ($\geq 100\times$), these datasets were subsampled by randomly selecting read pairs with probabilities 0.1, 0.3 and 0.3, respectively. Subsampling was performed using Samtools `v1.14` [145].

For each of the 15 benchmark datasets, we calculated aggregate and paralog-specific copy number profiles using Parascopy `v1.7` [140] and PSV reference allele frequencies f_v: (i) equal to 0.9999 for the simulated dataset SIM-R; (ii) obtained from the 503 European ancestry samples from 1kGP for the datasets HG001–HG004 and SIM-U; (iii) and obtained from the 504 East-Asian ancestry samples from 1kGP for the datasets HG005–HG007. Based on these copy number profiles, we called pooled and paralog-specific variants using ParascopyVC `v1.8`.

We limited copy number analysis and variant calling to 167 low-copy repeat loci, selected in Prodanov & Bansal (2022) [140].

As ParascopyVC analysis is limited to the duplicated loci and to the loci with known aggregate or paralog-specific copy number values, ParascopyVC provides a set of paralog-specific or pooled regions, where the corresponding variants are called. In case of the simulated WGS datasets, we used these all paralog-specific genomic regions on chromosomes 1–22, where reference copy number ranges between 4 and 8. On the other hand, each GIAB WGS benchmark variant call set is limited to a set of high-confidence benchmarking regions. Consequently, we overlapped the high-confidence regions with the ParascopyVC paralog-specific regions and selected regions where aggregate copy number matches the reference copy number and is bounded between 4 and 8. Additionally, as all benchmark variant calls are diploid, we discard all subregions with paralog-specific copy number other than two. In case of the GIAB WGS datasets, we discarded Parascopy paralog-specific regions with non-`PASS` filters, as they do not provide confident copy number estimates.

Finally, we calculated precision and recall using RTG tools `v3.12.1` [78, 79] for 15 datasets and three variant calling tools: FreeBayes `v1.3.5` [40], GATK HaplotypeCaller `v4.2.2` [41] and ParascopyVC `v1.7.8`.

4.3.9 Evaluating pooled variant calling

In contrast to FreeBayes and GATK, ParascopyVC calculates pooled variant genotypes in addition to paralog-specific variant genotypes. In order to obtain ground truth pooled variant calls, we merged benchmark variant calls appearing on various repeat copies. If one of the repeat copies is not covered by the high-confidence benchmarking regions, we remove the pooled region from the observation. Similarly to paralog-specific regions, we select a set of pooled regions that overlap the high-confidence benchmarking regions for the dataset and where aggregate copy number matches the reference copy number. Finally, we evaluate

ParascopyVC accuracy separately for different aggregate copy number values (4, 6 and 8) using RTG tools.

4.4 Acknowledgements

Chapter 4, in part, is currently being prepared for submission for publication of the paper by Timofey Prodanov and Vikas Bansal. The dissertation author was the primary author of this paper.

appendix A

Supplementary Material for Chapter 2

A.1 Supplementary Figures and Tables

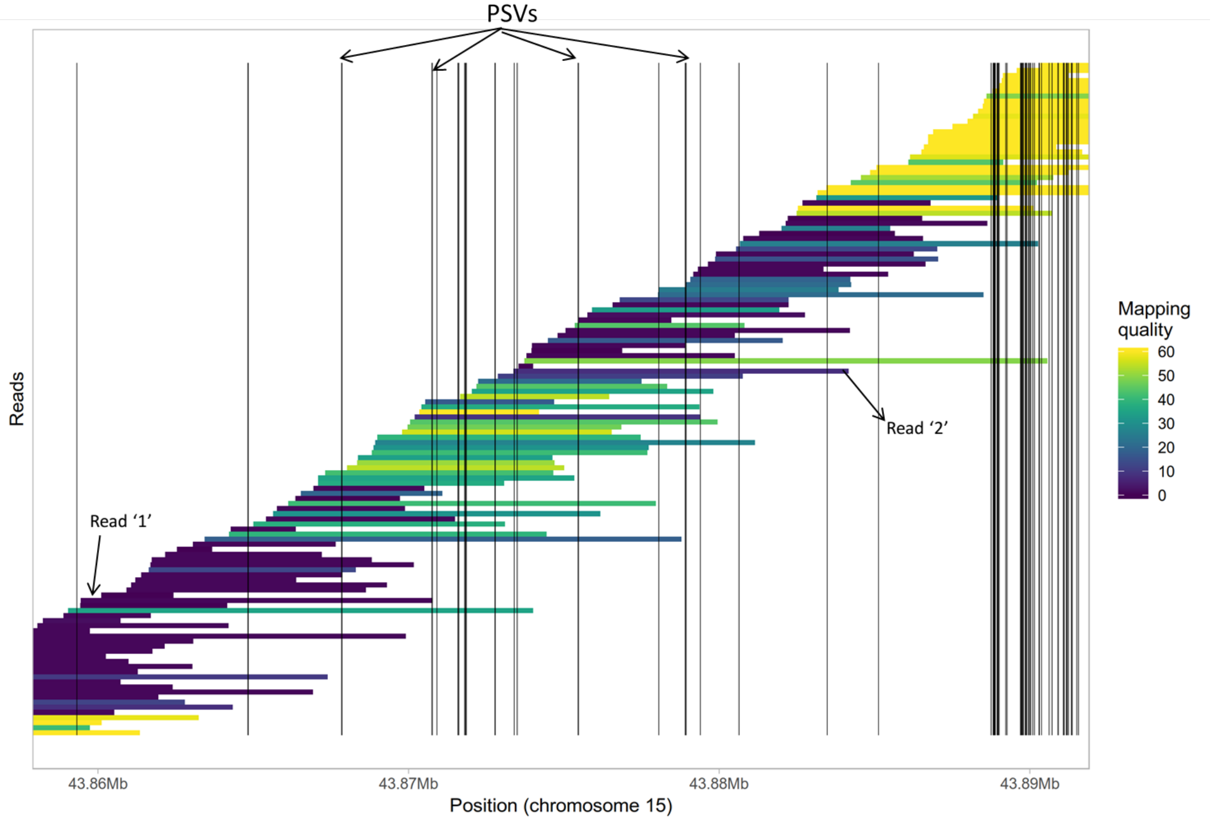


Figure A.1. **Illustration of sub-optimal mapping of long reads in segmental duplications.** Reads mapped using the Minimap2 aligner to a 35 kb region from a segmental duplication on chromosome 15 (covering the *STRC* gene) are shown. Reads are shown as horizontal bars (color-coded by mapping quality) while PSVs are shown as vertical lines. Several reads overlap multiple PSVs (e.g. read '2' overlaps 6 PSVs) but are still assigned low mapping quality. Other reads overlap no PSVs (e.g. read '1') and hence cannot be mapped uniquely.

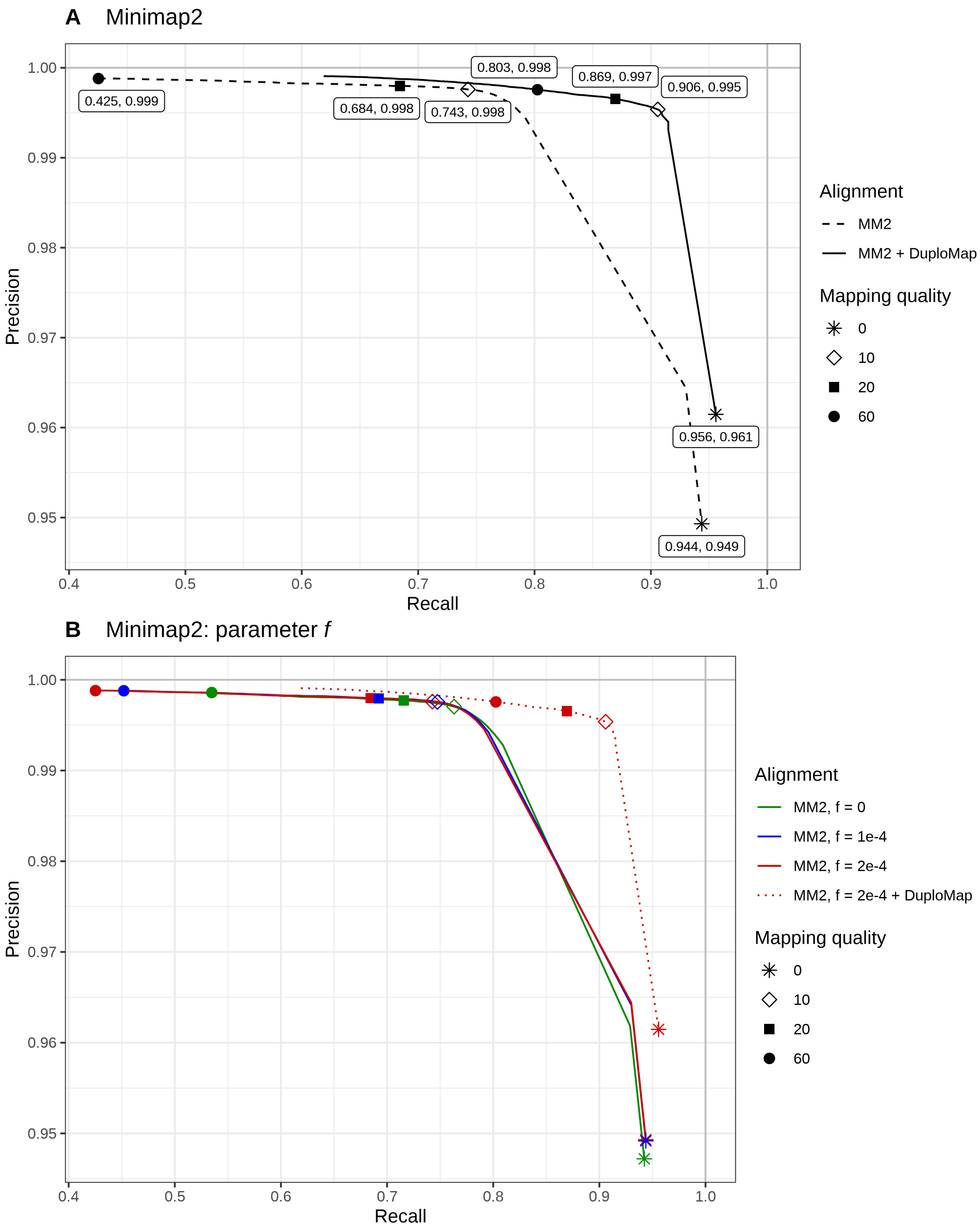


Figure A.2. **Accuracy of read mapping using Minimap2 and Minimap2+DuploMap on simulated long read data in segmental duplications.** Reads of median length 8.5 kb were used for simulations. (a) Accuracy of Minimap2 and Minimap2 + DuploMap alignments. (b) MM2 accuracy with different values of parameter f (discarding top f of the repetitive minimizers, $2 \cdot 10^{-4}$ by default).

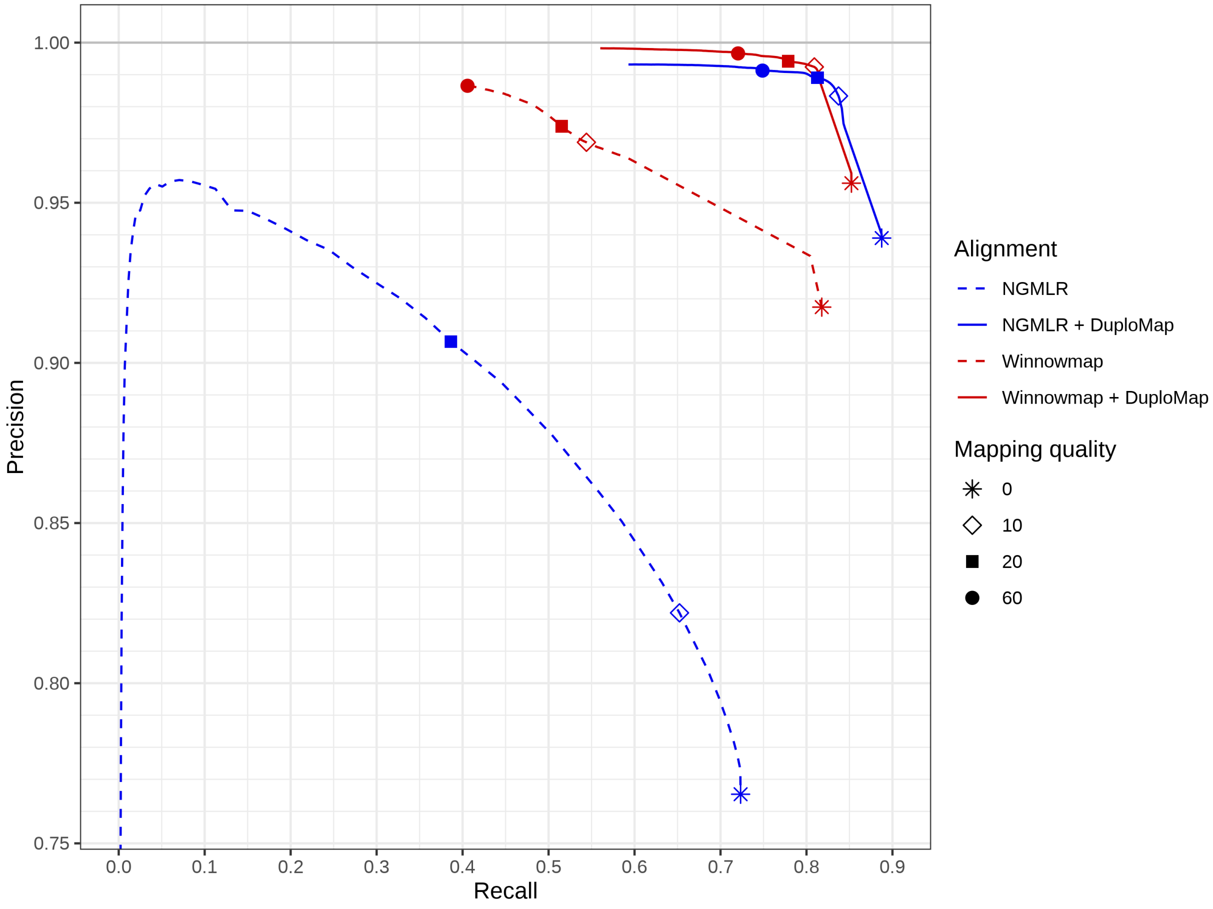


Figure A.3. **Improvement in accuracy of read mapping in segmental duplications using DuploMap in combination with the NGMLR and Winnowmap alignment tools.** Each precision-recall curve is plotted using different mapping quality thresholds.

Table A.1. **Running time and memory usage of long read alignment tools and DuploMap on simulated SMS reads.** Running time shows the elapsed real time for each aligner using 8 cores. Mapping speed shows the average number of reads analyzed per second (by a single core). Note that DuploMap analyses only a subset of reads that intersect segmental duplications. Only the subset of reads that intersect segmental duplications were mapped using BLASR and Minimap2 with $f = 0$ due to their long running time. All tools were run on a CentOS 6.6 system with Intel Xeon CPU E5-2670 @ 2.60 GHz, with jobs managed by a Torque/PBS system.

Sequencing technology	Length (kb)	Aligner	Running time (hh:mm)	Mapping speed (reads / second)	Memory usage (Gb)
PacBio CLR	8.5	Minimap2	3:54	105.1	13.33
PacBio CLR	8.5	NGMLR	35:18	11.2	10.55
PacBio CLR	8.5	BLASR	-	3.6	29.29
PacBio CLR	8.5	Minimap2, $f = 0$	-	15.6	67.68
PacBio CLR	8.5	DuploMap	1:39	10.7	20.16
PacBio CLR	20	Minimap2	3:57	44.9	15.11
PacBio CLR	20	DuploMap	1:27	4.9	17.58
PacBio CLR	50	Minimap2	2:57	19.4	15.38
PacBio CLR	50	DuploMap	1:30	2.1	16.95
ONT	8.4	Minimap2	4:37	91.4	10.90
ONT	8.4	DuploMap	1:24	11.3	15.38

Table A.2. **Running time and memory usage of DuploMap on real data.** Running time represents elapsed real time using 8 cores

Genome	Sequencing technology	Median coverage	Running time (hh:mm)	Memory usage (Gb)
HG001	ONT	36	4:57	38.29
HG001	PacBio CCS	29	4:09	29.54
HG002	PacBio CLR	45	8:31	61.88
HG002	PacBio CCS	29	6:29	25.79
HG002	ONT	58	14:28	55.91
HG003	PacBio CLR	20	3:54	32.35
HG004	PacBio CLR	19	3:24	29.13
HG005	PacBio CCS	32	5:51	30.58

Table A.3. **Simulations with non-reference PSVs.** Two-copy segmental duplications in the human genome (hg19) were used for assessing the impact of unreliable PSVs (non-reference) on the accuracy of DuploMap. Of the 52,276 high-complexity PSVs in two-copy segmental duplications, we modified the genome sequence for one of the two copies for 0, 15 and 30% of the PSVs. Reads were simulated using the modified genome and mapped using DuploMap. The percentage of PSV positions (total count = 104,552) with high quality genotypes (all filters pass and quality score ≥ 60) is shown in column 2. The precision and recall for reads mapped with mapping quality ≥ 30 is also shown.

Non-ref PSVs (%)	PSV positions genotyped (%)	Incorrect genotypes (%)	Precision (%)	Recall (%)
0	86.6	0.000	99.91	86.85
15	85.0	0.001	99.90	86.54
30	81.3	0.000	99.91	86.17

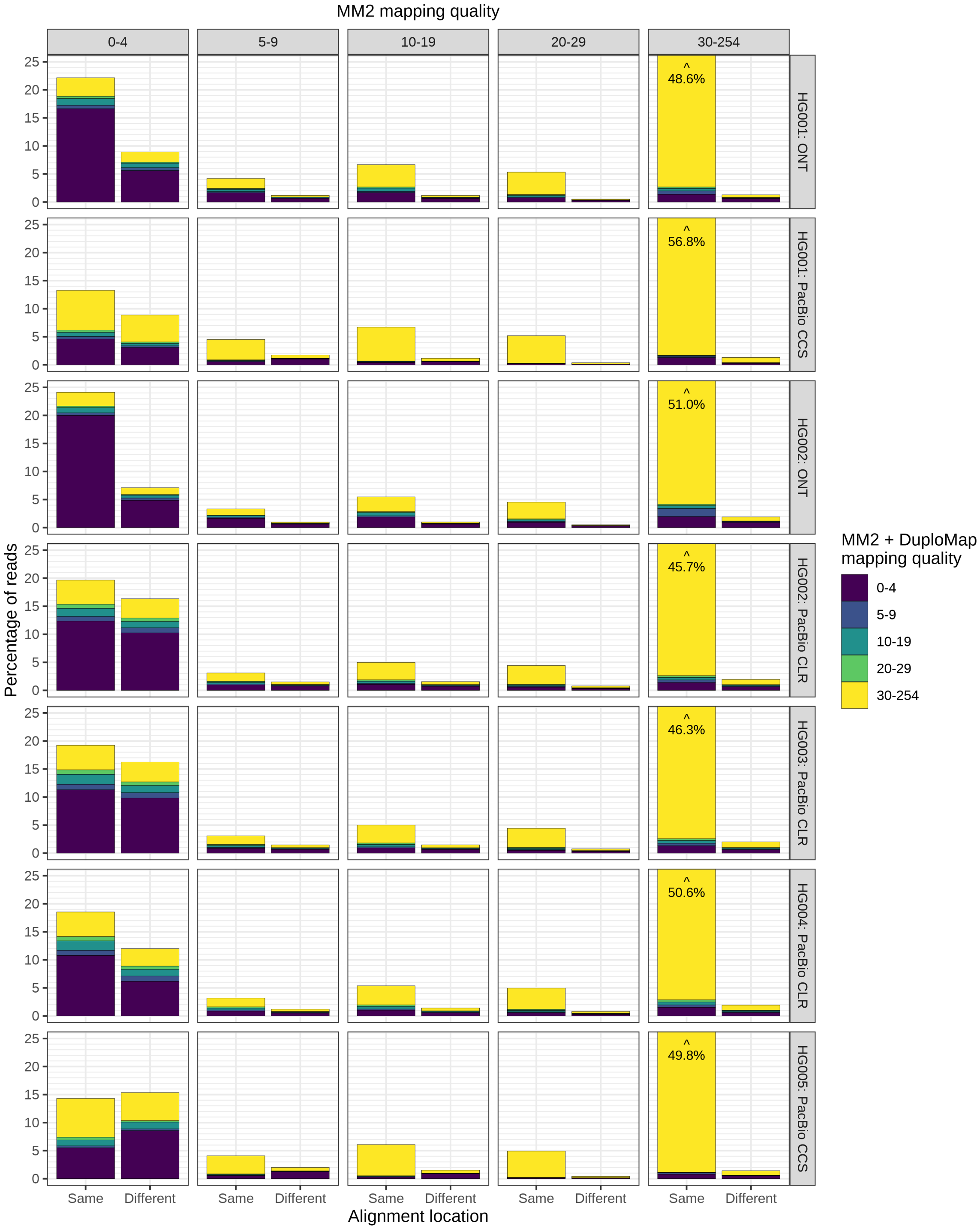


Figure A.4. **Comparison of mapping qualities and alignment locations for reads aligned with Minimap2 and Minimap2 + DuploMap on multiple long-read datasets.** Column contain reads with corresponding mapping quality in the MM2 alignments. Two bars in each subplot represent reads that have same or different alignments in MM2 and MM2 + DuploMap. Bar height represents percentage of reads in the corresponding category out of all analyzed reads in the dataset, and color shows alignment mapping quality after MM2 + DuploMap. Some bars are clipped, in that cases total bar height is shown at the top of the bar.

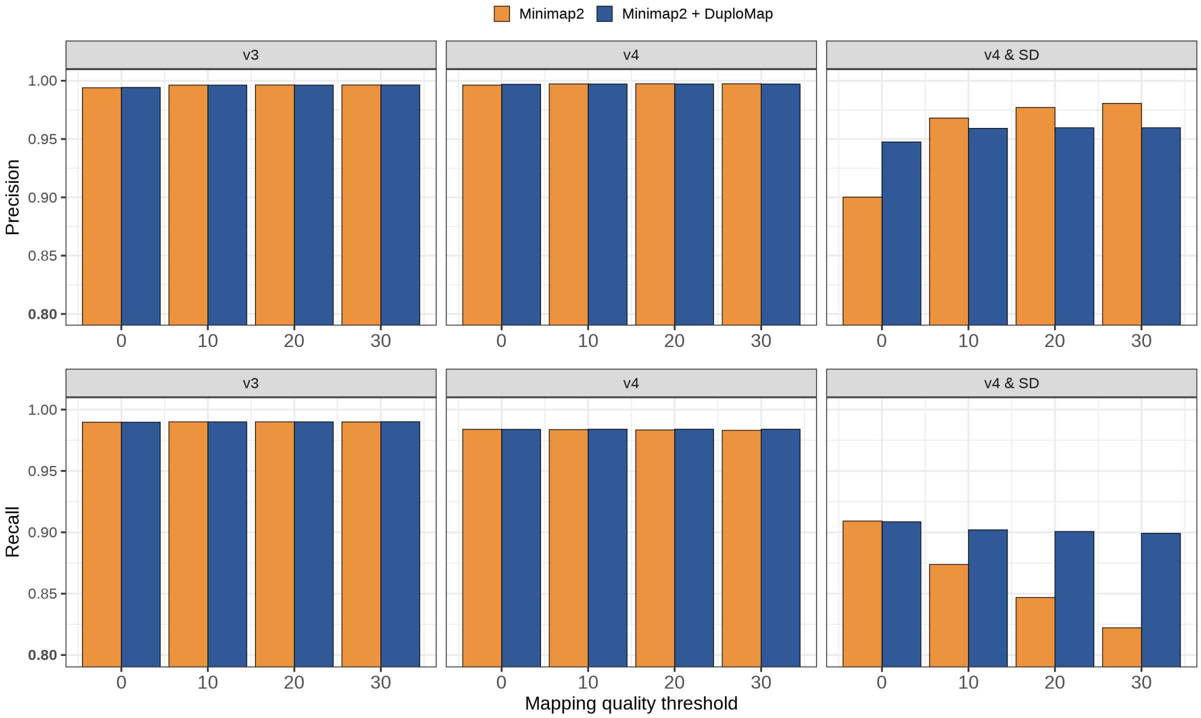


Figure A.5. **Precision and recall of variant calling in segmental duplications using simulated reads aligned with Minimap2 and Minimap2 + DuploMap.** Three columns show different subsets of variants: within all *Long-SegDups* regions; within *Long-SegDups* regions with sequence similarity between 99.0% and 99.9%; and within *Long-SegDups* regions with sequence similarity between 99.9% and 100%.

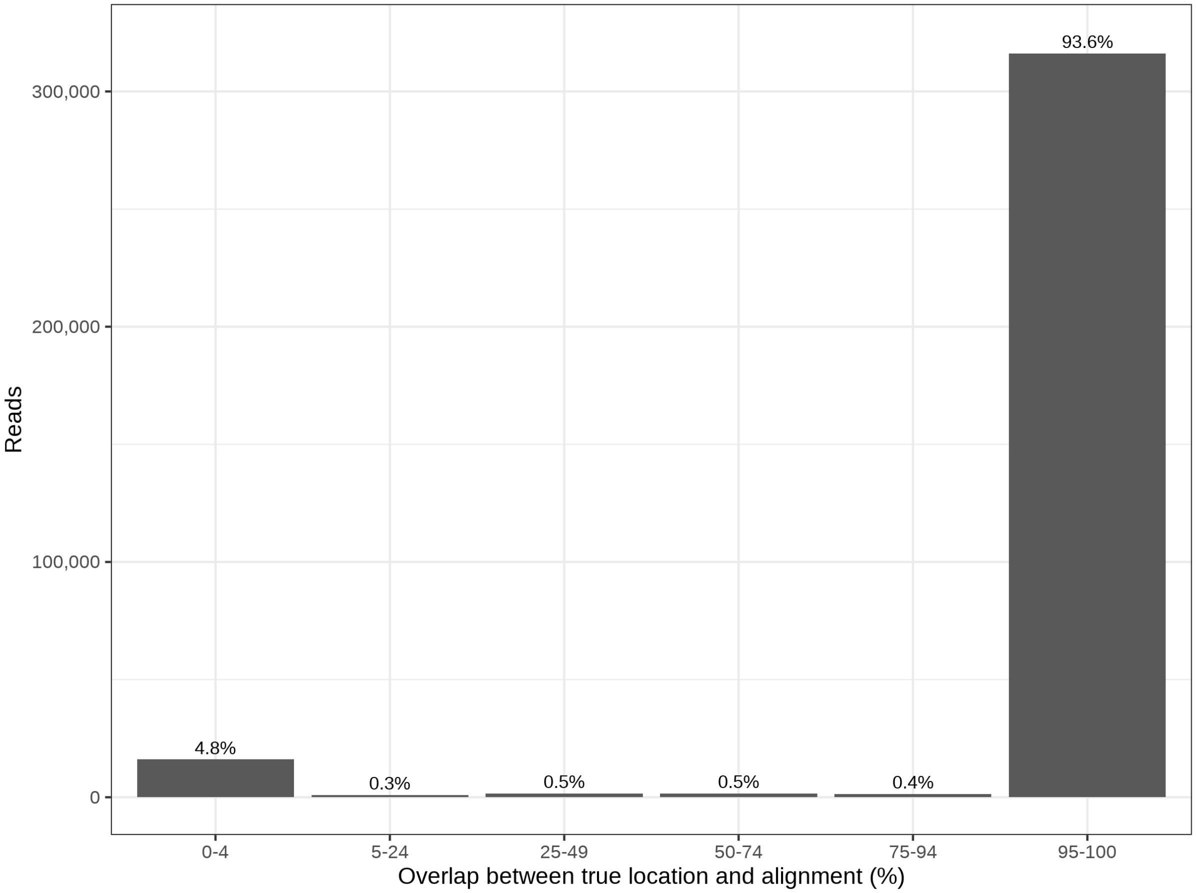


Figure A.6. **Overlap percentage between true location and alignment locations in Minimap2 mapping of long simulated reads.**

Table A.4. **Comparison of variant calling accuracy for HG002 CCS reads aligned with Minimap2 (MM2) and DuploMap.** SNVs were called using Longshot (mapping quality threshold of 10). ‘v3’ refers to the high-confidence regions of the genome in the GIAB v3.3.2 call set for each genome and ‘v4’ refers to the expanded high-confidence regions in the GIAB v4.1 callset for HG002. ‘SD’ or *Long-SegDups* refers to the genomic regions in which reads were realigned using DuploMap.

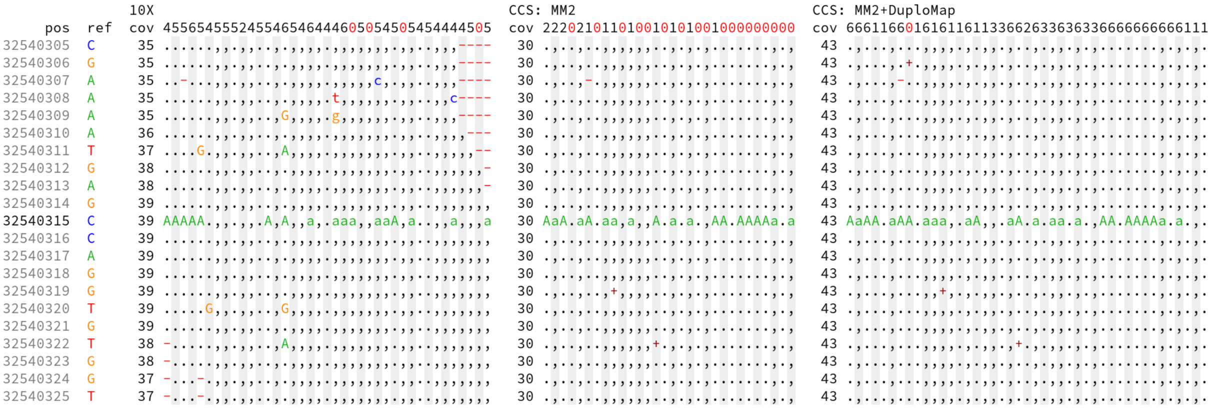
Alignment method	Genome subset	Variant quality	Number of variants	Precision	Recall	F1
MM2	v3	30	3,010,414	0.9963	0.9900	0.9931
MM2 + DuploMap	v3	30	3,010,534	0.9963	0.9900	0.9932
MM2	v4	30	3,319,220	0.9973	0.9837	0.9904
MM2 + DuploMap	v4	30	3,320,654	0.9972	0.9840	0.9905
MM2	v4 ∩ SD	30	36,044	0.9680	0.8738	0.9185
MM2 + DuploMap	v4 ∩ SD	30	37,548	0.9592	0.9020	0.9297
MM2	v4 ∩ SD	60	35,902	0.9701	0.8723	0.9186
MM2 + DuploMap	v4 ∩ SD	60	37,421	0.9611	0.9007	0.9299
MM2	v4 ∩ SD	90	35,582	0.9727	0.8668	0.9167
MM2 + DuploMap	v4 ∩ SD	90	37,192	0.9634	0.8974	0.9292

Table A.5. **Mappability of 17 disease-associated genes with Minimap2 and Minimap2 + DuploMap for HG002 PacBio CCS dataset.** MM2 coverage columns show percentage of bases covered by at least 15 reads (half the median coverage) with high mapping quality (≥ 10 and ≥ 20) in Minimap2 alignments in all exons of the corresponding gene. Last two columns show difference between percentage of covered bases in *Minimap2 + DuploMap* and *Minimap2* alignments.

Gene	Chromosome	Sum exon length	MM2 coverage (%) MQ ≥ 10	MM2 coverage (%) MQ ≥ 20	Δ MM2 + D coverage (%) MQ ≥ 10	Δ MM2 + D coverage (%) MQ ≥ 20
NAIP	5	7,704	20.6	9.7	+79.4	+90.3
C4B	6	5,427	36.8	28.6	+63.2	+71.4
SMN1	5	2,234	0.0	0.0	+59.1	+59.1
GTF2I	7	5,889	55.0	52.6	+45.0	+47.4
C4A	6	5,427	57.9	53.6	+42.1	+46.4
GTF2IRD2	7	5,394	48.3	22.0	+18.7	+45.0
PPIP5K1	15	6,575	90.3	81.6	+9.7	+18.4
CATSPER2	15	4,538	95.2	95.2	+4.8	+4.8
PDPK1	16	8,106	95.3	93.7	+4.7	+6.3
SMN2	5	2,671	62.9	57.1	+4.5	+10.3
NEB	2	26,310	99.5	98.0	+0.5	+2.0
OTOA	16	4,180	100.0	96.3	+0.0	+3.7
CFC1	2	1,669	100.0	0.0	+0.0	+100.0
OCLN	5	6,549	100.0	94.1	+0.0	+5.9
PMS2	7	5,150	97.9	85.4	+0.0	+12.5
NCF1	7	2,022	100.0	93.0	+0.0	+7.0
CR1	1	9,953	86.4	85.4	-1.0	+0.0

```
                 10X                                    CCS: MM2          CCS: MM2+DuploMap
      pos  ref   cov 662266666662554565666236454264533  cov 000000000000  cov 666666666666666665
120245932  T      31 ..,,..,,,.,,,.,,,.,,,,,.....,.,--   12 ,.,,,..,,...   18 ,.,..,,,,,,..,,...
120245933  G      33 ..,,..,,,.,,,.,,,.,,,,,.....,.,.,   12 ,.,,,..,,...   18 ,.,..,,,,,,..,,...
120245934  A      33 ..,,..,,,.,,,.,,,.,,,,,C....,.,.,   12 ,.,,,..,,...   18 ,.,..,,,,,,..,,...
120245935  A      33 ..,,..,,,.,,,.,,,.,,,,,.....,.,.,   12 ,.,,,..,,...   18 ,.,..,,,,,,..,,...
120245936  G      33 ..,,..,,,.,,,.,,,.,,,,,.....,.,.,   12 ,.,,,..+,...   18 ,.,..,,,,,,..+,...
120245937  A      33 ..,,..,,,.,,,.,,,.,,,,,G....,.,.,   12 ,.,,,..,,...   18 ,.,..,,,,,,..,,...
120245938  A      33 ..,,..,,,.,,,.,,,.,,,,,.....,.g.,   12 ,.,,,..,,...   18 ,.,..,,,,,,..,,...
120245939  A      33 ..,,..,,,.,,,.,,,.,,,,,.....,.,.,   12 ,.,,,..,,...   18 ,.,..,,,,,,..,,...
120245940  C      33 ..,,..,,,.,,,.,,,.,,,,,.....,.,.,   12 ,.,,,..,,...   18 ,.,..,,,,,,..,,...
120245941  A      33 ..,,..,,,.,,,.,,,.,,,,,.....,.,.,   12 ,.,,,..,,...   18 ,.,..,,,,,,..,,...
120245942  T      33 AAaaAAaaaAaaaAaaaAaaaaa.AAAAaAaAg   12 aAaaaAAaaAAA   18 aAaAAaaaaaaAAaaAAA
120245943  T      31 ..--..,,,.,,,.,,,.,,,,,.....,.,.,   12 ,.,,,..,,...   18 ,.,..,,,,,,..,,...
120245944  T      31 ..--..,,,.,,,.,,,.,,,,,.....,.,.,   12 ,.,,,..,,...   18 ,.,..,,,,,,..,,...
120245945  T      31 ..--..,,,.,,,.,,,.,,,,,.....,.,.,   12 ,.,,,..,,...   18 ,.,..,,,,,,..,,...
120245946  C      31 ..--..,,,.,,,.,,,.,,,,,.....,.,.,   12 ,.,,,..+,...   18 ,.,..,,,,,,..+,...
120245947  A      30 ..--..,,,.,,,.,,,.,,,,,-....,.,.,   12 ,.,,,..,,...   18 ,.,..,,,,,,..,,...
120245948  C      30 ..--..,,,.,,,.,,,.,,,,,-....,.,.,   12 ,.,,,..,,...   18 ,.,..,,,,,,..,,...
120245949  C      30 ..--..,,,.,,,.,,,.,,,,,-....,.,.,   12 ,.,,,..,,...   18 ,.,..,,,,,,..,,...
120245950  C      30 ..--..,,,.,,,.,,,.,,,,,-....,.,.,   12 ,.,,,..,,...   18 ,.,..,,,,,,..,,...
120245951  T      29 -.--..,,,.,,,.,,,.,,,,,-....,.,.,   12 ,.,,,..,,...   18 ,.,..,,,,,,..,,...
120245952  T      29 -.--..,,,.,,,.,,,.,,,,,-....,.,.,   12 ,.,,,..,,...   18 ,.,..,,,,,,..,,...
```

Figure A.7. **Example of a variant outside the GIAB high-confidence region identified using Minimap2 + DuploMap alignments of PacBio CCS reads.** Pileups of 10X Genomics linked-reads and PacBio CCS reads for the individual HG002 (aligned with Minimap2 and Minimap2 + DuploMap) in a window around the position chr1:120,245,942 (hg38) are shown. Each row shows a single position, and each column represents a single read. First digit of mapping quality is shown on top (0-6) and is highlighted in red for reads with mapping quality less than 10. The variant lies within 333 kb duplication with sequence similarity 99.2%. The variant is present in the GIAB and 10X Genomics calls with genotype equal to 1/1. However, all CCS reads mapped using Minimap2 have low MAPQ and hence no variant is called. Minimap2 + DuploMap alignments have high mapping quality at this locus enabling Longshot to identify the variant with the correct genotype.



```
                    10X                                         CCS: MM2                           CCS: MM2+DuploMap
      pos   ref    cov 455654555245546546444605054505454444505  cov 222021011010010101001000000000  cov 6661166016161161133662633636336666666666111
 32540305   C       35 ......,,.,,..,.,,,,.,,,,,,,.,,..,,,----   30 .,..,..,,,,,,..,.,.,.......,.,   43 .,...,...,,,,.,.,,.,..,.,,.,.,.......,.,.,.
 32540306   G       35 ......,,.,,..,.,,,,.,,,,,,,.,,..,,,----   30 .,..,..,,,,,,..,.,.,.......,.,   43 .,...,.+.,,,,.,.,,.,..,.,,.,.,.......,.,.,.
 32540307   A       35 ..-...,,.,,..,.,,,,.,,,,,c,.,,..,,,----   30 .,..,-.,,,,,,..,.,.,.......,.,   43 .,...,-..,,,,.,.,,.,..,.,,.,.,.......,.,.,.
 32540308   A       35 ......,,.,,..,.,,,,.t,,,,,,.,,..,,c----   30 .,..,..,,,,,,..,.,.,.......,.,   43 .,...,...,,,,.,.,,.,..,.,,.,.,.......,.,.,.
 32540309   A       35 ......,,.,,..,G,,,,.g,,,,,,.,,..,,,----   30 .,..,..,,,,,,..,.,.,.......,.,   43 .,...,...,,,,.,.,,.,..,.,,.,.,.......,.,.,.
 32540310   A       36 ......,,.,,..,.,,,,.,,,,,,,.,,..,,,,---   30 .,..,..,,,,,,..,.,.,.......,.,   43 .,...,...,,,,.,.,,.,..,.,,.,.,.......,.,.,.
 32540311   T       37 ....G.,,.,,..,A,,,,.,,,,,,,.,,..,,,,,--   30 .,..,..,,,,,,..,.,.,.......,.,   43 .,...,...,,,,.,.,,.,..,.,,.,.,.......,.,.,.
 32540312   G       38 ......,,.,,..,.,,,,.,,,,,,,.,,..,,,,,,-   30 .,..,..,,,,,,..,.,.,.......,.,   43 .,...,...,,,,.,.,,.,..,.,,.,.,.......,.,.,.
 32540313   A       38 ......,,.,,..,.,,,,.,,,,,,,.,,..,,,,,,-   30 .,..,..,,,,,,..,.,.,.......,.,   43 .,...,...,,,,.,.,,.,..,.,,.,.,.......,.,.,.
 32540314   G       39 ......,,.,,..,.,,,,.,,,,,,,.,,..,,,,,,,   30 .,..,..,,,,,,..,.,.,.......,.,   43 .,...,...,,,,.,.,,.,..,.,,.,.,.......,.,.,.
 32540315   C       39 AAAAA.,,.,,.A,A,,a,.aaa,,aaA,a..,,a,,,a   30 AaA.aA.aa,a,,A.a.a.,AA.AAAAa.a   43 AaAA.aAA.aaa,.aA,,.aA.a.aa.a.,AA.AAAAa.a.,.
 32540316   C       39 ......,,.,,..,.,,,,.,,,,,,,.,,..,,,,,,,   30 .,..,..,,,,,,..,.,.,.......,.,   43 .,...,...,,,,.,.,,.,..,.,,.,.,.......,.,.,.
 32540317   A       39 ......,,.,,..,.,,,,.,,,,,,,.,,..,,,,,,,   30 .,..,..,,,,,,..,.,.,.......,.,   43 .,...,...,,,,.,.,,.,..,.,,.,.,.......,.,.,.
 32540318   G       39 ......,,.,,..,.,,,,.,,,,,,,.,,..,,,,,,,   30 .,..,..,,,,,,..,.,.,.......,.,   43 .,...,...,,,,.,.,,.,..,.,,.,.,.......,.,.,.
 32540319   G       39 ......,,.,,..,.,,,,.,,,,,,,.,,..,,,,,,,   30 .,..,..,+,,,,..,.,.,.......,.,   43 .,...,...,,+,.,.,,.,..,.,,.,.,.......,.,.,.
 32540320   T       39 .....G,,.,,..,G,,,,.,,,,,,,.,,..,,,,,,,   30 .,..,..,,,,,,..,.,.,.......,.,   43 .,...,...,,,,.,.,,.,..,.,,.,.,.......,.,.,.
 32540321   G       39 ......,,.,,..,.,,,,.,,,,,,,.,,..,,,,,,,   30 .,..,..,,,,,,..,.,.,.......,.,   43 .,...,...,,,,.,.,,.,..,.,,.,.,.......,.,.,.
 32540322   T       38 -.....,,.,,..,A,,,,.,,,,,,,.,,..,,,,,,,   30 .,..,..,,,,,,+.,.,.,.......,.,   43 .,...,...,,,,.,.,,.,+.,.,,.,.,.......,.,.,.
 32540323   G       38 -.....,,.,,..,.,,,,.,,,,,,,.,,..,,,,,,,   30 .,..,..,,,,,,..,.,.,.......,.,   43 .,...,...,,,,.,.,,.,..,.,,.,.,.......,.,.,.
 32540324   G       37 -...-.,,.,,..,.,,,,.,,,,,,,.,,..,,,,,,,   30 .,..,..,,,,,,..,.,.,.......,.,   43 .,...,...,,,,.,.,,.,..,.,,.,.,.......,.,.,.
 32540325   T       37 -...-.,,.,,..,.,,,,.,,,,,,,.,,..,,,,,,,   30 .,..,..,,,,,,..,.,.,.......,.,   43 .,...,...,,,,.,.,,.,..,.,,.,.,.......,.,.,.
```

Figure A.8. **Example of how the improved mappability of DuploMap reduces false negatives in variant calling using PacBio CCS reads.** Pileups of 10X Genomics linked-reads and PacBio CCS reads for the individual HG002 (aligned with Minimap2 and Minimap2 + DuploMap) in a window around the position chr15:32,540,315 (hg38) are shown. Each row shows a single position, and each column represents a single read. First digit of mapping quality is shown on top (0-6) and is highlighted in red for reads with mapping quality less than 10. The variant lies within a 218 kb duplication with sequence similarity 99.5%. The variant is present in the GIAB benchmark variant calls and the 10X Genomics calls with genotype equal to 0/1. However, all CCS reads mapped using Minimap2 that have high mapping quality have the alternative allele 'A' resulting in a homozygous variant call (genotype = 1/1). After realignment using DuploMap, all reads have high mapping quality and the variant is called using Longshot with the correct genotype (0/1).

```
                    10X                                           CCS: MM2           CCS: MM2+DuploMap
      pos   ref     cov 666666666166464241633656661646666666  cov 2222222222222      cov 666666666666666666666
143457461   A        35 .,,...,,,,,,.,.,.,,.,,.,c.,..,.,,.,-   13 .,.,,,.....,,       21 .,.,,,,....,.,.,.,,,.
143457462   A        35 .,,C..,,,,,,.,.,.,,.,,.,,.,..,.,,.,-   13 .,.,,,.....,,       21 .,.,,,,....,.,.,.,,,.
143457463   T        35 .,,...,,,,,,.,.,.,,.,,.,,.,..,.,,.,-   13 .,.,,,.....,,       21 .,.,,,,....,.,.,.,,,.
143457464   T        35 .,,...,,,,,,.,.,.,,.,,.,g.,..,.,,.,-   13 .,.,,,.....,,       21 .,.,,,,....,.,.,.,,,.
143457465   T        36 .,,...,,,,,,.,.,.,,.,,.,,.,..,.,,.,.   13 .,.,,,.....,,       21 .,.,,,,....,.,.,.,,,.
143457466   C        36 .,,...,,,,,,.,.,.,,.,,.,,.,..,.,,.,.   13 .,.,,,.....,,       21 .,.,,,,....,.,.,.,,,.
143457467   T        36 .,,...,,,,,,.,.,.,,.,,.,,.,..,.,,.,.   13 .,.,,,.....,,       21 .,.,,,,....,.,.,.,,,.
143457468   T        36 .,,...,,,,,,.,.,.,,.,,.,,.,..,.,,.,.   13 .,.,,,.....,,       21 .,.,,,,....,.,.,.,,,.
143457469   A        36 .,,...,,,,,,.,.,.,,.,,.,c.,..,.,,.,.   13 .,.,,,.....,,       21 .,.,,,,....,.,.,.,,,.
143457470   C        36 .,,...,,,,,,.,.,.,,.,,.,,.,..,.,,.,.   13 .-.,,,.....,,       21 .-.,,,,....,.,.,.,,,.
143457471   C        36 .,,...,,,tttT,TtTt,Tt,.,,.t.T,.,,.,.   13 .,.,,,.....,,       21 .,.,,,tT.T.t.t.t.t,,T
143457472   A        36 .,,...,,,,,,.,.,.,,.,,.,g.,..,.,,.,.   13 .,.,,,.....,,       21 .,.,,,,....,.,.,.,,,.
143457473   A        36 .,,...,,,,,,.,.,.,,.,,.,,.,..,.,,.,.   13 .,.,,,.....,,       21 .,.,,,,....,.,.,.,,,.
143457474   C        36 .,,...,,,,,,.,.,.,,.,,.,,.,..,.,,.,.   13 .,.,,,.....,,       21 .,.,,,,....,.,.,.,,,.
143457475   T        36 .,,...,,,,,,.,.,.,,.,,.,,.,..,.,,.,.   13 .,.,,,.....,,       21 .,.,,,,....,.,.,.,,,.
143457476   C        36 .,,...,,,,,,.,.,.,,.,,.,,.,..,.,,.,.   13 .,.,,,.....,,       21 .,.,,,,....,.,.,.,,,.
143457477   A        35 .,-...,,,,,,.,.,.,,.,,.,,.,..,.,,.,.   13 .,.,,,.....,,       21 .,.,,,,....,.,.,.,,,.
143457478   A        35 .,-...,,,,,,.,.,.,,.,,.,,.,..,.,,.,.   13 .,.,,,.....,,       21 .,.,,,,....,.,.,.,,,.
143457479   A        34 .,-...-,,,,,.,.,.,,.,,.,c.,..,.,,.,.   13 .,.,,,.....,,       21 .,.,,,,....,.,.,.,,,.
143457480   A        33 -,-...-,,,,,.,.,.,,.,,.,,.,..,.,,.,.   13 .,.,,,.....,,       21 .,.,,,,....,.,.,.,,,.
143457481   G        32 ---...-,,,,,.,.,.,,.,,.,,.,..,.,,.,.   13 .,.,,,.....,,       21 .,.,,,,....,.,.,.,,,.
```

Figure A.9. **Example of a potential false negative call in the GIAB benchmark variant call set identified using DuploMap alignments.** Pileups of 10X Genomics linked-reads and PacBio CCS reads for the individual HG002 (aligned with Minimap2 and DuploMap) in a window around the position chr1:143,457,471 (hg38) are shown. Each row shows a single position, and each column represents a single read. First digit of mapping quality is shown on top (0-6). The position lies within a 220 kb long segmental duplication with sequence similarity 99.6%. The variant lies in high-confidence GIAB regions, but is absent in the GIAB benchmark variant calls. Nevertheless, the variantt is present in the 10X Genomics calls. However, all CCS reads mapped using Minimap2 have reference allele 'C' and hence the variant is not called. After realignment using DuploMap, the number of reads covering this position increases from 13 to 21 and includes 8 reads that support alternative allele 'T'. Using Longshot, a variant is called at this position that matches the 10X variant call.

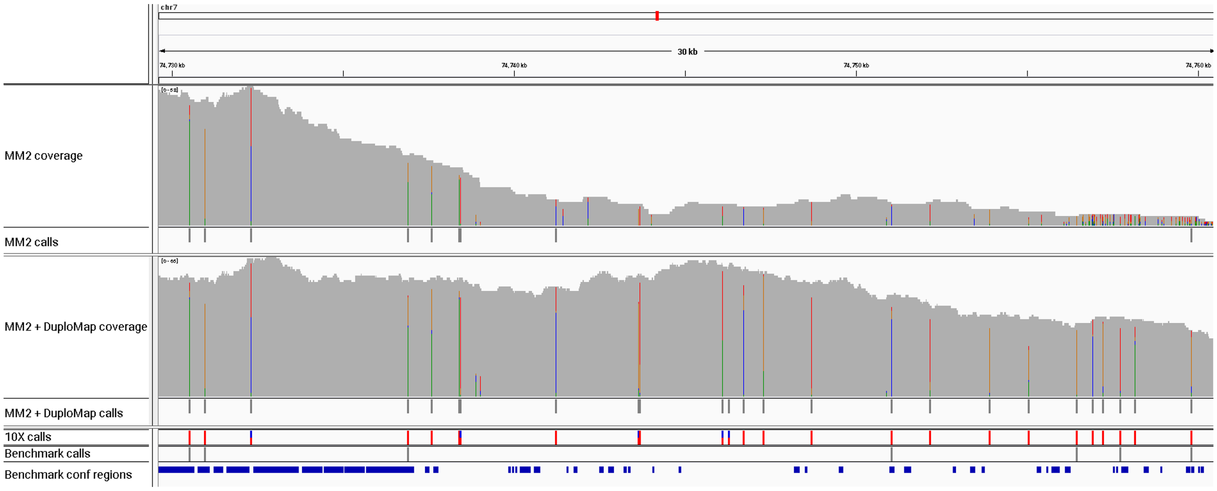


Figure A.10. **Example of a medically-relevant gene (*GTF2I*) with additional variants called using DuploMap alignments.** An Integrated Genomics Viewer (IGV) [146] view of a 30 kb region (chr7:74729600-74760692, hg38 reference) within the *GTF2I* gene that overlaps a segmental duplication is shown. The region is not well covered using Minimap2 alignments (reads with mapping quality ≥ 10) but shows improved coverage using Minimap2+Duplomap alignments. Variants called using the Minimap2, Minimap2+Duplomap alignments, and 10X reads are also shown. 25 SNVs are called using Minimap2+DuploMap alignments that are identical to the 10X variant calls. Only 9 SNVs are called using the Minimap2 alignments. The region is partially covered in the GIAB v4.1 benchmark variant calls with only 7 variant calls.

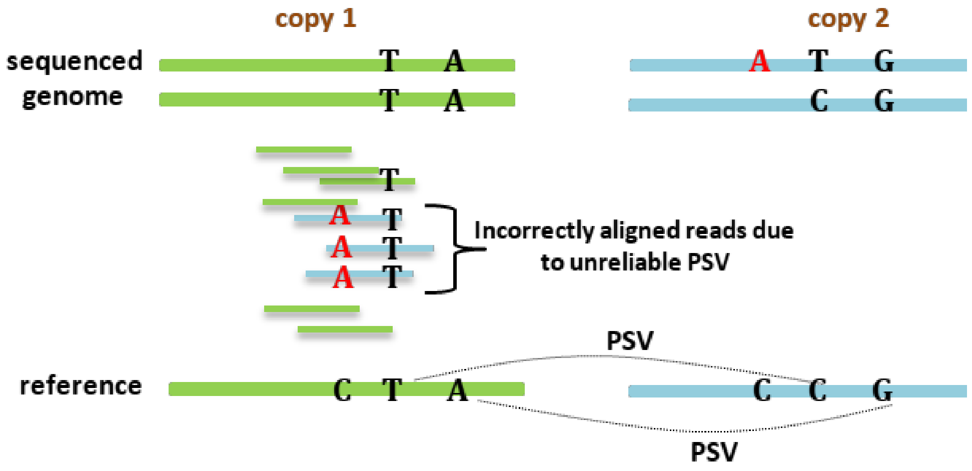


Figure A.11. **Illustration of how incorrect short read mapping due to unreliable PSVs can lead to false positive and false negative variant calls.** A two-copy segmental duplication is shown with two PSVs that distinguish 'copy 1' and 'copy 2'. The sequenced genome carries a variant (A allele) on one of the haplotypes of 'copy 2'. One of the two PSVs is actually a variant in the sequenced genome with the 'T' allele instead of the 'C' allele in 'copy 2'. Hence, reads with the 'A' allele that originate from 'copy 2' are mismapped to 'copy 1' resulting in a false positive variant call at the homologous position in 'copy 1' with the alternate allele being identical to the PSV allele.

A.2 Supplementary Methods

A.2.1 Filtering PSVs

To identify low-complexity PSVs we count the number of unique k-mers (with $k = 3$) in a window around the PSV. PSVs for which the number of k-mers divided by the maximal number of k-mers for the window of the same size is less than 60% for substitutions and 80% for indels are filtered out. We also filter out PSVs for which it is difficult to distinguish between the two alleles due to high sequencing error rates. For a read r that covers a PSV v, we calculate the alignment probabilities for each of the two alleles of the PSV $s_v^{(i)} = P(r_v \mid S_v^{(i)})$ and $s_v^{(j)} = P(r_v \mid S_v^{(i)})$. We say that the read has an *ambiguous* alignment for the PSV if $\max\{s_v^{(i)}, s_v^{(j)}\}/\min\{s_v^{(i)}, s_v^{(j)}\} < 4$. After the first iteration of the DuploMap algorithm, we remove all PSVs for which 30% or more reads have an ambiguous alignment. This filtering removes PSVs that were not identified as low-complexity but still have noisy local realignment probabilities. It is possible that the PSV in the sequenced genome has a sequence different from the two known alleles $S_v^{(i)}$ and $S_v^{(j)}$. This step can also filter out such PSVs.

A.2.2 Identifying candidate alignment locations for a read

In the PSV database, we store each pair of homologous sequences as a collection of pairs of windows $(w^{(1)},\ w^{(2)})$, where each window is approximately 100 bp in length. The windows are constructed from the pairwise alignment such that window $w^{(1)}$ in one of the sequences is aligned to the window $w^{(2)}$ in the other. For an aligned read r, we consider windows $\left\{w_i^{(1)}\right\}_{i=1}^{n}$ that intersect its primary alignment. Using the database, we can identify all windows $\left\{w_i^{(2)}\right\}_{i=1}^{n}$ that are homologous to the windows of the primary alignment of the read. Without loss of generality, suppose that all pairs of windows are on same strand in the genome. We reorder the indices so that windows $\left\{w_i^{(2)}\right\}_{i=1}^{n}$ are sorted by their genomic positions. Additionally, we define a function $pos_1\left(w^{(2)}\right)$ that returns a genomic position of the window $w^{(1)}$. To identify possible alignment locations we search for pairs of indices $i \leq j$ such that

1. windows $w_i^{(2)}$ and $w_j^{(2)}$ have the same order in the read: $pos_1\left(w_i^{(2)}\right) \leq pos_1\left(w_j^{(2)}\right)$,
2. location is not too short: $j \geq i + m$, where m is half of the number of non-overlapping windows in the initial alignment,
3. location generated from windows $w_i^{(2)}$ and $w_j^{(2)}$ is not more than 20% longer than the biggest of the read length and the initial alignment size,
4. no other pair of indices $i' \leq i,\ j' \geq j$ produces a possible alignment location.

For an existing primary alignment with start x_l, end x_r and soft clipping y_l and y_r we generate an alignment location by adding padding of size $\max\left\{0, y_l + pos_1\left(w_i^{(2)}\right) - x_l\right\}$ to the left of the window $w_i^{(2)}$. Similarly, we add padding of size $\max\left\{0, y_r + x_r - pos_1\left(w_j^{(2)}\right)\right\}$ to the right of the window $w_j^{(2)}$.

A.2.3 LCS-based filtering of alignment locations

We filter possible alignment locations using longest common subsequences (LCS) between the k-mers of the read and the k-mers of each candidate alignment location ($k = 11$, by default). The LCSk++ algorithm [73] is used to find the LCS. If one or more of the alignment locations for a read is located near a gap or missing sequence in the reference genome, the LCS score may not reflect the alignment of the full sequence of the read. To avoid this behavior, we compute the LCS scores for a pair of locations using a truncated read sequence. The read is truncated using the location of the last (or first) k-mer that is shared between the read and both locations.

To select a smallest non-empty subset of alignment locations that dominate all other locations we construct a directed graph, where each node represents a single location. For a pair of locations i and j if location i dominates location j we add an edge from the node j to node i (worse to best). We add edges in both directions if neither location dominates. Afterwards, we split the graph on strongly connected components [147] and select all locations from the sink component.

A.2.4 Identifying reads with high discordance with PSVs

To find reads that show high discordance with PSVs, we calculate the number of conflicts (mismatches) between each read and the PSVs it intersects. For a given read r mapped to location i, we analyse all reliable PSVs that intersect the new primary alignment location for the read. The second position for different PSVs may lie in different homologous locations (denoted by $(-i)$). We define the conflict rate for the read r as

$$\frac{\sum_v \mathbb{1}\left(s_v^{(-i)}/s_v^{(i)} \geq 10\right)}{\sum_v \mathbb{1}\left(\max\{s_v^{(i)}, s_v^{(-i)}\}/\min\{s_v^{(i)}, s_v^{(-i)}\} \geq 10\right)},$$

where $\mathbb{1}$ denotes indicator function, $s_v^{(i)} = P(r_v \mid S_v^{(i)})$ and $s_v^{(-i)} = P(r_v \mid S_v^{(-i)})$ represent alignment probabilities for two alleles of the PSV v. In the above formula, the denominator represents the numbers of PSVs that have big difference between alignment probabilities for two alleles. The value in the numerator shows the number of PSVs that do not support location i.

For a given cluster of segmental duplications, we estimate the average conflict rate using all reads mapped to the cluster with high mapping quality and with at least five PSVs. We use the average conflict rate and the binomial test to test if the observed number of conflicting PSVs is higher than expected. Reads for which the Bonferroni-corrected p-value is lower than 0.05 are assigned a low mapping quality (5 by default).

A.2.5 Mappability of exons

To calculate the mappability of disease-associated genes using long reads, we calculated the percentage of positions covered by at least 10 reads with mapping quality greater than a specific threshold. We only analysed positions that were located in at least one exon of the GENCODE annotation for that gene [130].

A.2.6 Estimating coverage

To calculate read coverage for PacBio and Oxford Nanopore whole-genome sequencing we selected 200,000 positions at random for the hg38 genome (using `bedtools random`). We then selected 100,000 positions (at random) that lie on chromosomes 1-22 outside of centromeres and telomeres. For each position x we counted the number of reads (passing samtools flag 3844) with alignment starting at position $\leq x$ and ending at position $\geq x$. Then, the median value of the measured coverages was taken.

A.2.7 Pileups

We constructed pileups using the pileuppy tool `v0.2.1` available at https://gitlab.com/tprodanov/pileuppy.

A.2.8 Datasets

Alignment and variant calling files can be found at the following links:

- HG002 (NA24385) PacBio CLR: https://ftp-trace.ncbi.nih.gov/giab/ftp/data/AshkenazimTrio/HG002_NA24385_son/PacBio_MtSinai_NIST/PacBio_minimap2_bam
- HG002 (NA24385) PacBio CCS: https://ftp-trace.ncbi.nih.gov/giab/ftp/data/AshkenazimTrio/HG002_NA24385_son/PacBio_CCS_15kb/GRCh38_no_alt_analysis

- HG002 (NA24385) Oxford Nanopore: https://ftp-trace.ncbi.nlm.nih.gov/giab/ftp/data/AshkenazimTrio/HG002_NA24385_son/Ultralong_OxfordNanopore/combined_2018-08-10
- HG002 (NA24385) 10X: https://ftp-trace.ncbi.nih.gov/giab/ftp/data/AshkenazimTrio/analysis/10XGenomics_ChromiumGenome_LongRanger2.2_Supernova2.0.1_04122018/GRCh38
- HG002 (NA24385) GIAB benchmark calls: https://ftp-trace.ncbi.nlm.nih.gov/giab/ftp/release/AshkenazimTrio/HG002_NA24385_son
- HG003 (NA24149) PacBio CLR: https://ftp-trace.ncbi.nlm.nih.gov/giab/ftp/data/AshkenazimTrio/HG003_NA24149_father/PacBio_MtSinai_NIST/PacBio_minimap2_bam
- HG004 (NA24143) PacBio CLR: https://ftp-trace.ncbi.nlm.nih.gov/giab/ftp/data/AshkenazimTrio/HG004_NA24143_mother/PacBio_MtSinai_NIST/PacBio_minimap2_bam
- HG005 (NA24631) PacBio CCS: https://ftp-trace.ncbi.nih.gov/giab/ftp/data/ChineseTrio/HG005_NA24631_son/PacBio_SequelII_CCS_11kb/HG005_GRCh38
- HG005 (NA24631) GIAB benchmark calls: https://ftp-trace.ncbi.nlm.nih.gov/giab/ftp/release/ChineseTrio/HG005_NA24631_son
- HG001 (NA12878) Oxford Nanopore: https://github.com/nanopore-wgs-consortium/NA12878
- HG001 (NA12878) PacBio CCS: https://ftp-trace.ncbi.nih.gov/giab/ftp/data/NA12878/PacBio_SequelII_CCS_11kb
- HG001 (NA12878) 10X: https://ftp-trace.ncbi.nih.gov/giab/ftp/data/NA12878/10Xgenomics_ChromiumGenome_LongRanger2.1_09302016/NA12878_GRCh38
- HG001 (NA12878) Platinium Genome: ftp://ussd-ftp.illumina.com/2017-1.0/hg38/small_variants/NA12878/

appendix B

Supplementary Material for Chapter 3

B.1 Supplementary Figures and Tables

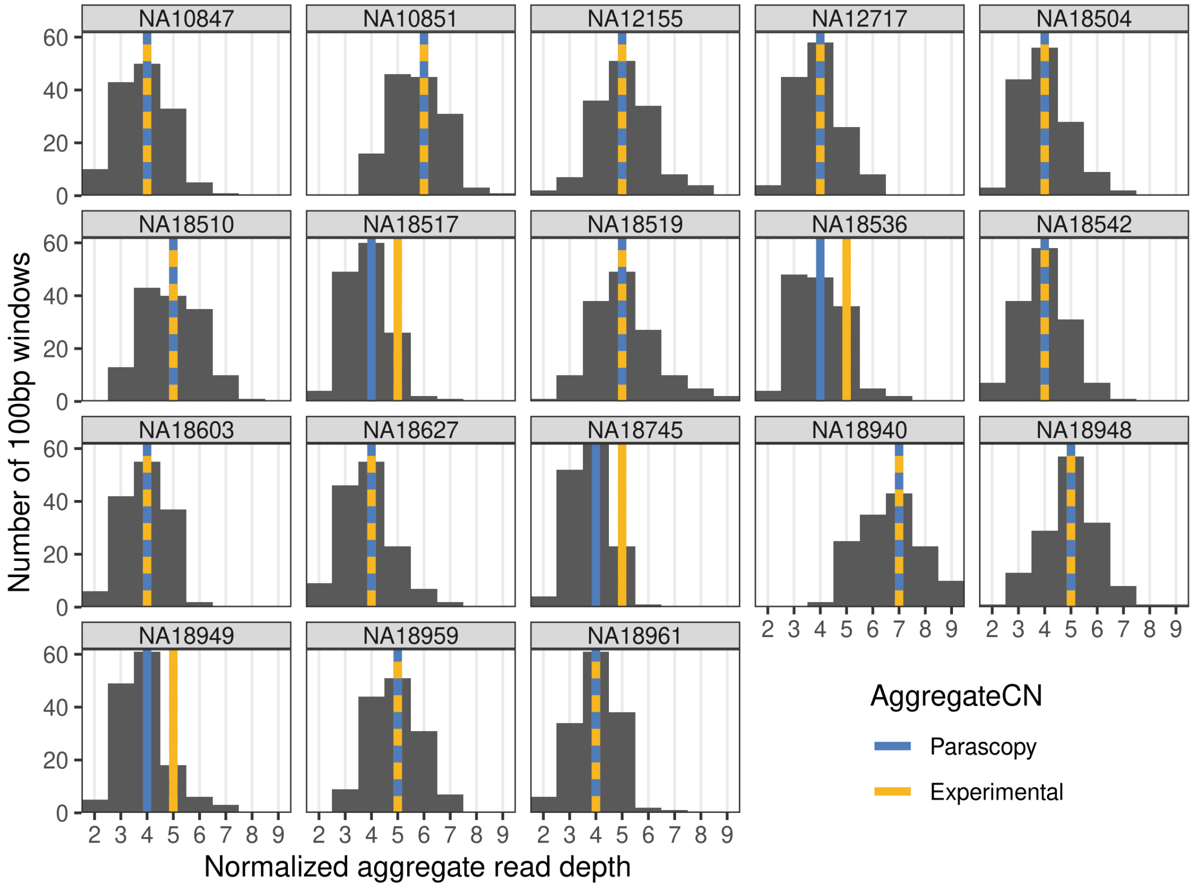


Figure B.1. **Comparison between Parascopy and experimental *AggregateCN* estimates for 18 1kGP samples.** Histograms show aggregate read depth distribution at 140 100 bp windows within *NPY4R/2* duplication. Experimental *AggregateCN* values were obtained for 18 1kGP samples using ddPCR [148]. Parascopy *AggregateCN* estimates match with experimental values in 14 samples (shown with dashed blue-yellow lines). In the remaining 4 samples, Parascopy and ddPCR *AggregateCN* estimates are shown with separate blue and yellow vertical lines, respectively. The copy number estimates from CNVnator and QuicK-mer2 match Parascopy's estimates for all 4 samples in which Parascopy and experimental copy number estimates disagree.

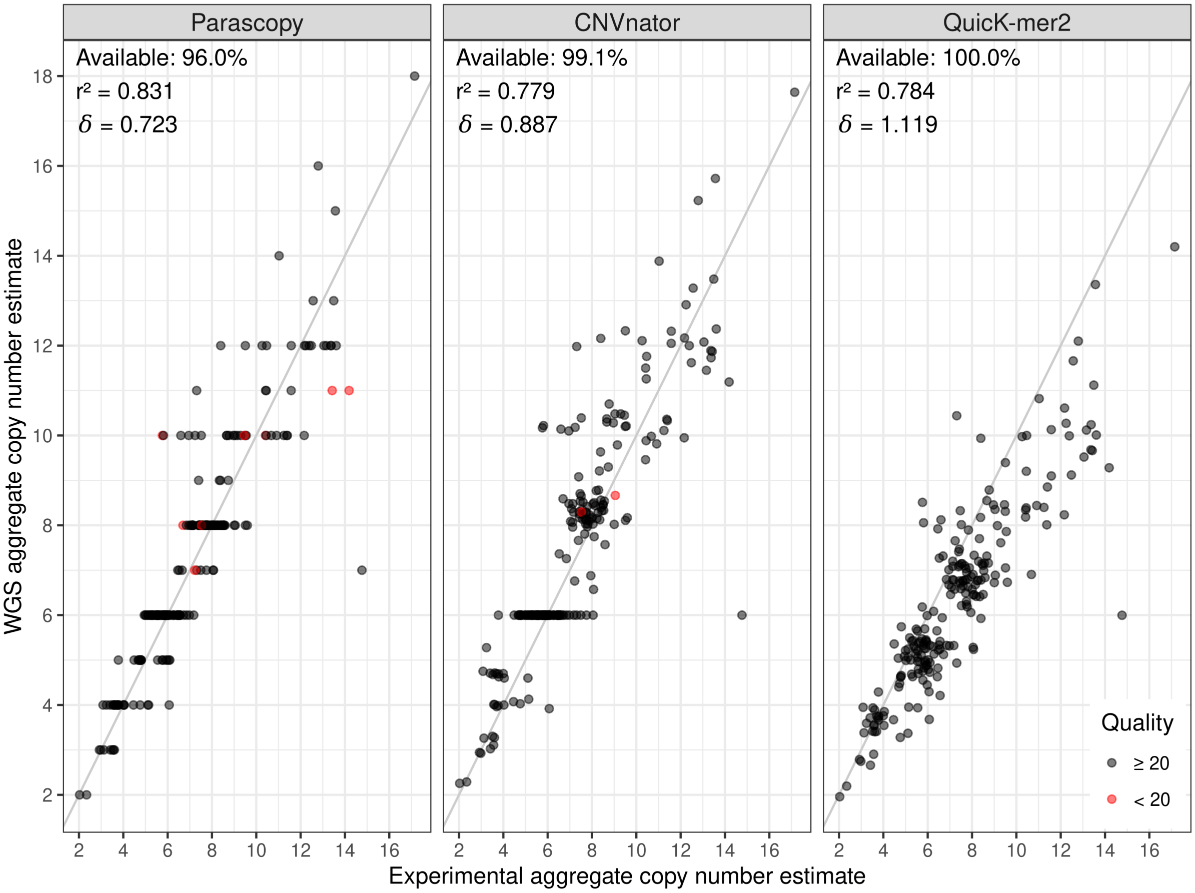


Figure B.2. **Comparison of experimental *AggregateCN* estimates for 225 1kGP samples at the *AMY1A/B/C* locus with Parascopy, CNVnator and QuicK-mer2 *AggregateCN* estimates.** The percentage of available *AggregateCN* estimates, Pearson correlation coefficient value (r^2) and average absolute error (δ) for each method are shown in the top left of each plot. Low quality copy number estimates are marked in red.

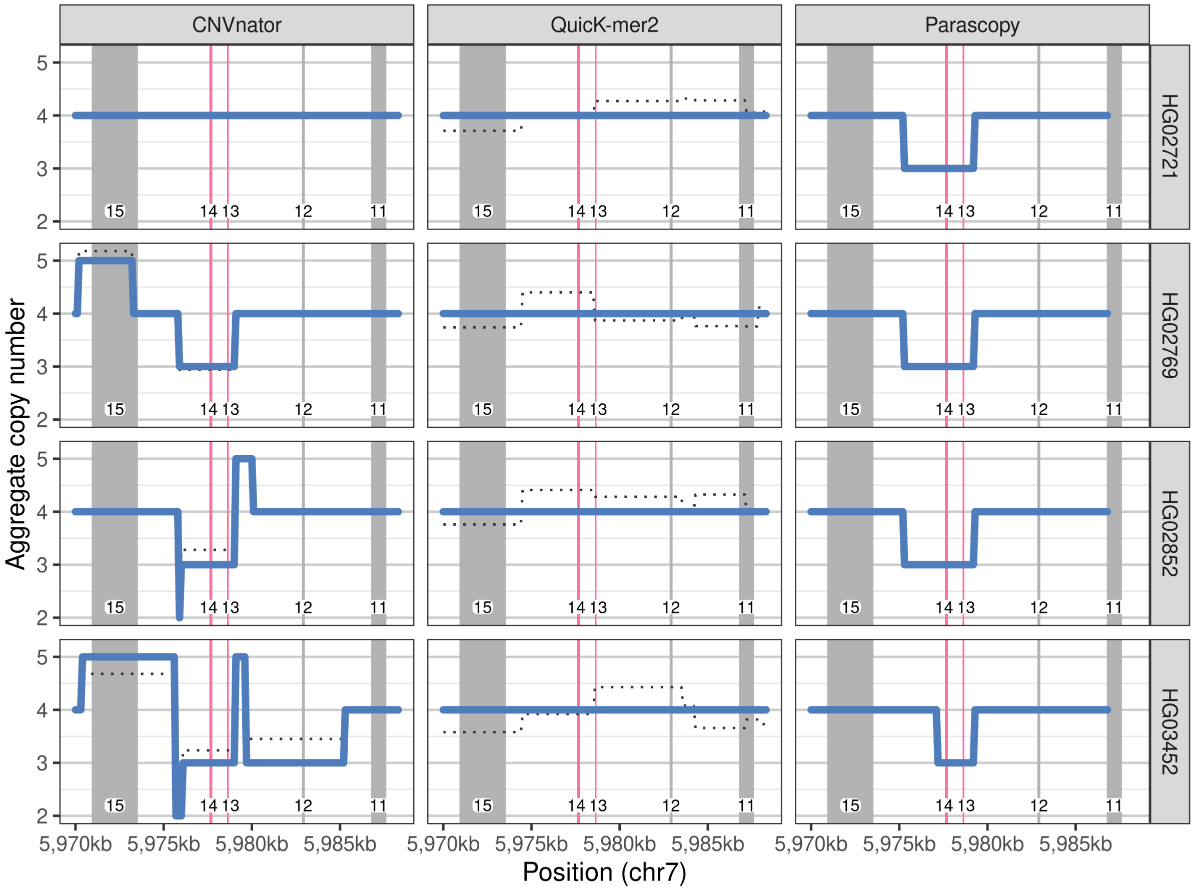


Figure B.3. **Comparison of CNVnator, QuicK-mer2 and Parascopy *AggregateCN* estimates for four samples at the *PMS2* locus.** The four panels correspond to the four samples which were previously reported [112] to have a partial deletion within the *PMS2CL* pseudogene (overlapping *PMS2* exons 13 and 14). The three vertical columns show copy number profiles for the three CNV-detection methods: CNVnator, QuicK-mer2 and Parascopy. Black dotted line shows fractional *AggregateCN* estimates, while the blue solid line shows integer *AggregateCN* estimates. Vertical gray and red rectangles display the duplicated *PMS2* exons 11–15.

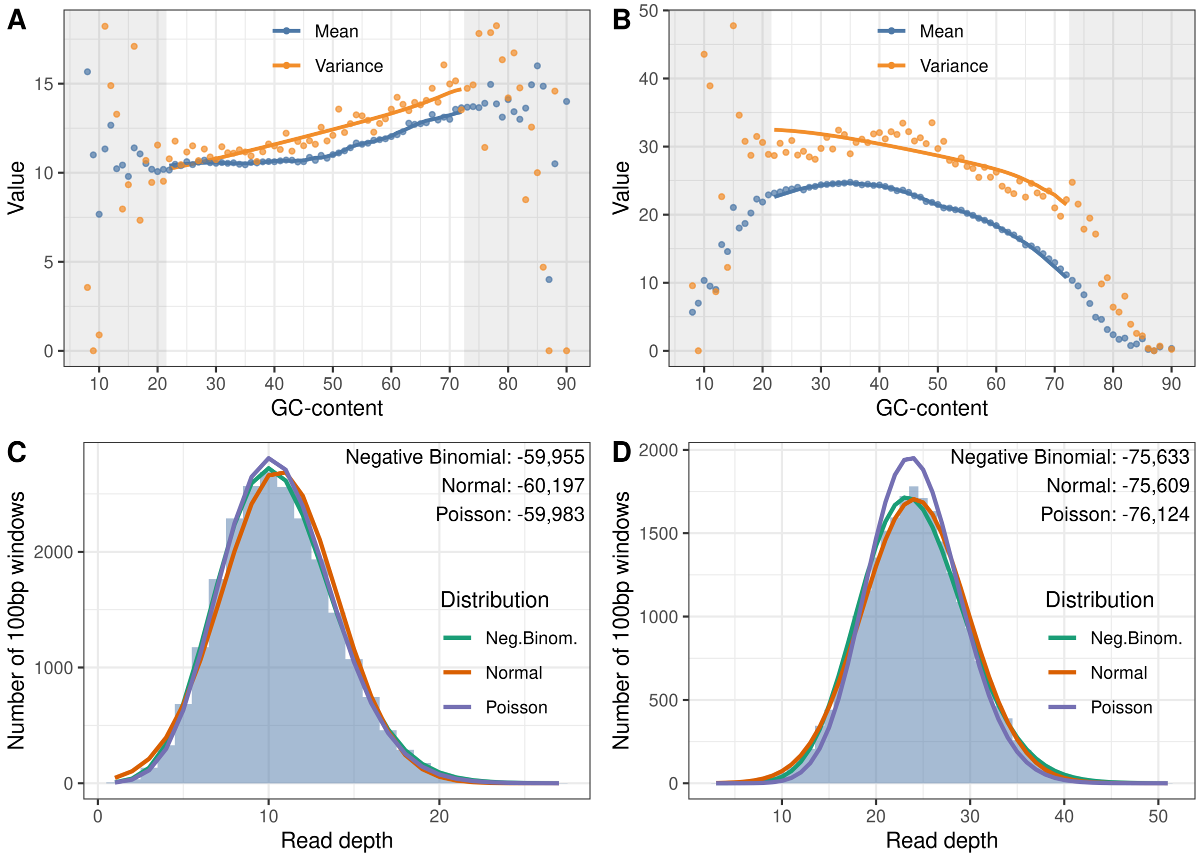


Figure B.4. **Estimation of mean read depth (and variance) using non-duplicated windows across the genome.** The mean and variance values across different GC-bins are shown for one sample with PCR-free WGS data (A) and PCR-based WGS data (B). Dots show empirical read depth mean and variance as a function of GC-content, while solid lines show smoothed mean and variance approximations, obtained using the LOWESS procedure [149, 150] (not calculated in gray areas). Fit of various distributions (and corresponding log-likelihood values) to the read depth are shown for PCR-free data (C) and PCR-based data (D). For PCR-free WGS, both Negative Binomial and Poisson distributions give a similar fit of the read depth distribution while the Poisson has a significantly worse fit for PCR-based WGS.

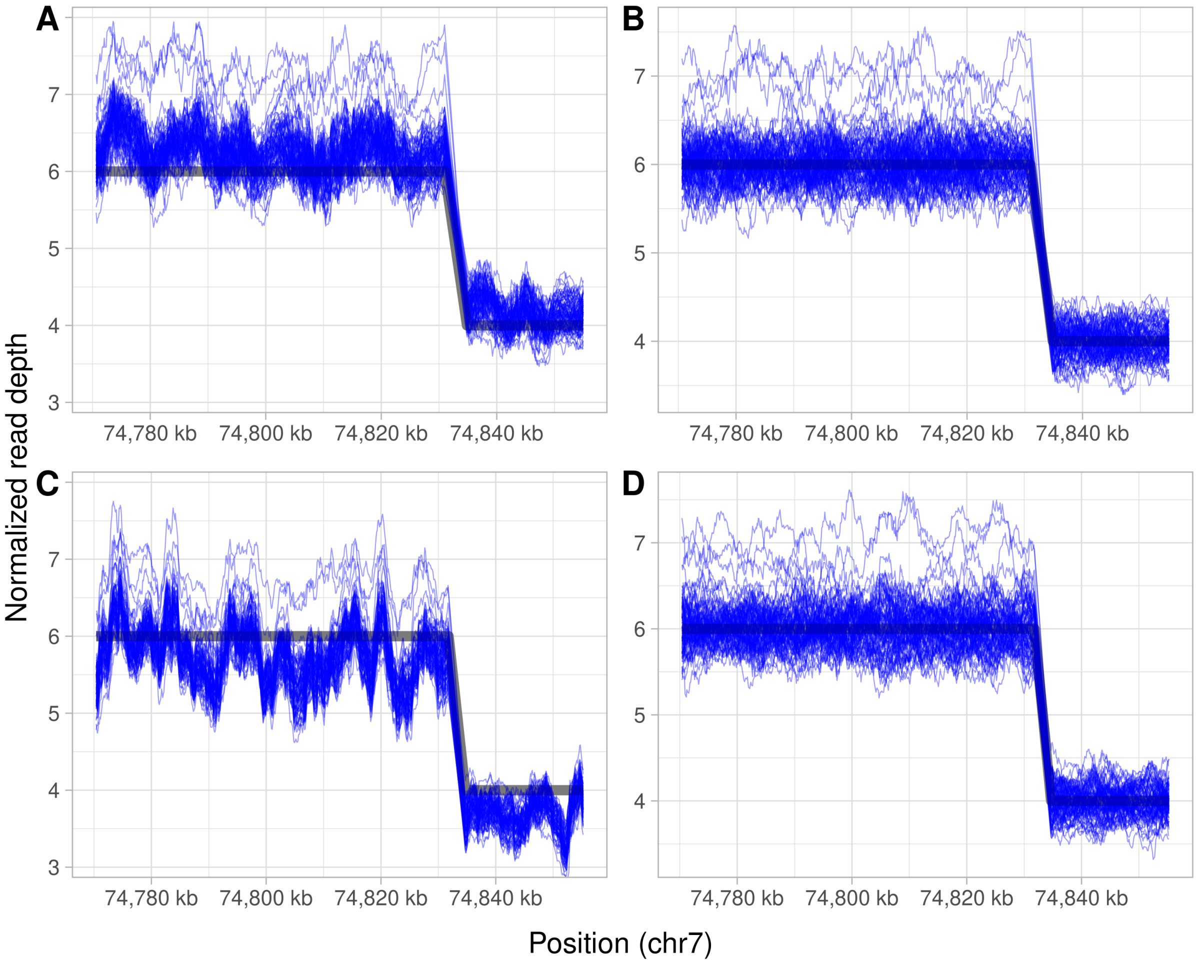


Figure B.5. **Normalized read depth in moving windows for 83 Han Chinese samples from IGSR and BGI datasets.** Each panel consists of 83 blue lines, each shows normalized read depth at three- and two-copy duplication chr7:74,769,000-74,856,500 that harbors *NCF1* and *GTF2IRD2* genes. Normalized read depth is averaged for each sample across moving 2,500 kb windows. Gray line shows reference copy number. (A) IGSR dataset: no read depth scaling. (B) IGSR dataset: scaling read depth based on window-specific multipliers m_w (same for all samples). (C) BGI dataset: no read depth scaling. (B) BGI dataset: scaling read depth based on window-specific multipliers m_w (same for all samples).

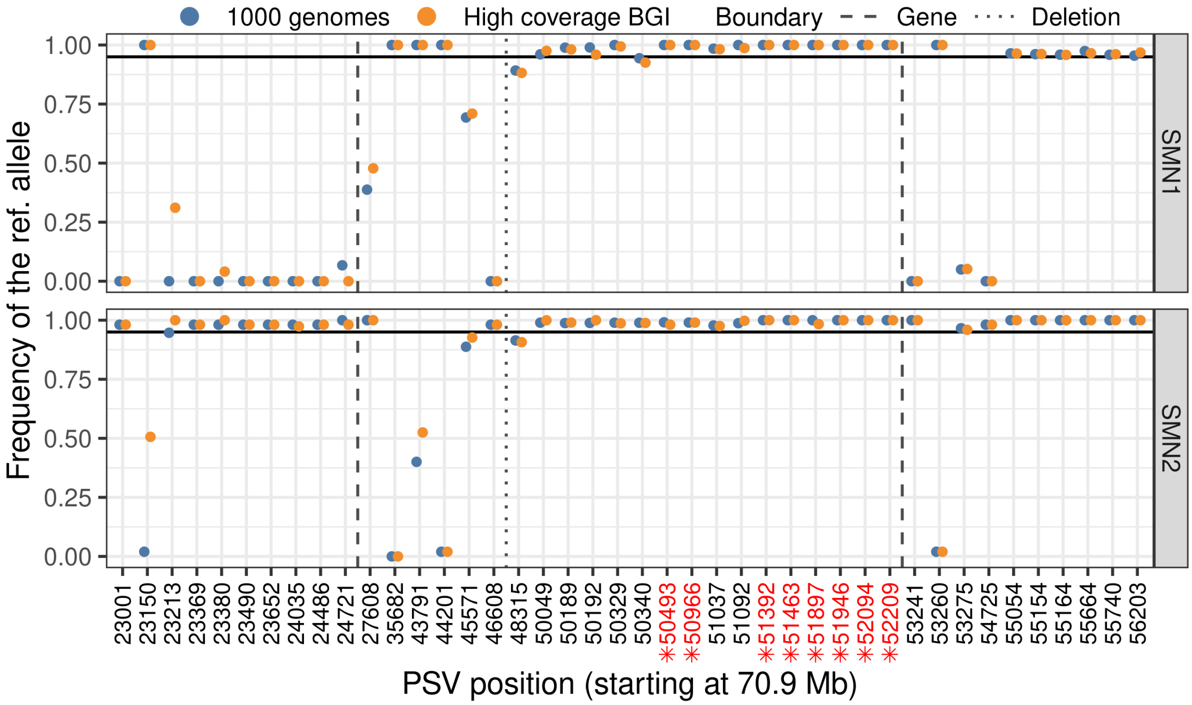


Figure B.6. **PSV *f*-values for the *SMN1/2* locus estimated using Parascopy for the same set of 83 Han Chinese samples with two different WGS datasets.** The plot shows 42 PSVs in the vicinity of the *SMN1* gene, of them 22 lie within *SMN1*. Horizontal black line shows reliability threshold (0.95), reliable PSVs have *f*-values over the threshold on both copies. Reliable PSVs used in SMNCopyNumberCaller are shown in red and marked by an asterisk.

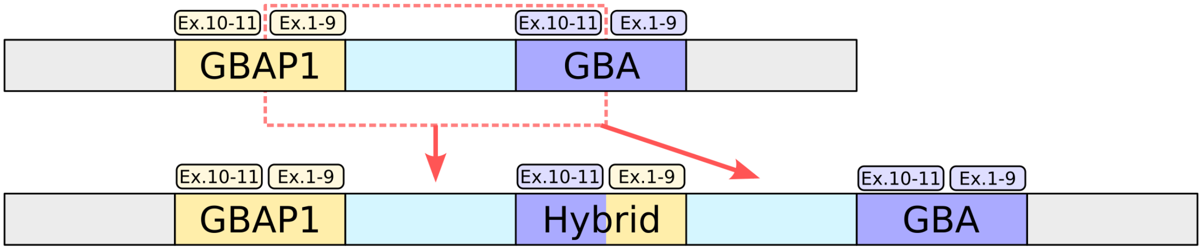


Figure B.7. **Structure of the duplication at the *GBA* locus.** The duplication affects a region between the *GBAP1* pseudogene exons 1-9 and the *GBA* gene exons 10-11, including a region between *GBAP1* and *GBA*, which is unique in the reference genome (shown in light blue). The duplication was constructed using a visual inspection of *GBA* locus *de-novo* assemblies. *De-novo* assemblies were obtained using SPAdes [151] based on the reads mapped to chr1:155,200,000-155,260,000 in two 1kGP samples (NA19031 and NA19159).

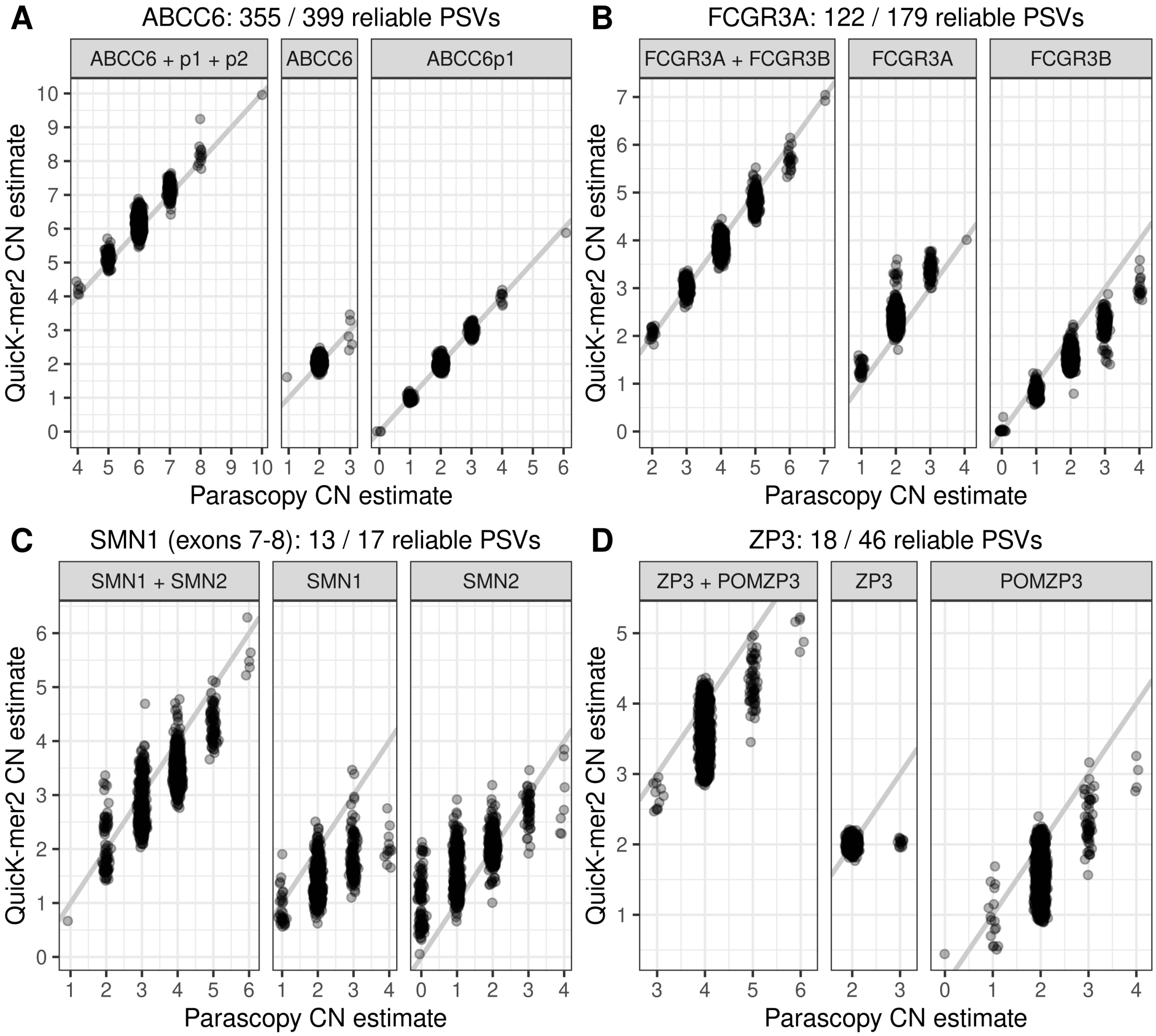


Figure B.8. **Comparison of Parascopy and QuicK-mer2 *AggregateCN* and *ParalogCN* estimates for four duplicated disease-associated genes: (A) *ABCC6*, (B) *FCGR3A/B* (B), (C) *SMN1/2* and (D) *ZP3/POMZP3*.** The total number and number of reliable PSVs are also shown for each locus. For visual clarity, small jitter was added to the integer Parascopy copy number estimates.

Table B.1. **Accuracy of detecting partial deletions at the *SMN1/2* and *PMS2/PMS2CL* loci using four different methods.** For the *SMN1/2* partial deletion, 79/1109 samples were identified to carry the deletion event using MLPA data [113]. For the *PMS2/PMS2CL* deletion, 4/150 samples with deletions were identified using LR-PCR [112]. *AggregateCN* estimates at single positions within *SMN1* exons 2 & 8 and *PMS2* exons 14 & 15 were used for evaluation of each method.

Method	*SMN1/2*		*PMS2/PMS2CL*	
	Sensitivity	Specificity	Sensitivity	Specificity
Parascopy (Qual ≥ 20)	1.000	1.000	0.500	1.000
Parascopy (Qual ≥ 0)	1.000	1.000	1.000	1.000
SMNCopyNumberCaller	1.000	0.999	—	—
CNVnator (Qual ≥ 20)	0.797	0.850	0.500	0.743
CNVnator (Qual ≥ 0)	0.823	0.849	0.750	0.691
QuicK-mer2	0.709	0.382	0.000	1.000

Table B.2. **Concordance of Parascopy copy number estimates between replicate samples across 167 duplicated loci obtained using two independent sets of model parameters.**

Data type	Metric	(1) CHB	(2) IBS	(3) IBS 2/3	(4) IBS 1/3
AggregateCN	Available estimates (%)	96.3	95.9	94.0	88.3
	Concordance (%)	100.0	100.0	100.0	99.9
	Mean absolute difference	0.000	0.000	0.000	0.001
ParalogCN	Available estimates (%)	71.8	71.9	71.7	70.4
	Concordance (%)	98.9	98.9	98.8	98.4
	Mean absolute difference	0.043	0.033	0.034	0.038

(1) 103 Han Chinese samples analyzed using EAS and EUR model parameters
(2) 107 Iberian samples analyzed using EAS and EUR model parameters
(3-4) 107 Iberian samples subsampled to two-third & one-third coverage (EAS model parameters) and compared to full-coverage dataset (EUR model parameters)

B.2 Supplementary Methods

B.2.1 Creating homology table

The structure of the homology table is similar to the databases of segmental duplications that is available for human and other genome assemblies [6]. The motivation for constructing the homology table is to store extra information for pairs of homologous sequences and to additionally store short pairs of homologous sequences (< 1 kb) that are absent in segmental duplication databases.

To create the homology table, we split the reference genome into artificial reads of fixed length (900 bp) at a gap of every 150 bp and align the reads back to the reference genome using BWA [31]. For reads from repetitive regions, BWA reports the original location of the read as well as other regions in the genome, which we will call *homologous segments*. For each homologous sequence, we store the genomic coordinates, sequence alignment (CIGAR) and alignment strand. We filter homologous segments by length and sequence similarity (≥ 250 bp and $\geq 96\%$, by default). If the number of retained homologous segments for a region is more than a threshold (≥ 10 by default), we discard the read and mark its location as too complex for subsequent analysis.

Next, we combine overlapping read–homologous segment pairs into long *duplications*. To do that, we construct a directed graph where each node stores a read–segment pair (a read with n homologous segments would be represented by n nodes in the graph). We create an edge between nodes if both reads and their homologous segments overlap and are on the same strand. Direction of the edge is determined by the order of the reads and should match the order of the corresponding homologous segments.

We start the graph simplification by removing transitive edges. Next, we attempt to simplify the graph in cases where a node has two or more in- or out-edges. Without loss of generality, suppose there is a node with two or more out-edges and all homologous segments are on the "+" strand. The first node consists of the read $\tilde{a}$ and homologous segment $\tilde{b}$ and out-edges lead to nodes with reads $\{a_i\}$ and homologous segments $\{b_i\}$. We define two vectors of distances $d_{i1} = \text{dist}(\tilde{a}, a_i)$ and $d_{i2} = \text{dist}(\tilde{b}, b_i)$, where $\text{dist}(x, y)$ represents the distance between starts of genomic regions x and y. In other words, d_{i1} and d_{i2} represent the jump between two consecutive reads and their homologous counterparts. We will also denote the length of the region x as $\text{len}(x)$. Afterwards, we mark an edge i as *redundant* if at least one of the following statements is true:

- $d_{i1} \; > \; \min\{\text{len}(\tilde{a}),\; \text{len}(a_i)\} \,/\, 2$,
- $d_{i2} \; > \; \min\{\text{len}(\tilde{b}),\; \text{len}(b_i)\} \,/\, 2$,
- $\min\{d_{i1},\; d_{i2}\} \; > \; \max\{d_{i1},\; d_{i2}\} \,/\, 2$.

We then remove all redundant out-edges if at least one non-redundant out-edge remains. The reasoning behind this procedure is that if there are several possible continuations of the long homologous region, they must conflict with each other (because all transitive edges were already removed). This implies that there is a repeat, which is shorter than the artificial read size, and the reads or their homologous segments align to different copies of this repeat. In that case we want to keep an edge where jumps d_{i1} and d_{i2} are relatively small (first two statements), and similar in size (third statement). Note, that removed edges do not lead to removed nodes, so no information about additional small homologous regions is lost.

After graph simplification, we search for paths in the graph such that every node except the first has one in-edge, and every node except the last has one out-edge. We discard short paths (less than 4 nodes by default), if they do not form a separate connected component in the graph, and discard regions represented by these nodes from any further analysis. Finally, we combine a path of nodes into a single duplication. If an obtained duplication overlaps itself, we split it into several shorter duplications such that no duplication is self-overlapping.

B.2.2 Calculating background read depth

In order to accurately estimate copy number from read depth in repetitive regions of the genome, we calculate read depth in a large number of unique regions of the genome (reference copy number 2). In case of the human genome, we use a predetermined set of fixed-length (default = 100 bp) non-overlapping windows ($\approx$ 90,000 windows for both hg19 and hg38 versions of the human genome). For other genomes, it is feasible to select a random set of non-overlapping fixed length windows outside repetitive regions.

For each window, the read depth is calculated by counting the number of read pairs for which the center of the first read lies within the window. This ensures that each read is counted once, and read counts from nearby windows are independent. For the same reason, we consider each paired-end read as a single entity. A window may have an abnormal read depth if it contains an insertion or deletion, overlaps a transposable element or other short duplication, or if it has low-complexity sequence. Such windows can skew the background read depth distribution. Therefore, we discard a window and call it *irregular* if at least 10% of reads in the window correspond to one of the three categories: reads with low mapping quality (< 10); reads not mapped in a proper pair; or reads with soft clipping at the ends. In addition, we also remove one adjacent window to the left and right of each *irregular* window.

For each sample and each GC-content value we aim to find a separate set of distribution parameters that would explain read depth in unique regions of the genome. To a select read depth distribution we evaluated Gaussian, Poisson and Negative Binomial distributions (Figure B.4C-D) and found that Negative Binomial fits the observed values better, which is consistent with the previous studies [152, 153].

Next, we use LOWESS smoothing procedure [149, 150] to approximate read depth mean and variance for various GC-content values. As input, LOWESS takes a list of points (x, y) with the corresponding weights, if needed. As output, LOWESS provides a smoothed mean $\tilde{y}$ value for each requested $\tilde{x}$ value. Additionally, LOWESS has two parameters: local polynomial degree (we use degree 1) and a *fraction* parameter, which specifies that for each $\tilde{x}$ value only the closest *fraction* of the input points will be used. We run LOWESS smoothing procedure with $\tilde{x} = [0, 100] \cap \mathbb{Z}$ and the following parameters:

- **Read depth mean:** As input to LOWESS we use a set of points with x = window GC-content and y = window read depth (one point for each window). By default, we use *fraction* parameter = 0.1.

- **Read depth variance:** For read depth variance we cannot use individual read depth observations, therefore we create one point for each GC-content value, for which there are at least 10 windows. As y values we use read depth variance for windows with the corresponding GC-content, and we provide weight of the point based on the number of the corresponding windows. Here, we use *fraction* parameter = $2/3$.

As small number of genomic windows can lead to incorrect estimation of read depth mean and variance, we discard a set of very small and very large GC-content values and do not use windows with such GC values in copy number estimation. By default, we keep GC-content values u, for which there are both $\geq$ 1000 input genomic windows with GC-content values $\leq u$ and $\geq$ 1000 windows with values $\geq u$. This way, for the default set of 90,000 windows for the hg38 reference genome we keep GC-content values from 22 to 72 (Figure B.4A-B).

For each non-discarded GC-content value we calculate Negative Binomial parameters as $n = \mu^2/(v - \mu)$ and $p = \mu/v$ according to the methods of moments [154], where μ and v are the read depth mean and variance approximations obtained using the LOWESS procedure. As Negative Binomial random variables must have variance greater or equal to the mean, we update variance as $v \longleftarrow \max\{v, \mu + 0.001\}$ before calculating parameters n and p.

B.2.3 Re-mapping reads

In order to accurately calculate aggregate read depth and PSV-allelic read depth, we use a set of pooled reads. For each region R, we re-map reads from regions, homologous to R, back to R. We do that by utilizing sequence alignments between duplication copies, stored in the homology table. If a read is mapped to one of the duplication copies without gaps, read alignment to a different copy can be easily inferred using the sequence alignment between duplication copies (stored in the homology table). Otherwise, we merge the read alignment with the duplication sequence alignment in order to obtain a set of read positions matching second copy positions. Next, we fill the gaps in the alignment using Needleman-Wunsch algorithm [33]. Additionally, if the original read alignment contained soft clipping, we perform semi-global alignment [34] to check if the ends of the read can be aligned to the second copy. This procedure ensures that the vast majority of reads can be re-mapped without performing a full realignment of the read to the region R.

B.2.4 Finding aggregate copy number profiles

For a region group with reference copy number c_r we calculate aggregate copy number profiles using a matrix of aggregate read depth observations $\{o_w^{(s)}\}$ for all windows $w \in W$ and samples $s \in S$.

B.2.4.1 Estimating number of *AggregateCN* states

Average normalized aggregate read depth for a sample is calculated as

$$\overline{o}^{(s)} = \frac{1}{|W|} \sum_{w \in W} \frac{o_w^{(s)} \cdot 2 \cdot p_w^{(s)}}{n_w^{(s)} \cdot \left(1 - p_w^{(s)}\right)},$$

where $p_w^{(s)}$ and $n_w^{(s)}$ are Negative Binomial (NB) parameters for sample s and window w. Next, we select the minimum and maximum *AggregateCN* values:

$$c_1 = \max\left\{0,\ c_r - 2 \cdot B_l,\ \min\left\{c_r - B_l,\ \min_s \left\lfloor \overline{o}^{(s)} \right\rfloor - 1\right\}\right\},$$

$$c_K = \min\left\{c_r + 2 \cdot B_r,\ \max\left\{c_r + B_r,\ \max_s \left\lceil \overline{o}^{(s)} \right\rceil + 1\right\}\right\},$$

where B_l and B_r are bounds on how much copy number can differ from the reference copy number c_r, and are equal 5 and 7 by default, respectively. Range of *AggregateCN* values from c_1 to c_K will be used as a set of hidden states in *AggregateCN* Hidden Markov Model.

B.2.4.2 HMM definition

We define a homogeneous discrete-time HMM [127] for generating the read depth in T windows across a region with reference copy number c_r as follows:

1. For all windows in the region, we define a set of hidden states C, $|C| = K$ corresponding to K possible *AggregateCN* states. We denote hidden state of a sample s at window w as $Z_w^{(s)}$.

2. The initial state distribution $\pi_c = \max\left\{\sqrt{t}, \frac{1}{|S|}\right\}$ for *AggregateCN* $c \neq c_r$ and $\pi_{c_r} = 1 - \sum_{c \neq c_r} \pi_c$. Here and later we use an input parameter t, which is equal to 10^{-5} by default.

3. Transition parameters $\tilde{a}_{\nearrow w}$ and $\tilde{a}_{\searrow w}$ define all possible transitions (for fixed i and w):

$$a_{ijw} = \begin{cases} \tilde{a}_{\nearrow w}^{j-i} & \text{if } j > i, \\ \tilde{a}_{\searrow w}^{i-j} & \text{if } j < i, \\ 1 - \sum_{j \neq i} a_{ijw} & \text{if } j = i. \end{cases}$$

   On the first iteration $\tilde{a}_{\nearrow w} = \tilde{a}_{\searrow w} = t$ for all windows w. By default, we limit the maximal *AggregateCN* jump between two consecutive windows and set $a_{ijw} = 0$ if $|i - j| > 6$.

4. The emission probabilities are defined for each sample s using Negative Binomial parameters $n_w^{(s)}$ and $p_w^{(s)}$ corresponding to the sample s and GC-content of the window w. Using the fact that sum of NB-distributed random variables with parameters $(n_1,\ p)$ and

$(n_2,\ p)$ is a NB random variable with parameters $(n_1 + n_2,\ p)$, we can multiply parameter n by the copy number in order to calculate emission probability of a certain hidden state. Therefore, we calculate emission probability of copy number c at window w as

$$b_w^{(s)}(c) = P_{\text{NB}}\left(o_w^{(s)};\ m_w \cdot n_w^{(s)} \cdot c/2,\ p_w^{(s)}\right)^{\psi(m_w)}$$

where $o_w^{(s)}$ is the observed aggregate read depth. We divide c by 2 as background read depth was calculated for regions with copy number 2. For copy number zero we use $c = 0.01$ to allow possible errorneous read alignments. m_w is a scale parameter and $\psi\,(m_w)$ is the scale parameter weight, both are equal to one on the first iteration.

B.2.4.3 Updating emission probabilities using scale parameters

After the first iteration we introduce non-trivial multipliers m_w for each window w. This scale parameter is used to remove window-specific sequencing bias that is shared across all samples. Windows with large bias should contribute less to the likelihood, so we assign a weight to the scale parameter based on the distance $|m_w - 1|$ ($m_w = 1$ represents no significant bias). For a region with reference copy number c_r we expect that all scale parameters should be within $(\frac{c_r-1}{c_r},\ \frac{c_r+1}{c_r})$, so scale parameters outside these bounds are assigned weight 0. For scale parameters within the bounds we distribute weights according to the tricube kernel [155]:

$$\psi(m_w) = \left(1 - \min(1, |m_w - 1| \cdot c_r)^{\,3}\,\right)^3.$$

On each HMM iteration we run the Forward-Backward algorithm [128] to obtain a range of matrices $\gamma_{c,w}^{(s)} = P\left(Z_w^{(s)} = c \,\middle|\, o_{1:T}^{(s)}\right)$ — probability of sample s having copy number c at window w. We use γ to update the scale parameters:

$$m_w \leftarrow \arg\max_m \prod_{s \in S} \sum_{c \in C} \gamma_{c,w}^{(s)} \cdot P_{\text{NB}}\left(o_w^{(s)};\ m_w \cdot n_w^{(s)} \cdot c/2,\ p_w^{(s)}\right),$$

where S is the set of all samples and C is the set of hidden states (and copy number values they represent).

B.2.4.4 Updating initial and transition probabilities

In addition to probabilities matrices γ, the Forward-Backward algorithm provides matrices $\alpha_{c,w}^{(s)} = P\left(o_{1:w}^{(s)}, Z_w^{(s)} = c\right)$ and $\beta_{c,w}^{(s)} = P\left(o_{w+1:T}^{(s)} \middle| Z_w^{(s)} = c\right)$ — forward and backward probabilities, respectively. Then, total probability of a sample s is $\tau(s) = P\left(o_{1:T}^{(s)}\right) = \sum_{c \in C} \alpha_{c,T}^{(s)}$. According to the Baum-Welch algorithm [127], we update initial probabilities as

$$\pi_c \leftarrow \frac{1}{|S|} \sum_{s \in S} \gamma_{c,1}^{(s)}.$$

Additionally, we bound $\pi_c \leftarrow \max\left\{\pi_c, \sqrt{t}, \frac{1}{|S|}\right\}$.

Let $\xi_{w,i,j}^{(s)}$ denote the probability of going from state i at window w to state j at window $w+1$ at the sample s. It can be calculated as

$$\xi_{ijw}^{(s)} = \frac{1}{\tau(s)} \cdot \alpha_{i,w}^{(s)} \cdot a_{ijw} \cdot \beta_{j,w+1}^{(s)} \cdot b_{w+1}^{(s)}(j),$$

where a_{ijw} denotes transition probability between hidden states i and j at windows w and $w+1$. Similarly, we can calculate the probability of increasing or decreasing copy number:

$$\xi_{\nearrow w}^{(s)} = \sum_{i \in C,\ j \in C,\ j > i} \xi_{ijw}^{(s)},$$

$$\xi_{\searrow w}^{(s)} = \sum_{i \in C,\ j \in C,\ j < i} \xi_{ijw}^{(s)}.$$

Next, we average these values across all samples to get probabilities of increasing or decreasing aggregate copy number:

$$\tilde{a}_{\nearrow w} = \frac{1}{|S|} \sum_{s \in S} \xi^{(s)}_{\nearrow w}\,, \qquad \tilde{a}_{\searrow w} = \frac{1}{|S|} \sum_{s \in S} \xi^{(s)}_{\searrow w}.$$

B.2.4.5 Speeding up HMM convergence

In some cases, the iterative HMM converges slowly. Therefore, to speed up the conversion we search for peaks in $\tilde{a}_{\nearrow w}$ and $\tilde{a}_{\searrow w}$. We define peaks as local maxima higher than t and higher than any other values in the 10 window neighbourhood to the left and right. For each peak we set $\tilde{a}$ to the sum of $\tilde{a}$ over the neighbourhood of the peak, and decrease $\tilde{a}$ values in the neighbourhood to t. Next, we bound $\tilde{a}_{\nearrow w}$ and $\tilde{a}_{\searrow w}$ to be at least t and at most 0.1, and set new transition probabilities a_{ijw} based on $\tilde{a}_{\nearrow w}$ and $\tilde{a}_{\searrow w}$ as described above.

B.2.4.6 Log-likelihood convergence

Similar to most HMM applications, we aim to repeat HMM parameter refinement until the log-likelihood $\mathscr{L} = \sum_{s \in S} \log \tau(s)$ stops increasing. However, the emission probabilities definition includes the scale parameter weight exponent $\psi(m_w)$, which equals to one for all windows on the first iteration and can be lower on the subsequent iterations. This can lead to a drop in log-likelihood after the first iteration. Therefore, we allow the log-likelihood to decrease between the first and second iteration, and stop only after third iteration if the increase in the log-likelihood is less than 0.01.

B.2.4.7 Aggregate copy number quality

To assign a quality to each *AggregateCN* estimate, we calculate a probability of the prediction and probabilities of alternative predictions. Suppose, the Viterbi path [129] for a sample s contains a constant stretch between windows u and v with $Z^{(s)}_{u:v} = c$. We calculate the probability of such a stretch using forward and backward probability matrices $\alpha^{(s)}$ and $\beta^{(s)}$:

$$P\left(Z^{(s)}_{u:v} = c \,\middle|\, o^{(s)}_{1:T}\right) = \frac{1}{\tau(s)} \cdot \alpha^{(s)}_{c,u} \cdot \beta^{(s)}_{c,v} \cdot \prod_{k=u}^{v-1} a_{cck} \cdot b_{k+1}(c).$$

Next, we calculate probabilities of the alternative non-overlapping paths to calculate the probability of an error:

$$P_{\text{error}} = 1 - \frac{P\left(Z^{(s)}_{u:v} = c \,\middle|\, o^{(s)}_{1:T}\right)}{\sum_{c' \in C} P\left(Z^{(s)}_{u:v} = c' \,\middle|\, o^{(s)}_{1:T}\right)}.$$

Finally, *AggregateCN* prediction quality is calculated as a Phred [156] quality score: $-10 \cdot \log_{10} P_{\text{error}}$.

B.2.5 Estimating paralog-specific copy number using PSVs

For a sample s and PSV v we observe allele counts X_{sv}, which form the observed data X. Note that X_{sv} is a tuple as it stores read counts for two or more alleles of the PSV. We can easily calculate probability $P(X_{sv} \mid \widehat{G}_v = \hat{g})$ of allele counts given the PSV genotype $\hat{g}$ using multinomial distribution [157, 158].

In a region with reference copy number c_r, each PSV v is assigned a vector $f_v \in [0, 1]^{c_r/2}$, where f_{vk} represents the frequency of the reference allele on k-th copy of the duplication across the whole population (there are total $c_r/2$ copies). For example, a PSV has reference allele A on the first copy and C on the second copy, but all reads in all samples support C on both copies. In that case $f_{v,1} = 0$ and $f_{v,2} = 1$ as the frequency of the reference allele is 0 on the first copy.

To calculate the probability of observing read counts X_{sv} in case of the sample paralog-specific copy number *(ParalogCN)* g:

$$P(X_{sv} \mid G_s = g, f_v) = \sum_{\hat{g}} P(X_{sv} \mid \widehat{G}_v = \hat{g}) \cdot P(\widehat{G}_v = \hat{g} \mid G_s = g, f_v).$$

and to calculate probability $P(\widehat{G}_v = \hat{g} \mid G_s = g, f_v)$ of observing PSV genotype $\hat{g}$ given the *ParalogCN* g and frequencies f_v we need to calculate coefficients in a multivariate polynomial. For example, in a 2-copy duplication ($c_r = 4$) PSV v has two values $f_1 = f_{v,1}$ and $f_2 = f_{v,2}$. Then, if sample paralog-specific copy number $g = 2{,}2$ (each copy is represented twice) we need to expand a polynomial

$$\big(f_1 \cdot u_1 + (1 - f_1) \cdot u_2\big)^2 \cdot \big((1 - f_2) \cdot u_1 + f_2 \cdot u_2\big)^2.$$

Then the probability $P(\widehat{G}_v = \hat{g} \mid G_s = g, f_v)$ of PSV genotype $\hat{g}$ is a coefficient in front of $u_1^{\hat{g}_1} \cdot u_2^{\hat{g}_2}$. For example, probability of a PSV genotype $\hat{g} = 4{,}0$ would be $f_1^2 \cdot (1 - f_2)^2$. This can be generalized for any PSV genotype, any paralog-specific, aggregate and reference copy numbers. Additionally, we use f as variables (instead of numeric values) to expand the multivariate polynomial (with variables $u_{\dots}$ and $f_{\dots}$) in advance, which significantly speeds up PSV genotype probability calculation.

B.2.5.1 EM algorithm

Next, we define the total likelihood of the model $L(X, f) = \prod_{s \in S} P(X_s \mid f)$ as the product of probabilities for all samples. Probability of a single sample is $P(X_s \mid f) = \sum_g P(X_s, G_s = g \mid f)$, and probability of a sample and a *ParalogCN* g is

$$P(X_s, G_s = g \mid f) = p(g) \prod_{v \in V} P(X_{sv} \mid G_s = g, f_v),$$

where V is a total set of PSVs and $p(g)$ is a paralog-specific copy number prior. We define *ParalogCN* priors based on the distance to the reference *ParalogCN* (all copies are represented twice), with the smallest prior = 10^{-6} assigned to the most extreme *ParalogCN*s. For example, for a two-copy duplication *ParalogCN* 4,0 is assigned a prior 10^{-6}, *ParalogCN* 3,1 is assigned a prior 10^{-3} and *ParalogCN* 2,2 is set to 1 minus all other priors.

According to the Expectation-Maximization algorithm [159], on the **E-step** we find the distribution of the hidden variables (paralog-specific copy numbers):

$$P(G_s = g \mid X_s, f) = \frac{P(X_s, G_s = g \mid f)}{\sum_{g'} P(X_s, G_s = g' \mid f)}.$$

We will denote $P(G_s = g \mid X_s, f)$ as $\zeta_{s,g}$.

Next, during the **M-step** we maximize log-likelihood:

$$f^\star \leftarrow \arg\max_f \; p(f) \cdot \sum_{s \in S} \zeta_{s,g} \cdot \log \frac{P(G_s = g \mid X_s, f)}{\zeta_{s,g}}.$$

This step can be solved numerically and independently for all PSVs. We select prior $p(f)$ in a way to encourage higher values of f:

$$p(f) = \prod_{v \in V} P_{\text{Beta}} \left(x \in \left[\max_k(f_{vk}) - 10^{-6},\; \max_k(f_{vk}) \right];\; \alpha = 5,\; \beta = 1 \right).$$

After the EM algorithm converges, we use frequency matrix f to select a set of reliable PSVs $\left\{ v \,\middle|\, \min_{k=1}^{c_r/2} f_{vk} \geq 0.95 \right\}$. We use reliable PSVs to calculate probability of all *ParalogCN* values for all samples, including samples with non-reference *AggregateCN*. Next, for each duplication copy we calculate marginal probability over all *ParalogCN* probabilities:

$$P([G_s]_k = c \mid X_s, f) = \sum_{g \text{ s.t. } g_k = c} P(G_s = g \mid X_s, f).$$

This allows us to calculate most likely *ParalogCN* value $c = \arg\max_{c'} P([G_s]_k = c' \mid X_s, f)$ for each duplication copy, and to calculate the corresponding Phred [156] quality score $-10 \cdot \log_{10} P([G_s]_k \neq c \mid X_s, f)$. Additionally, we do not assign copy number to some duplication copies if the quality is near zero (< 5). For example a sample in three-copy duplication can have a paralog-specific copy number (2,?,?), when we do not have enough information to distinguish between the second and the third copies.

B.2.5.2 Information content of the PSVs

In certain cases, the EM algorithm may converge to an undesirable solution. For example, suppose that there is a set of unreliable PSVs V_{unrel} that exhibit PSV genotype 4,0 in all samples (all reads support an allele corresponding to the first copy of the duplication). Additionally, there is a smaller set of reliable PSVs V_{rel} that exhibit PSV genotype 2,2 (alleles from both copies are present with equal proportions). This situation occurs in the two-copy duplication that includes genes *SERF1A* and *SMN1*. Depending on the relative sizes of PSV sets V_{unrel} and V_{rel}, the EM algorithm can converge to two possible solutions:

- Assign $f_{v,1} \simeq 1$ for all $v \in V_{\text{unrel}}$ and predict $\textit{ParalogCN} = 4, 0$ for all samples. In that case the EM algorithm would assign $f_{v',1} \simeq 1/2$ for all $v' \in V_{\text{rel}}$. Note that $f_{v,2}$ can be anything for both PSV sets, as there are no samples that have a second copy. This solution can be explained in the following way: for every sample there are four haplotypes, all of which are more similar to the first copy of the duplication than to the second, therefore it would be correct to set $\textit{ParalogCN} = 4, 0$ for all samples.

- Assign $f_{v,2} = 0$ for all $v \in V_{\text{unrel}}$ (frequency of the reference allele on the second copy is 0). In this solution $f_{v,1} \simeq 1$ for $v \in V_{\text{unrel}}$ and $f_{v',1} \simeq f_{v',2} \simeq 1$ for $v' \in V_{\text{rel}}$; *ParalogCN* prediction for all samples would be 2,2. This solution is more appropriate in the following way: even though four haplotypes of each sample are more similar to the first copy, two of them are significantly different from the other two (at reliable PSVs V_{rel}), and share many similarities with the second copy of the duplication. Therefore this solution provides more information.

To encourage the EM algorithm to converge to the second solution we add a weight to each PSV based on multiple samples and call it *information content* of the PSV. Suppose a PSV v has n alleles, then

$$I(v) = \frac{1}{|S|} \sum_{s \in S} \sum_{\hat{g}} P(\widehat{G}_v = \hat{g} \mid X_{sv}) \cdot \frac{\sum_{k=1}^{n} \mathbb{1}\,[\hat{g}_k \neq 0] - 1}{n - 1}$$

The fraction on the right represents how well PSV alleles are represented in the PSV genotype. For example, PSV genotypes of a PSV with three alleles would have the following weights: 1 for genotypes without zeros, such as (2,2,2), (3,2,1); 1/2 for genotypes with one zero, such as (4,2,0), (3,3,0); 0 for genotypes with two zeros, such as (6,0,0), (0,6,0) and (0,0,6). This way, if some allele of a PSV is consistently missing in many samples — the PSV would get low information content I.

During the E-step (calculating sample *ParalogCN* probabilities) we use the PSV information content as an exponent:

$$P(X_s, G_s = g \mid f) = p(g) \prod_{v \in V} P^{\,I(v)}(X_{sv} \mid G_s = g, f_v),$$

This forces PSVs with information content close to 0 have a very small effect on the paralog-specific copy number calculation and on the total likelihood.

B.2.5.3 Selecting starting states for EM algorithm

A single read can cover several PSVs if they are close enough. Therefore, we filter out PSVs if they are closer than 100 bp to each other. While discarding neighboring PSVs we first remove PSVs with low information content, and then remove PSVs that represent insertions or deletions.

The EM algorithm is not guaranteed to reach the global maximum and can be trapped at local maxima. A standard approach to avoid local maxima is to use several starting solutions. We cluster the PSVs to obtain several starting positions for the EM algorithm as follows: for each PSV v we have a vector of allele counts $\{X_{sv}\}_{s \in S}$, which we then transform into a numeric vector by calculating the fraction of the allele corresponding to the first copy: $\{[X_{sv}]_1 / \sum X_{sv}\}_{s \in S}$. Next, we construct a Pearson correlation matrix over PSVs and split it into two clusters based on the hierarchical clustering [160]. Then we use three starting PSV sets: the two separate clusters and all PSVs together. We start by assigning $f_{v,\cdot} = 0.9$ to PSVs in the starting set and $f_{v,\cdot} = 0.5$ to all other PSVs. Next, we iteratively run E- and M-steps until the algorithm converges and finally select the result from the starting set that produced the highest total likelihood.

B.2.6 Extending homology table to include an additional repeat copy for *OTOA*

Analysis of the two-copy *OTOA* locus on chromosome 16 using Parascopy showed that majority of the samples have three duplication copies. More than 75% and 91% samples in the European and African continental populations had *AggregateCN* = 6. Alignment of the *OTOA* and *OTOAP1* (pseudo-gene) sequences to the recently generated high-quality genome assembly for a human cell line, CHM13 [1] using Minimap2 [38] generated three independent hits for both sequences. We denote the hit that is least similar to *OTOA* and *OTOAP1* as *OTOAP∗* and add it to the homology table. It is not required to create the whole homology table anew to do this — we find sequence homologies between the *OTOAP∗* sequence and hg38 reference genome and add them to the homology table. This allows us to find reliable PSVs that distinguish *OTOA* and *OTOAP∗* and detect paralog-specific copy number of an extended three-copy duplication.

Next, to analyze the new three-copy duplication, we can use the standard aligment files mapped to the hg38 reference genome, which contain reads mapped to *OTOA* and *OTOAP1*, but not to *OTOAP*∗*. Since Parascopy first re-maps reads from different repeat copies to a single copy, it is not important to have correct read alignments to all three copies.

B.2.7 Subsampling reads

In order to estimate the impact of read depth on the accuracy of Parascopy, we artificially reduced coverage for 107 samples from the Iberian population (IBS) from the 1kGP WGS data. Since the 1kGP samples were sequenced to an average read depth of 33×, we randomly and independently selected read pairs with probabilities $1/3$ and $2/3$ to create subsampled datasets with average read depth 11× and 22×.

B.2.8 Paralog-specific copy number validation using trios

To determine trio concordance for *ParalogCN* values, it is useful to model the *ParalogCN* for each homologous chromosome or haplotype. Suppose an individual has c copies of a repeat copy R ($\mathit{ParalogCN}_R = c$), then one of the homologous chromosome has $a \in [0, c]$ copies of R, while the other homologous chromosome would have $b = c - a$ copies of R. In a sample that is concordant with the reference, repeat copy R appears once on each homologous chromosome, i.e. $c = 2$ and $a = b = 1$. Here (a, b) is the *diploid ParalogCN* at the repeat copy R for the individual.

Suppose the diploid *ParalogCN* at the repeat copy R for two parents is $(a_m,\ b_m)$ and $(a_f,\ b_f)$. Then the four possible diploid *ParalogCN* values for a child are: $(a_m,\ a_f)$, $(a_m,\ b_f)$, $(b_m,\ a_f)$ and $(b_m,\ b_f)$. Note that this does not model situations when the different copies of R lie on several non-homologous chromosomes or lie far from each other on the same homogolous chromosome, in both cases a child can receive a different combination of copies of R.

In a sample set with n individuals, let n_c be the number of samples with c copies of R (summed over two homologous chromosomes) and let $c_{\max} = \arg\max_c n_c$ be the maximal observed copy number. Suppose vector ϕ stores the allele frequency distribution of copy number values for a single homologous chromosome in the population. If the two homologous chromosomes are independent, the expected number of samples with *ParalogCN*$_R = c$ would be $e_c = \sum_{a=0}^{c} n \cdot \phi_a \cdot \phi_{c-a}$. We calculate the likelihood of the copy number frequencies ϕ using chi-square distribution with $c_{\max}$ degrees of freedom and

$$\chi^2 = \sum_{c=0}^{c_{\max}} \frac{(e_c - n_c)^2}{e_c}.$$

Next, we use maximum likelihood to estimate the most-likely frequencies ϕ according to the observed diploid *ParalogCN*$_R$ counts in each continental population. On all steps, only samples with high-quality ($\geq$ 20) *ParalogCN*$_R$ estimates were used.

Throughout 5 continental populations, 167 duplicated loci and a total of 384 paralogs (1920 total entries), we estimated copy number frequencies in 1474 cases with at least 10 samples. In only 20 cases χ^2 p-value was under 0.05, and in only 4 cases (0.3%) p-values were under 0.05 after the Benjamini-Hochberg correction [161] (controlling false-discovery rate). This shows that (i) probabilistic model is consistent with *ParalogCN* estimates; (ii) Parascopy *ParalogCN* values do not violate Hardy–Weinberg equilibrium in the vast majority of the cases.

Using the distribution ϕ_c, we can calculate the probability of observing a child with c copies of R when the two parents have c_m and c_f copies:

$$\begin{aligned} p = P(c \mid c_m, c_f) = &\sum_{\substack{a_m + b_m = c_m \\ a_m,\, b_m \geq 0}} P\left(ParalogCN_R = (a_m,\ b_m)\right) \times \\ &\sum_{\substack{a_f + b_f = c_f \\ a_f,\, b_f \geq 0}} P\left(ParalogCN_R = (a_f,\ b_f)\right) \times \\ &\sum_{\substack{x \in \{a_m,\, b_m\} \\ y \in \{a_f,\, b_f\}}} \frac{1}{4} \cdot \mathbb{1}[x + y = c]. \end{aligned}$$

The probability $P\big(ParalogCN_R = (a,\ b)\big)$ of having diploid *ParalogCN* at the repeat copy R can be calculated as:

$$P\big(ParalogCN_R = (a,\ b)\big) = \frac{\phi_a \cdot \phi_b}{\sum_{a'+b'=a+b} \phi_{a'} \cdot \phi_{b'}}.$$

A low value of probability p implies that the child's *ParalogCN* estimate is not consistent with the parental *ParalogCN* values. We use a default threshold of $p < 0.01$ to identify discordant trios. Stricter thresholds (such as $p < 0.05$ and $p < 0.1$) produce similar results.

APPENDIX C

Supplementary Material for Chapter 4

C.1 Supplementary Methods

C.1.1 Hashing sequencing reads

On average, a length of the sequencing read name varies between 30 and 40 symbols, and, according to the BAM/CRAM file specification, can be as long as 256 symbols. In order to reduce the volume of stored and transferred information, we transform the read name into an 8-byte integer using the Fowler–Noll–Vo hash function (FNV). Specifically, we use the 64-bit flavor of the FNV-1 version of the hash function. Finally, we multiply the hash integer by two and replace the last bit with read-pair information (first or second read mate in the pair). This allows us to (i) store and transfer the ordinal read pair number together with the read hash and (ii) quickly determine the hash of the corresponding read mate.

C.1.2 Identifying possible locations for sequencing reads in the duplicated regions

In order to identify possible sequencing read locations, we utilize the overlaps between reads and PSVs. Although it is possible to use the homology table to find the possible read locations, this approach is more time-consuming, as it requires several table fetch operations for a large amount of reads.

As every PSVs contains information about its homologous positions, finding possible read locations is a relatively straightforward operation. We assume that the analyzed reads are short enough not to overlap the same PSV several times, and greedily search for clusters of PSV homologous coordinates, such that the total cluster length does not exceed the read length by more than 10 bp. Finally, we extend each cluster by the read length to the left and to the right.

In certain cases, we can either discard one of the possible locations, or be certain that the original location is correct. Consider a read with an *original* alignment to one of the repeat copies, that was remapped to another repeat copy to get a *pooled* alignment. We say that an alignment has unique tail if it contains at least 15 bp that do not overlap any entry in the homology table. If the original alignment has high mapping quality (≥ 50), has a unique tail, and has the same or fewer clipped basepairs than the pooled alignment (or if the original alignment matches the pooled alignment and has no clipping at all) — we say that the original alignment location is certainly correct. Finally, if the original alignment is much better than the pooled alignment (aligned length is at least 15 bp more) — we say that the pooled alignment is certainly *in*correct. We cannot easily confirm that the original location is correct, as it is possible that there exists a third repeat copy that the read can map to, and checking for such cases would significantly hamper the execution time.

In certain cases, out of two read mates, only one has a confirmed location or a set of possible locations, while the other read mate does not overlap any PSVs and does not have a unique tail. For such read pairs, we extend all possible first read locations to both sides by the insert length, and keep PSV-based location probabilities derived from the first read mate.

Bibliography

1. Nurk, S. *et al.* The complete sequence of a human genome. *Science* **376,** 44–53 (2022).

2. 1000 Genomes Project Consortium *et al.* A global reference for human genetic variation. *Nature* **526,** 68 (2015).

3. Bailey, J. A., Yavor, A. M., Massa, H. F., Trask, B. J. & Eichler, E. E. Segmental duplications: organization and impact within the current human genome project assembly. *Genome Res.* **11,** 1005–1017 (2001).

4. Samonte, R. V. & Eichler, E. E. Segmental duplications and the evolution of the primate genome. *Nat. Rev. Genet.* **3,** 65–72 (2002).

5. Sharp, A. J. *et al.* Segmental duplications and copy-number variation in the human genome. *Am. J. Hum. Genet.* **77,** 78–88 (2005).

6. Karolchik, D. *et al.* The UCSC Table Browser data retrieval tool. *Nucleic Acids Res.* **32,** D493–D496 (2004).

7. Pang, A. W. *et al.* Towards a comprehensive structural variation map of an individual human genome. *Genome Biol.* **11,** 1–14 (2010).

8. Mahmoud, M. *et al.* Structural variant calling: the long and the short of it. *Genome Biol.* **20,** 1–14 (2019).

9. Stankiewicz, P. & Lupski, J. R. Genome architecture, rearrangements and genomic disorders. *Trends Genet.* **18,** 74–82 (2002).

10. Chen, X. *et al.* Spinal muscular atrophy diagnosis and carrier screening from genome sequencing data. *Genet. Med.* **22,** 945–953 (2020).

11. Sanger, F., Nicklen, S. & Coulson, A. R. DNA sequencing with chain-terminating inhibitors. *P. Natl. Acad. Sci. USA* **74,** 5463–5467 (1977).

12. International Human Genome Sequencing Consortium. Initial sequencing and analysis of the human genome. *Nature* **409,** 860–921 (2001).

13. Green, R. E. *et al.* Analysis of one million base pairs of Neanderthal DNA. *Nature* **444,** 330–336 (2006).

14. Heather, J. M. & Chain, B. The sequence of sequencers: The history of sequencing DNA. *Genomics* **107,** 1–8 (2016).

15. Schadt, E. E., Turner, S. & Kasarskis, A. A window into third-generation sequencing. *Hum. Mol. Genet.* **19,** R227–R240 (2010).

16. Van Dijk, E. L., Auger, H., Jaszczyszyn, Y. & Thermes, C. Ten years of next-generation sequencing technology. *Trends Genet.* **30,** 418–426 (2014).

17. Clarke, J. *et al.* Continuous base identification for single-molecule nanopore DNA sequencing. *Nat. Nanotechnol.* **4,** 265–270 (2009).

18. Delahaye, C. & Nicolas, J. Sequencing DNA with nanopores: troubles and biases. *PLoS One* **16,** e0257521 (2021).

19. Jain, M. *et al.* Nanopore sequencing and assembly of a human genome with ultra-long reads. *Nat. Biotechnol.* **36,** 338–345 (2018).

20. Hon, T. *et al.* Highly accurate long-read HiFi sequencing data for five complex genomes. *Sci. Data* **7,** 1–11 (2020).

21. Spies, N. *et al.* Genome-wide reconstruction of complex structural variants using read clouds. *Nat. Methods* **14,** 915–920 (2017).

22. Elyanow, R., Wu, H.-T. & Raphael, B. J. Identifying structural variants using linked-read sequencing data. *Bioinformatics* **34,** 353–360 (2018).

23. Zhang, L., Zhou, X., Weng, Z. & Sidow, A. Assessment of human diploid genome assembly with 10x Linked-Reads data. *GigaScience* **8,** giz141 (2019).

24. Knuth, D. E., Morris Jr, J. H. & Pratt, V. R. Fast pattern matching in strings. *SIAM J. Comput.* **6,** 323–350 (1977).

25. Apostolico, A., Iliopoulos, C., Landau, G. M., Schieber, B. & Vishkin, U. Parallel construction of a suffix tree with applications. *Algorithmica* **3,** 347–365 (1988).

26. Manber, U. & Myers, G. Suffix arrays: a new method for on-line string searches. *SIAM J. Comput.* **22,** 935–948 (1993).

27. Burrows, M. & Wheeler, D. J. A block-sorting lossless data compression algorithm. in *Technical Report 124*, 1–18 (Systems Research Center, Palo Alto, CA, 1994).

28. Ferragina, P. & Manzini, G. Opportunistic data structures with applications. in *Proceedings of the 41st Annual Symposium on Foundations of Computer Science*, 390–398 (2000).

29. Langmead, B., Trapnell, C., Pop, M. & Salzberg, S. L. Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biol.* **10,** 1–10 (2009).

30. Langmead, B. & Salzberg, S. L. Fast gapped-read alignment with Bowtie 2. *Nat. Methods* **9,** 357–359 (2012).

31. Li, H. & Durbin, R. Fast and accurate short read alignment with Burrows–Wheeler transform. *Bioinformatics* **25,** 1754–1760 (2009).

32. Li, H. Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM. Preprint at https://doi.org/10.48550/arXiv.1303.3997 (2013).

33. Needleman, S. B. & Wunsch, C. D. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.* **48,** 443–453 (1970).

34. Smith, T. F., Waterman, M. S., *et al.* Identification of common molecular subsequences. *J. Mol. Biol.* **147,** 195–197 (1981).

35. Knudsen, B. & Miyamoto, M. M. Sequence alignments and pair hidden Markov models using evolutionary history. *J. Mol. Biol.* **333,** 453–460 (2003).

36. Farrar, M. Striped Smith–Waterman speeds database searches six times over other SIMD implementations. *Bioinformatics* **23,** 156–161 (2007).

37. Marco-Sola, S., Moure, J. C., Moreto, M. & Espinosa, A. Fast gap-affine pairwise alignment using the wavefront algorithm. *Bioinformatics* **37,** 456–463 (2021).

38. Li, H. Minimap2: pairwise alignment for nucleotide sequences. *Bioinformatics* **34,** 3094–3100 (2018).

39. Li, H. New strategies to improve Minimap2 alignment accuracy. *Bioinformatics* **37,** 4572–4574 (2021).

40. Garrison, E. & Marth, G. Haplotype-based variant detection from short-read sequencing. Preprint at https://doi.org/10.48550/arXiv.1207.3907 (2012).

41. Poplin, R. *et al.* Scaling accurate genetic variant discovery to tens of thousands of samples. Preprint at https://doi.org/10.1101/201178 (2018).

42. Edge, P. & Bansal, V. Longshot enables accurate variant calling in diploid genomes from single-molecule long read sequencing. *Nat. Commun.* **10,** 4660 (2019).

43. Poplin, R. *et al.* A universal SNP and small-indel variant caller using deep neural networks. *Nat. Biotechnol.* **36,** 983–987 (2018).

44. Abyzov, A., Urban, A. E., Snyder, M. & Gerstein, M. CNVnator: an approach to discover, genotype, and characterize typical and atypical CNVs from family and population genome sequencing. *Genome Res.* **21,** 974–984 (2011).

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

46. Shen, F. & Kidd, J. M. Rapid, paralog-sensitive CNV analysis of 2457 human genomes using QuicK-mer2. *Genes* **11,** 141 (2020).

47. Handsaker, R. E. *et al.* Large multiallelic copy number variations in humans. *Nat. Genet.* **47,** 296–303 (2015).

48. Treangen, T. J. & Salzberg, S. L. Repetitive DNA and next-generation sequencing: computational challenges and solutions. *Nat. Rev. Genet.* **13,** 36–46 (2012).

49. Bailey, J. A. *et al.* Recent segmental duplications in the human genome. *Science* **297,** 1003–1007 (2002).

50. Mandelker, D. *et al.* Navigating highly homologous genes in a molecular diagnostic setting: a resource for clinical next-generation sequencing. *Genet. Med.* **18,** 1282–1289 (2016).

51. Clendenning, M. *et al.* A frame-shift mutation of PMS2 is a widespread cause of Lynch syndrome. *J. Med. Genet.* **45,** 340–345 (2008).

52. Mandelker, D. *et al.* Comprehensive diagnostic testing for stereocilin: an approach for analyzing medically important genes with high homology. *J. Mol. Diagn.* **16,** 639–647 (2014).

53. Zhao, J. *et al.* A missense variant in NCF1 is associated with susceptibility to multiple autoimmune diseases. *Nat. Genet.* **49,** 433–437 (2017).

54. Ebbert, M. T. *et al.* Systematic analysis of dark and camouflaged genes reveals disease-relevant genes hiding in plain sight. *Genome Biol.* **20,** 1–23 (2019).

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

56. Tyson, J. R. *et al.* MinION-based long-read sequencing and assembly extends the *Caenorhabditis elegans* reference genome. *Genome Res.* **28,** 266–274 (2018).

57. Audano, P. A. *et al.* Characterizing the major structural variant alleles of the human genome. *Cell* **176,** 663–675 (2019).

58. Shafin, K. *et al.* Nanopore sequencing and the Shasta toolkit enable efficient de novo assembly of eleven human genomes. *Nat. Biotechnol.* **38,** 1044–1053 (2020).

59. Wenger, A. M. *et al.* Accurate circular consensus long-read sequencing improves variant detection and assembly of a human genome. *Nat. Biotechnol.* **37,** 1155–1162 (2019).

60. Pendleton, M. *et al.* Assembly and diploid architecture of an individual human genome via single-molecule technologies. *Nat. Methods* **12,** 780–786 (2015).

61. Huddleston, J. *et al.* Discovery and genotyping of structural variation from long-read haploid genome sequence data. *Genome Res.* **27,** 677–685 (2017).

62. Ebler, J., Haukness, M., Pesout, T., Marschall, T. & Paten, B. Haplotype-aware diplotyping from noisy long reads. *Genome Biol.* **20,** 116 (2019).

63. Luo, R., Sedlazeck, F. J., Lam, T.-W. & Schatz, M. C. A multi-task convolutional deep neural network for variant calling in single molecule sequencing. *Nat. Commun.* **10,** 998 (2019).

64. Sedlazeck, F. J. *et al.* Accurate detection of complex structural variations using single-molecule sequencing. *Nat. Methods* **15,** 461–468 (2018).

65. Mikheenko, A., Bzikadze, A. V., Gurevich, A., Miga, K. H. & Pevzner, P. A. TandemTools: mapping long reads and assessing/improving assembly quality in extra-long tandem repeats. *Bioinformatics* **36,** i75–i83 (2020).

66. Jain, C. *et al.* Weighted minimizer sampling improves long read mapping. *Bioinformatics* **36,** i111–i118 (2020).

67. Miga, K. H. *et al.* Telomere-to-telomere assembly of a complete human X chromosome. *Nature* **585,** 79–84 (2020).

68. Sudmant, P. H. *et al.* Diversity of human copy number variation and multicopy genes. *Science* **330,** 641–646 (2010).

69. Vollger, M. R. *et al.* Long-read sequence and assembly of segmental duplications. *Nat. Methods* **16,** 88–94 (2019).

70. Fredman, D. *et al.* Complex SNP-related sequence variation in segmental genome duplications. *Nat. Genet.* **36,** 861–866 (2004).

71. Estivill, X. *et al.* Chromosomal regions containing high-density and ambiguously mapped putative single nucleotide polymorphisms (SNPs) correlate with segmental duplications in the human genome. *Hum. Mol. Genet.* **11,** 1987–1995 (2002).

72. Li, H., Ruan, J. & Durbin, R. Mapping short DNA sequencing reads and calling variants using mapping quality scores. *Genome Res.* **18,** 1851–1858 (2008).

73. Pavetić, F., Žužić, G. & Šikić, M. *LCSk*++: Practical similarity metric for long strings. Preprint at https://doi.org/10.48550/arXiv.1407.2407 (2014).

74. Stöcker, B. K., Köster, J. & Rahmann, S. SimLoRD: simulation of long read data. *Bioinformatics* **32,** 2704–2706 (2016).

75. Yang, C., Chu, J., Warren, R. L. & Birol, I. NanoSim: nanopore sequence read simulator based on statistical characterization. *GigaScience* **6,** 1–6 (2017).

76. Zook, J. M. *et al.* Extensive sequencing of seven human genomes to characterize benchmark reference materials. *Sci. Data* **3,** 160025 (2016).

77. Zook, J. M. *et al.* An open resource for accurately benchmarking small variant and reference calls. *Nat. Biotechnol.* **37,** 561–566 (2019).

78. Cleary, J. G. *et al.* Joint variant and de novo mutation identification on pedigrees from high-throughput sequencing data. *J. Comput. Biol.* **21,** 405–419 (2014).

79. Cleary, J. G. *et al.* Comparing variant call files for performance benchmarking of Next-Generation Sequencing variant calling pipelines. Preprint at https://doi.org/10.1101/023754 (2015).

80. Chaisson, M. J. & Tesler, G. Mapping single molecule sequencing reads using basic local alignment with successive refinement (BLASR): application and theory. *BMC Bioinf.* **13,** 238 (2012).

81. Bailey, J. A., Church, D. M., Ventura, M., Rocchi, M. & Eichler, E. E. Analysis of segmental duplications and genome assembly in the mouse. *Genome Res.* **14,** 789–801 (2004).

82. Luo, R. *et al.* Clair: Exploring the limit of using a deep neural network on pileup data for germline variant calling. *Nat. Mach. Intell.* **2,** 220–227 (2020).

83. Chailangkarn, T., Noree, C. & Muotri, A. R. The contribution of GTF2I haploinsufficiency to Williams syndrome. *Mol. Cell. Probes* **40,** 45–51 (2018).

84. Sherry, S. T. *et al.* dbSNP: the NCBI database of genetic variation. *Nucleic Acids Res.* **29,** 308–311 (2001).

85. Eberle, M. A. *et al.* A reference data set of 5.4 million phased human variants validated by genetic inheritance from sequencing a three-generation 17-member pedigree. *Genome Res.* **27,** 157–164 (2017).

86. Karczewski, K. J. *et al.* The mutational constraint spectrum quantified from variation in 141,456 humans. *Nature* **581,** 434–443 (2020).

87. Zook, J. M. *et al.* Integrating human sequence data sets provides a resource of benchmark SNP and indel genotype calls. *Nat. Biotechnol.* **32,** 246–251 (2014).

88. Marks, P. *et al.* Resolving the full spectrum of human genome variation using Linked-Reads. *Genome Res.* **29,** 635–645 (2019).

89. Wang, O. *et al.* Efficient and unique cobarcoding of second-generation sequencing reads from long DNA molecules enabling cost-effective and accurate sequencing, haplotyping, and de novo assembly. *Genome Res.* **29,** 798–808 (2019).

90. Chen, Z. *et al.* Ultralow-input single-tube linked-read library method enables short-read second-generation sequencing systems to routinely generate highly accurate and economical long-range sequencing information. *Genome Res.* **30,** 898–909 (2020).

91. Simpson, J. T. *et al.* Detecting DNA cytosine methylation using nanopore sequencing. *Nat. Methods* **14,** 407–410 (2017).

92. Rand, A. C. *et al.* Mapping DNA methylation with high-throughput nanopore sequencing. *Nat. Methods* **14,** 411–413 (2017).

93. Chen, J.-M., Cooper, D. N., Chuzhanova, N., Férec, C. & Patrinos, G. P. Gene conversion: mechanisms, evolution and human disease. *Nat. Rev. Genet.* **8,** 762–775 (2007).

94. Dumont, B. L. Interlocus gene conversion explains at least 2.7% of single nucleotide variants in human segmental duplications. *BMC Genomics* **16,** 456 (2015).

95. Sudmant, P. H. *et al.* An integrated map of structural variation in 2,504 human genomes. *Nature* **526,** 75–81 (2015).

96. Heller, D., Vingron, M., Church, G. & Gaeg, S. SDip: A novel graph-based approach to haplotype-aware assembly based structural variant calling in targeted segmental duplications sequencing. Preprint at https://doi.org/10.1101/2020.02.25.964445 (2020).

97. Lefebvre, S. *et al.* Identification and characterization of a spinal muscular atrophy-determining gene. *Cell* **80,** 155–165 (1995).

98. Gonzalez, E. *et al.* The influence of CCL3L1 gene-containing segmental duplications on HIV-1/AIDS susceptibility. *Science* **307,** 1434–1440 (2005).

99. Shearer, A. E. *et al.* Copy number variants are a common cause of non-syndromic hearing loss. *Genome Med.* **6,** 1–10 (2014).

100. Mueller, M. *et al.* Genomic pathology of SLE-associated copy-number variation at the FCGR2C/FCGR3B/FCGR2B locus. *Am. J. Hum. Genet.* **92,** 28–40 (2013).

101. Carpenter, D. *et al.* Obesity, starch digestion and amylase: association between copy number variant at human salivary (AMY1) and pancreatic (AMY2) amylase genes. *Hum. Mol. Genet.* **24,** 3472–3480 (2015).

102. Armour, J. A. *et al.* Accurate, high-throughput typing of copy number variation using paralogue ratios from dispersed repeats. *Nucleic Acids Res.* **35,** e19 (2007).

103. Ito, T. *et al.* Rapid screening of copy number variations in STRC by droplet digital PCR in patients with mild-to-moderate hearing loss. *Hum. Genome Var.* **6,** 1–6 (2019).

104. Schouten, J. P. *et al.* Relative quantification of 40 nucleic acid sequences by multiplex ligation-dependent probe amplification. *Nucleic Acids Res.* **30,** e57 (2002).

105. Calucho, M. *et al.* Correlation between SMA type and SMN2 copy number revisited: an analysis of 625 unrelated Spanish patients and a compilation of 2834 reported cases. *Neuromuscular Disord.* **28,** 208–215 (2018).

106. Yoon, S., Xuan, Z., Makarov, V., Ye, K. & Sebat, J. Sensitive and accurate detection of copy number variants using read depth of coverage. *Genome Res.* **19,** 1586–1592 (2009).

107. Klambauer, G. *et al.* cn.MOPS: mixture of Poissons for discovering copy number variations in next-generation sequencing data with a low false discovery rate. *Nucleic Acids Res.* **40,** e69 (2012).

108. Layer, R. M., Chiang, C., Quinlan, A. R. & Hall, I. M. LUMPY: a probabilistic framework for structural variant discovery. *Genome Biol.* **15,** 1–19 (2014).

109. Gross, A. M. *et al.* Copy-number variants in clinical genome sequencing: deployment and interpretation for rare and undiagnosed disease. *Genet. Med.* **21,** 1121–1130 (2019).

110. Alkan, C. *et al.* Personalized copy number and segmental duplication maps using next-generation sequencing. *Nat. Genet.* **41,** 1061–1067 (2009).

111. Lopez-Lopez, D. *et al.* SMN1 copy-number and sequence variant analysis from next-generation sequencing data. *Hum. Mutat.* **41,** 2073–2077 (2020).

112. Gould, G. M. *et al.* Detecting clinically actionable variants in the 3' exons of PMS2 via a reflex workflow based on equivalent hybrid capture of the gene and its pseudogene. *BMC Med. Genet.* **19,** 1–13 (2018).

113. Vijzelaar, R. *et al.* The frequency of SMN gene variants lacking exon 7 and 8 is highly population dependent. *PLoS One* **14,** e0220211 (2019).

114. Lan, T. *et al.* Deep whole-genome sequencing of 90 Han Chinese genomes. *GigaScience* **6,** 1–7 (2017).

115. Byrska-Bishop, M. *et al.* High coverage whole genome sequencing of the expanded 1000 Genomes Project cohort including 602 trios. Preprint at https://doi.org/10.1101/2021.02.06.430068 (2021).

116. Verpy, E. *et al.* Mutations in a new gene encoding a protein of the hair bundle cause non-syndromic deafness at the DFNB16 locus. *Nat. Genet.* **29,** 345–349 (2001).

117. Hruska, K. S., LaMarca, M. E., Scott, C. R. & Sidransky, E. Gaucher disease: mutation and polymorphism spectrum in the glucocerebrosidase gene (GBA). *Hum. Mutat.* **29,** 567–583 (2008).

118. Kiiski, K. *et al.* A recurrent copy number variation of the NEB triplicate region: only revealed by the targeted nemaline myopathy CGH array. *Eur. J. Hum. Genet.* **24,** 574–580 (2016).

119. DePristo, M. A. *et al.* A framework for variation discovery and genotyping using next-generation DNA sequencing data. *Nat. Genet.* **43,** 491–498 (2011).

120. Zhu, M. *et al.* Using ERDS to infer copy-number variants in high-coverage genomes. *Am. J. Hum. Genet.* **91,** 408–421 (2012).

121. Prodanov, T. & Bansal, V. Sensitive alignment using paralogous sequence variants improves long-read mapping and variant calling in segmental duplications. *Nucleic Acids Res.* **48,** e114 (2020).

122. Wagner, J. *et al.* Benchmarking challenging small variants with linked and long reads. *Cell Genomics* **2,** 100128 (2022).

123. Casola, C., Zekonyte, U., Phillips, A. D., Cooper, D. N. & Hahn, M. W. Interlocus gene conversion events introduce deleterious mutations into at least 1% of human genes associated with inherited disease. *Genome Res.* **22,** 429–435 (2012).

124. Turro, E. *et al.* Whole-genome sequencing of patients with rare diseases in a national health system. *Nature* **583,** 96–102 (2020).

125. Taliun, D. *et al.* Sequencing of 53,831 diverse genomes from the NHLBI TOPMed Program. *Nature* **590,** 290–299 (2021).

126. Backman, J. D. *et al.* Exome sequencing and analysis of 454,787 UK Biobank participants. *Nature* **599,** 628–634 (2021).

127. Baum, L. E. & Petrie, T. Statistical inference for probabilistic functions of finite state Markov chains. *Ann. Math. Stat.* **37,** 1554–1563 (1966).

128. Stratonovich, R. L. Conditional Markov processes. in *Non-linear Transformations of Stochastic Processes*, 427–453 (Elsevier, 1965).

129. Viterbi, A. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory* **13,** 260–269 (1967).

130. Frankish, A. *et al.* GENCODE reference annotation for the human and mouse genomes. *Nucleic Acids Res.* **47,** D766–D773 (2019).

131. Stephens, Z. D. *et al.* Simulating next-generation sequencing datasets from empirical mutation and sequencing models. *PloS One* **11,** e0167047 (2016).

132. Horvath, J. E., Schwartz, S. & Eichler, E. E. The mosaic structure of human pericentromeric DNA: a strategy for characterizing complex regions of the human genome. *Genome Res.* **10,** 839–852 (2000).

133. European Polycystic Kidney Disease Consortium *et al.* The polycystic kidney disease 1 gene encodes a 14 kb transcript and lies within a duplicated region on chromosome 16. *Cell* **77,** 881–894 (1994).

134. Willey, C. J. *et al.* Prevalence of autosomal dominant polycystic kidney disease in the European Union. *Nephrol. Dial. Transpl.* **32,** 1356–1363 (2017).

135. Rehm, H. L. *et al.* ACMG clinical laboratory standards for next-generation sequencing. *Genet. Med.* **15,** 733–747 (2013).

136. Thongnoppakhun, W., Wilairat, P., Vareesangthip, K. & Yenchitsomanus, P.-t. Long RT-PCR amplification of the entire coding sequence of the polycystic kidney disease 1 (PKD1) gene. *Biotechniques* **26,** 126–132 (1999).

137. Clendenning, M. *et al.* Long-range PCR facilitates the identification of PMS2-specific mutations. *Hum. Mutat.* **27,** 490–495 (2006).

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

139. Kerzendorfer, C., Konopka, T. & Nijman, S. M. A thesaurus of genetic variation for interrogation of repetitive genomic regions. *Nucleic Acids Res.* **43,** e68 (2015).

140. Prodanov, T. & Bansal, V. Robust and accurate estimation of paralog-specific copy number for duplicated genes using whole-genome sequencing. *Nat. Commun.* **13,** 3221 (2022).

141. Guo, Y. *et al.* The effect of strand bias in Illumina short-read sequencing data. *BMC Genomics* **13,** 1–11 (2012).

142. Fisher, R. A. On the interpretation of χ^2 from contingency tables, and the calculation of P. *J. R. Stat. Soc.* **85,** 87–94 (1922).

143. Karp, R. M. Reducibility among combinatorial problems. in *Complexity of computer computations*, 85–103 (Springer, 1972).

144. Huang, W., Li, L., Myers, J. R. & Marth, G. T. ART: a next-generation sequencing read simulator. *Bioinformatics* **28,** 593–594 (2012).

145. Danecek, P. *et al.* Twelve years of SAMtools and BCFtools. *GigaScience* **10.** 10.1093/gigascience/giab008. https://doi.org/10.1093/gigascience/giab008 (2021).

146. Robinson, J. T. *et al.* Integrative genomics viewer. *Nat. Biotechnol.* **29,** 24–26 (2011).

147. Tarjan, R. Depth-first search and linear graph algorithms. *SIAM J. Comput.* **1,** 146–160 (1972).

148. Shebanits, K. *et al.* Copy number determination of the gene for the human pancreatic polypeptide receptor NPY4R using read depth analysis and droplet digital PCR. *BMC Biotechnol.* **19,** 31 (2019).

149. Cleveland, W. S. Robust locally weighted regression and smoothing scatterplots. *J. Am. Stat. Assoc.* **74,** 829–836 (1979).

150. Cleveland, W. S. LOWESS: A program for smoothing scatterplots by robust locally weighted regression. *Am. Stat.* **35,** 54 (1981).

151. Prjibelski, A., Antipov, D., Meleshko, D., Lapidus, A. & Korobeynikov, A. Using SPAdes de novo assembler. *Curr. Protoc. Bioinform.* **70,** e102 (2020).

152. Miller, C. A., Hampton, O., Coarfa, C. & Milosavljevic, A. ReadDepth: a parallel R package for detecting copy number alterations from short sequencing reads. *PloS One* **6,** e16327 (2011).

153. Sampson, J., Jacobs, K., Yeager, M., Chanock, S. & Chatterjee, N. Efficient study design for next generation sequencing. *Genet. Epidemiol.* **35,** 269–277 (2011).

154. Savani, V. & Zhigljavsky, A. A. Efficient estimation of parameters of the negative binomial distribution. *Commun. Stat. — Theory Methods* **35,** 767–783 (2006).

155. Altman, N. S. An introduction to kernel and nearest-neighbor nonparametric regression. *Am. Stat.* **46,** 175–185 (1992).

156. Ewing, B. & Green, P. Base-calling of automated sequencer traces using phred. II. Error probabilities. *Genome Res.* **8,** 186–194 (1998).

157. Goya, R. *et al.* SNVMix: predicting single nucleotide variants from next-generation sequencing of tumors. *Bioinformatics* **26,** 730–736 (2010).

158. Hohenlohe, P. A. *et al.* Population genomics of parallel adaptation in threespine stickleback using sequenced RAD tags. *PLoS Genet.* **6,** e1000862 (2010).

159. Dempster, A. P., Laird, N. M. & Rubin, D. B. Maximum likelihood from incomplete data via the EM algorithm. *J. R. Stat. Soc. Ser. B Methodol.* **39,** 1–22 (1977).

160. Müllner, D. Fastcluster: Fast hierarchical, agglomerative clustering routines for R and Python. *J. Stat. Softw.* **53,** 1–18 (2013).

161. Benjamini, Y. & Hochberg, Y. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *J. R. Stat. Soc. Ser. B Methodol.* **57,** 289–300 (1995).