STRUCTURAL VARIANT CALLING USING THIRD-GENERATION SEQUENCING DATA

Detektion von genomischen Strukturvarianten mit Sequenziertechnologien der dritten Generation

DAVID HELLER

Dissertation zur Erlangung des Grades eines Doktors der Naturwissenschaften (Dr. rer. nat.) am Fachbereich Mathematik und Informatik der Freien Universität Berlin

Berlin, 2020

Erstgutachter: Prof. Dr. Martin Vingron
Zweitgutachter: Prof. Dr. Tobias Marschall
Datum der Disputation: 12. Januar 2021

CONTENTS

Appendix

LIST OF FIGURES

LIST OF TABLES

PUBLICATIONS

Some ideas, passages and figures have been published in:

[1] David Heller and Martin Vingron. "SVIM: structural variant identification using mapped long reads." In: *Bioinformatics* 35.17 (2019), pp. 2907–2915.

The following publications cover work performed as part of this PhD project on topics that are related to this thesis:

[1] Shilpa Garg, Arkarachai Arkarachai Fungtammasan, Andrew Carroll, Mike Chou, Anthony Schmitt, Xiang Zhou, Stephen Mac, Paul Peluso, Emily Hatas, Jay Ghurye, et al. "Efficient chromosome-scale haplotype-resolved assembly of human genomes." In: *Nature Biotechnology* (in press).

[2] Glenn Hickey, David Heller, Jean Monlong, Jonas A Sibbesen, Jouni Siren, Jordan Eizenga, Eric T Dawson, Erik Garrison, Adam M Novak, and Benedict Paten. "Genotyping structural variants in pangenome graphs using the vg toolkit." In: *Genome Biology* 21.1 (2020), pp. 1–17.

[3] Francisca M Real, Stefan A Haas, Paolo Franchini, Peiwen Xiong, Oleg Simakov, Heiner Kuhl, Robert Schöpflin, David Heller, M-Hossein Moeinzadeh, Verena Heinrich, et al. "The mole genome reveals regulatory rearrangements associated with adaptive intersexuality." In: *Science* (in press).

In normal life we hardly realize how much more we receive than we give and life cannot be rich without such gratitude. It is so easy to overestimate the importance of our own achievements compared with what we owe to the help of others.

— Dietrich Bonhoeffer

ACKNOWLEDGMENTS

Throughout my doctoral studies I have received a great deal of support and assistance. First and foremost, I would like to express my sincere gratitude to my supervisor Prof. Martin Vingron for his invaluable support, advice and encouragement. I also wish to thank the other members of my thesis advisory committee, Prof. Knut Reinert und Prof. Rosario Piro, for their guidance and helpful ideas which were of great benefit to this project. My work was made possible by the generous funding that I received through the doctoral program of the International Max Planck Research School for Computational Biology and Scientific Computing.

I would also like to acknowledge Prof. Stefan Mundlos, Shilpa Garg, René Rahn and all the other colleagues in Germany and abroad that I had the privilege to collaborate with over the last years. I wish to thank Prof. Benedict Paten for the wonderful opportunity of joining his group at the UC Santa Cruz for a few months and diving into the world of variant graphs.

I thank all my fellow group members for stimulating discussions, their valuable feedback and the great time I had in the group. Additionally, I would like to thank Kirsten Kelleher for proofreading this thesis and for patiently answering all my questions.

I wish to thank my wife for her immense support, love and sympathetic ear. I would also like to thank my parents for their steady encouragement and guidance through the years. Finally, I thank my friends, not only for their friendship but also for the opportunity to rest my mind outside of research from time to time.

1

INTRODUCTION

Life on earth comes in many different shapes, sizes and colors. According to recent studies, it is estimated that our planet is inhabited by several million or even more than a trillion species [64, 70]. Despite this great diversity, all living organisms have one thing in common. In each of their cells they possess the long molecule *deoxyribonucleic acid* (DNA). This molecule acts as a blueprint for the organism and comprises most of the information that is required to guide its development and define its unique morphology and behavior.

A DNA molecule is composed of two complementary strands of nucleotides (denoted forward and reverse strand) that are bound together into a double helix. Because there are four different nucleotides (adenine, cytosine, guanine and thymine), the DNA can be regarded as a string, i.e. a sequence of characters, drawn from the four-letter alphabet $\sum_{DNA} = \{A, C, G, T\}$. This string is structured into different segments of which the most prominent, known as *genes*, encode the structure of proteins and other important building blocks of the organism. With the exception of most microbial species, many organisms possess not only one but multiple DNA molecules in their cells which are referred to as *chromosomes*. The complete set of all chromosomes comprising an organism's entire genetic information is called its *genome*.

Our human genome with its approximately 3.1 billion nucleotides (or base pairs, bp) is distributed over 22 autosomes (chromosomes 1 to 22) and 2 sex chromosomes (chromosomes X and Y). Because humans belong to the *diploid* organisms, however, our cells possess two sets of chromosomes (as opposed to haploid organisms with only one chromosome set). Each set is inherited either from our mother or father and consists of the 22 autosomes as well as one of the two sex chromosomes. Therefore, the full diploid human genome comprises 44 autosomes (two of each kind) and 2 sex chromosomes (either XX or XY).

Although human genomes are highly similar, studies estimate that there are on average between 4 and 5 million differences, termed *variants*, between different individuals [1]. The different alternative forms of a DNA sequence that originate from a variant are termed *alleles*. A set of closely linked alleles that are inherited together from a single parent is known as a *haplotype*. While the majority of variants affect only a few base pairs, larger and more complex rearrangements referred to as *structural variants* do exist. In total, the sum of all variants in an average individual is estimated to affect more than 25 million nucleotides.

The entirety of genomic variants in a given individual (their *genotype*) influences to a large extent their unique observable properties and traits including physical appearance, character traits and susceptibility to disease (their *phenotype*). Therefore, the detection of the unique set of variants for a given individual (*variant calling*) and the reconstruction of its genome sequence (*genome assembly*) are highly important tasks. They provide the basis for deepening our understanding of the complex network of genes, proteins and regulatory elements that is defined by our genome.

Over the last few decades, new molecular and computational methods have been developed that enabled the analysis of genomes at unprecedented resolution and scale. In the mid-2000s, next-generation sequencing considerably reduced the cost of genome sequencing by producing large numbers of short sequence fragments known as *reads*. These reads can be localized and aligned on a reference genome facilitating a comparison of the sequenced genome with the reference. More recently, third-generation sequencing technologies have enabled the generation of substantially longer reads driving advances in variant calling and genome assembly.

Due to their large size and great impact, the characterization of structural variants is of particular importance in fields like genetics, medicine and genomics. Therefore, structural variant detection has been the target of extensive research efforts over the last decades. Due to biological, technical and algorithmic challenges, however, neither the comprehensive detection of structural variants nor the complete reconstruction of personal genome sequences could be fully achieved yet.

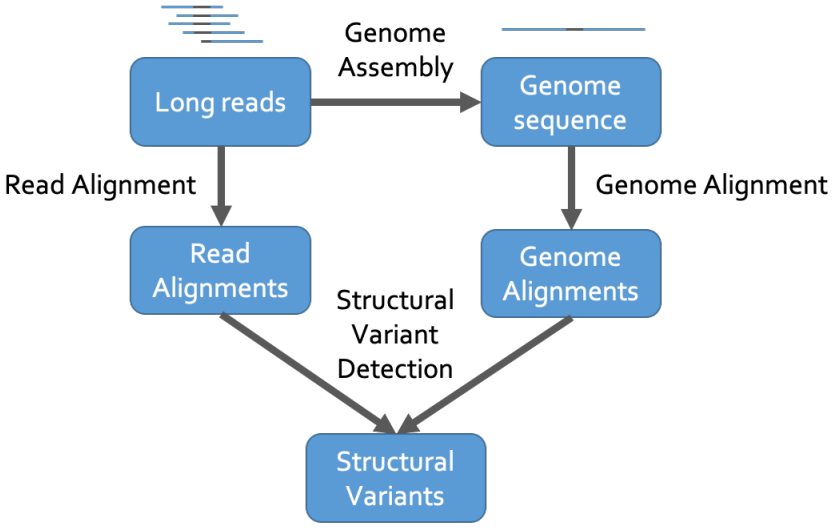


Figure 1.1: **Schematic overview over the topics discussed in this doctoral thesis.** Long sequencing reads from third-generation sequencing technologies can be aligned and compared to an existing reference genome (read alignment). Alternatively, the reads can be assembled into a complete genome sequence (genome assembly) followed by its comparison with the reference genome (genome alignment). The alignments of reads or genomes can be analyzed to detect structural variants.

1.1 RESEARCH OBJECTIVE

This thesis presents a new computational method for the detection of structural variants using third-generation sequencing data. We apply this new method on simulated and real sequencing datasets and compare it to existing approaches. Furthermore, we use our method to detect structural variants from genome assemblies. This work covers topics at the interface between genome assembly and structural variant calling as illustrated in Figure 1.1. Our novel contributions are:

- an accurate and user-friendly software tool for the **detection and genotype estimation of structural variants** from long reads or genome assemblies
- **classification of six classes of structural variation** including similar classes, such as insertions and two types of duplications
- **genotyping of structural variants** from long reads and genome assemblies with higher accuracy than existing methods

- **detection, filtering and validation of novel adjacencies** in a cohort of highly rearranged genomes

1.2 THESIS OVERVIEW

This chapter shortly introduced the scope of this thesis. The following Chapter 2 will give a more detailed introduction into the landscape of human genomic variation and the molecular and computational techniques used for its characterization. Chapter 3 presents the software tool *SVIM* for the detection of structural variants from third-generation sequencing data. The following Chapter 4 explains how the same concepts and algorithms can be adapted for the detection of structural variants from genome assemblies. Chapter 5 presents the evaluation of *SVIM* on various sequencing datasets and its comparison to existing methods. In Chapter 6, *SVIM* is applied on a set of highly rearranged patient genomes and used to investigate the structural variants and novel adjacencies in these genomes. Finally, the thesis will be concluded in Chapter 7 with a discussion of the findings and avenues for future research.

2

GENOMIC VARIATION AND GENOME SEQUENCING

The variation landscape of the human genome is complex and diverse. This is a consequence of the multitude of biological mechanisms that can cause changes in the genetic sequence and structure of the genome. Over the years, the scientific community has gained an increasingly deeper understanding of these mechanisms and the different classes of human genomic variation. The most comprehensive class of variation is termed *structural variation* and it ranges from simple alterations, such as deletions, to large and complex rearrangements that modify the greater structure of the genome. In this chapter, we characterize the landscape of human genomic variation with a particular emphasis on structural variation. We describe how structural variants form and what is known about their functional consequences. Finally, we introduce the molecular and computational approaches that have been applied now and in the past to detect and characterize structural and other types of variation.

2.1 HUMAN GENOMIC VARIATION

When comparing any two human genomes, a multitude of differences can be observed. Current studies estimate the average number of genetic variants in any given genome to be between 4.1 and 5.0 million but the number varies between populations [1]. For instance, it has been found that genomes from African populations harbor substantially more variants than genomes from other populations. All variants combined amount to a total sequence of approximately 25 Mbp in an average genome [1]. Despite these high numbers it is worth keeping in mind, however, that the vast majority of nucleotides (more than 99%) in any two human genomes are identical.

2.1.1 *Classes of genomic variation*

In most higher organisms, all the cells in the body are descendants of a single fertilized egg cell. Therefore, they inherit in principle the same genome which is why we speak of one genome for each organism. Nevertheless, each cell within an organism can gain private mutations making its genome different from that of the other cells. This variation between individual cells is termed *somatic variation*. It plays a particularly important role in cancer where tumor cells acquire multiple important somatic mutations that enable them to grow faster than non-tumor cells. Although somatic variants can be detected with approaches similar to those discussed in the following chapters, they are not the focus of this thesis. Instead, we describe approaches for the detection of *germline variants*, i.e. differences between the genomes of different individuals. These variants can be categorized into different classes according to their properties.

BY SIZE The first and most intuitive way of categorizing the variants in a genome is by their size. The smallest and most frequent class of variants in the human genome are the so-called *Single Nucleotide Variants* (SNVs). As the name indicates, SNVs denote changes of only a single base pair. Recent studies estimate that the average number of SNVs in a genome lies between 3.5 and 4.3 million [1]. The second most frequent class of variants in the human genome are short *insertions* and *deletions* (indels) with on average more than 500,000 variants per genome [1]. While insertions denote additional bases present in one genome but absent in the other, deletions represent the inverse case of bases missing from one of the genomes. A third class of variants in the human genome is known as *Structural Variants* (SVs). They are larger in size than SNVs and indels and can represent major rearrangements in the structure of the genome. SVs are the most diverse class of variants and encompass all conceivable genomic rearrangements of large size.

BY FREQUENCY Another way of categorizing variants in the human genome is by their frequency. Large-scale studies like the 1000 Genomes Project have analyzed variation in thousands of human genomes from different populations around the globe. They found

that some variants are common (i.e. they are present in more than 1% of the population) while others are rare [29]. The majority of variants in a given genome are common and therefore referred to as *polymorphisms* [1]. Nevertheless, the majority of variants found in the entire population are rare. Approximately 76 million variants detected in the 1000 Genomes Project had a frequency below 5% compared to only 8 million with a frequency above 5% [1].

BY COPY NUMBER Thirdly, the variants in a genome can be categorized into *copy-number variants* (CNVs) and *balanced variants*. CNVs, such as insertions, duplications and deletions, are adding or removing genetic material and can therefore have a direct effect on the dosage of a gene [77]. Balanced variants, such as inversions and translocations, on the other hand, merely change the orientation, order or location of genomic segments. Although they do not add or remove any genetic material they have the potential to disturb the gene structure and regulatory landscape of the genome causing phenotypic changes [76].

BY ZYGOSITY A fourth way of categorizing variants in a diploid genome, such as the human genome, is by their *zygosity*. This term refers to the presence of a variant in the two chromosome sets. Because a human cell possesses a pair of each autosome (and of chromosome X in female individuals), a given variant can be present in one or both of these *homologous*, i.e. related, chromosomes. Variants that are present in only one of the two homologous chromosomes are called *heterozygous* while variants present in both are called *homozygous*. The zygosity of a variant is often also referred to as the *genotype* of the individual at this position. But whereas the genotype denotes the complete genetic make-up or nucleotide sequence of an individual on both homologous chromosomes, the zygosity only refers to the presence or absence of a variant.

2.1.2 *Structural variation*

Structural variants are now widely defined as rearrangements larger than 50 bp [2, 19]. The term encompasses a broad spectrum of

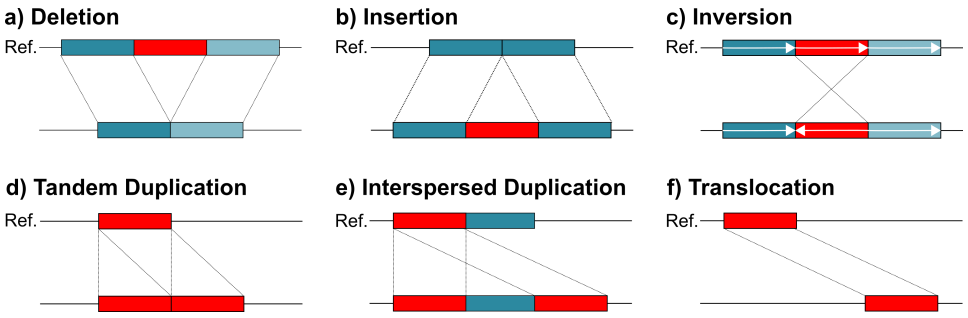


Figure 2.1: **Six classes of structural variation.** Structural variants can be categorized into deletions, insertions, inversions, duplications (in tandem or interspersed) and translocations. Each SV class is depicted in an individual genome (lower line) when compared to the reference genome (upper line). The regions being rearranged are marked in red.

genomic alterations that can affect from a few dozen to millions of base pairs. Typically, the scientific community distinguishes the following common SV classes (depicted in Figure 2.1) [2]:

- *Deletions*: A deletion denotes the removal of a genomic segment that is present in the reference genome but missing in the individual genome (panel a). Deletions are a type of CNV.
- *Insertions*: An insertion denotes that a genomic segment which is not present in the reference genome has been added at a certain location in the individual genome (panel b). Insertions are a type of CNV.
- *Inversions*: An inversion denotes the reversal of a genomic segment in the individual genome such that the sequence in the segment is replaced by its reverse-complement (panel c). Inversions are a type of balanced rearrangement.
- *Tandem duplications*: A tandem duplication denotes the duplication of a genomic segment and the adjacent insertion of the copy in the individual genome (panel d). Tandem duplications are a type of CNV and can also be regarded as a special type of insertion.
- *Interspersed duplications*: An interspersed duplication denotes the duplication of a genomic segment and the insertion of the copy at a distant location in the individual genome (panel e).

The copy can be inserted either somewhere else in the source chromosome (intra-chromosomal) or in a different chromosome (inter-chromosomal). Interspersed duplications are a type of CNV and can also be regarded as a special type of insertion.

- *Translocations*: A translocation denotes the change in position of a genomic segment (panel f). The genomic segment can be moved to another position in the same chromosome (intra-chromosomal) or to another chromosome (inter-chromosomal). The most common type of translocation, known as *reciprocal translocation*, denotes the exchange of genomic segments between two chromosomes. Translocations are often expressed in terms of *novel adjacencies*, i.e. connections between two genomic loci in the individual genome that are distant in the reference genome.

These six canonical classes are by no means complete and several other classes and subclasses of SVs have been defined. Furthermore, combinations of multiple SVs at the same locus can lead to complex or nested rearrangements. Numerous classes of complex SVs have been described, such as inversions flanked by deletions or duplications [16].

The boundaries between the six canonical SV classes are not always clear and some classes overlap. Duplications for instance can be regarded as a special type of insertion where the inserted sequence is identical to an existing portion of the genome. When the inserted copy misses a few bases, however, the rearrangement can be represented either as a duplication with a deletion or as a simple insertion. Ambiguities like these are common because many differences between two genomes have multiple possible interpretations.

Another aspect to keep in mind is that the variant class determined for a given genomic difference does not necessarily indicate how the difference developed during evolutionary history. To illustrate this, consider a deletion that is observed in an individual genome when compared to the reference genome. The most intuitive explanation for the deletion is that the genomic segment was removed in the evolutionary past of the individual genome while it was retained in that of the reference genome. However, another explanation is that the segment was added in a recent insertion event to the reference

genome and never existed in the lineage of the individual genome. Furthermore, the variant class that is identified for a given genomic difference is closely linked to the choice of one of the genomes as reference. All deletions detected in a given individual genome will be categorized as insertions and vice versa when the individual genome is chosen as reference. These two examples illustrate that differences between two genomes do not reveal much about their origin and are dependent on the choice of one genome as the reference.

2.1.2.1 *Mechanisms of SV formation*

SVs are known to form during DNA recombination, DNA replication and DNA repair [10]. Two general mechanisms can be distinguished that give rise to structural rearrangements in the genome: *homologous recombination* (HR) and *non-homologous recombination* [40]. While HR requires long stretches of DNA with high sequence similarity (*homology*), non-homologous recombination mechanisms require no or only short stretches of similar sequence (*microhomologies*).

HR forms the basis for several accurate DNA repair mechanisms that repair a damaged sequence using a similar sequence template [40]. Most often, the homologous chromosome of a diploid set is used for this purpose. However, the presence of two homologous regions in the genome can cause misalignment through a process called *Non-allelic homologous recombination* (NAHR, Figure 2.2a). Subsequent crossover between the two homologous chromosomes can result in deletions, duplications, inversions and translocations.

Another repair mechanism known as *break-induced replication* (BIR) can be triggered by nicks in one DNA strand causing the DNA replication process to fail (Figure 2.2b) [50]. To repair the strand breakage, the DNA strand aligns to an homologous region on another DNA molecule and uses it as a template for replication. The misalignment between different homologous regions can result in deletions, duplications and inversions constituting an alternative mechanism for NAHR [40].

SVs can also form as a result of repair processes that do not require extensive homology, such as non-homologous end joining (NHEJ) [10]. NHEJ is a repair mechanism that joins broken DNA ends originating from double-strand breaks (Figure 2.2c). Instead of

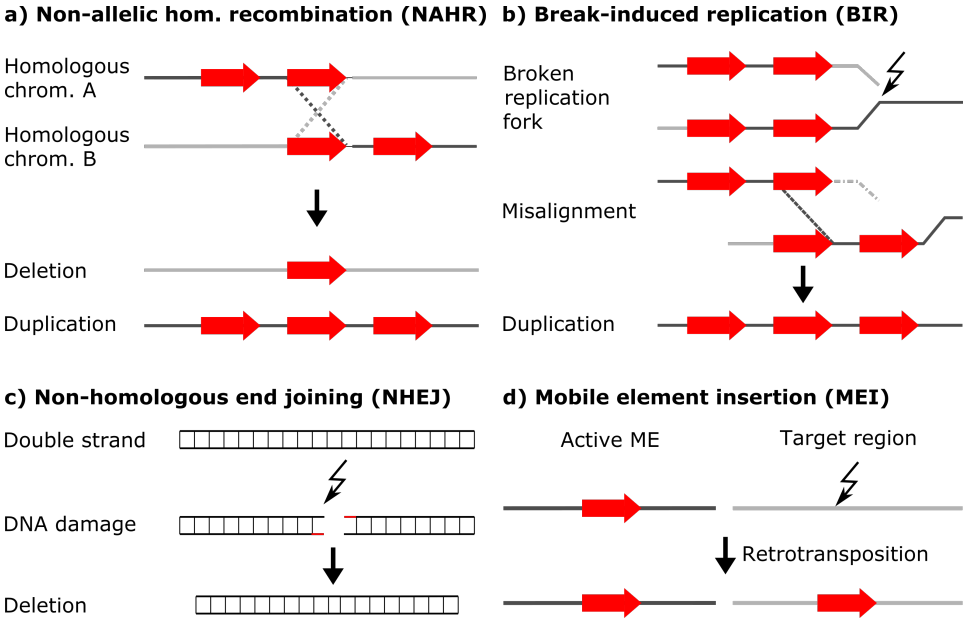


Figure 2.2: **Molecular mechanisms causing the formation of SVs.**
a | Non-allelic homologous recombination denotes the recombination between long homologous stretches of DNA (red arrows) from different genomic regions. In this example, the second region from chromosome A misaligns to the first instead of second region from chromosome B. Subsequent cross-over between the chromosomes leads to a deletion (light grey path) or a duplication (dark grey path).
b | Break-induced replication (BIR) is a mechanism to repair one-ended breaks in the DNA, e.g. caused by a broken replication fork. The broken DNA (upper line) is repaired by using an homologous region in another DNA molecule (lower line) as a template. Similarly to NAHR, a misalignment between different homologous regions can result in variations, such as duplications.
c | Non-homologous end joining (NHEJ) is a repair mechanism for double-strand breaks that rejoins the two ends without requiring extensive homology. As a consequence of this inaccurate repair mechanism, small deletions (in red), insertions or translocations can occur at the repair site.
d | Active mobile elements in the genome, such as LINE-1 elements, are able to duplicate their own sequence and to insert it at another location (target region) in the genome. These *mobile element insertions* (MEI) can in turn also promote the formation of new variants through homologous recombination.

long sequence homologies, NHEJ either uses microhomologies to guide repair or joins ends without any homology. This can lead to small insertions or deletions or even the fusion of unrelated DNA strands [40].

Another source of SVs in human genomes are duplications of mobile elements, such as Alu, LINE-1 and SVA elements [90]. Mobile elements are segments in the genome that are able to move around with the help of proteins. The mobile elements in the human genome duplicate using a process called *retrotransposition* and insert their duplicated sequence into other locations across the genome (Figure 2.2d). In total, they are responsible for a considerable fraction of the structural variation found in human genomes. A large-scale study that was part of the 1000 Genomes Project estimated that approximately 25% of detected SVs were mobile element insertions [92]. Most of these insertions formed during ancient retrotransposition events. However, a minority of elements, particularly LINE-1 elements, remains active and still contributes to human genomic variation. *DNA transposons*, another class of mobile elements, use a cut&paste mechanism to move around the genome. They are now inactive in most mammals except bats but remain active in plants and lower-order animals [41].

As we have seen, many SVs are caused either by the misalignment of homologous regions in the genome or the duplication of sequence by retrotransposition. Therefore, it is not surprising that SVs are commonly found in repetitive regions of the genome [3, 77, 85].

2.1.2.2 *Functional consequences*

Although SNVs and indels make up more than 99% of all variants in an average human genome, studies have shown that SVs affect more base pairs [1]. Consequently, SVs are responsible for a substantial fraction of human genetic diversity and have a major influence on both normal phenotype and disease [96]. When SVs alter the genotype of an organism, they can influence its phenotype through a wide range of mechanisms. Duplications or deletions, for instance, can change the copy number of a gene or regulatory element and thus have a direct impact on the gene's expression [68]. Alternatively, SVs in coding regions can alter the inner structure of a gene or

fuse different genes together. Furthermore, they can influence the regulation of genes by rearranging their regulatory elements [96]. SVs are also able to alter the 3-D architecture of the genome and thus affect the expression of distant genes [87].

The functional impact of SVs has been most thoroughly studied in the context of human disease. A growing number of syndromes and genomic disorders, such as Smith–Magenis syndrome, Williams–Beuren syndrome and DiGeorge syndrome, are associated with SVs [96]. SVs have also been linked to complex diseases, such as autism, schizophrenia, attention deficit hyperactivity disorder, Crohn's disease, rheumatoid arthritis and diabetes [81, 96]. Moreover, SVs are known to be a major driving force in many cancers [59]. Consequently, the characterization of SVs is of major importance to human medicine and genetics alike. It can contribute to the early detection of disorders and can help to elucidate the underlying genetic and molecular processes [35].

2.2 VARIANT DETECTION THROUGH THE YEARS

The technologies used for the analysis of genomic sequence and variation have changed rapidly over the last decades. From the first observations of chromosomes under a microscope to the high-throughput technologies that are in use today, the ability to investigate more and more genomes with increasing resolution has steadily improved. Nevertheless, each new technology has brought along unique challenges and weaknesses. For many applications, it is therefore necessary to combine multiple complementary technologies to obtain meaningful results.

2.2.1 *Cytogenetic and hybridization-based technologies*

Initial observations of genomic differences were made under a microscope using cytogenetic approaches [27]. At first, condensed and unstained chromosomes were viewed under the microscope in a process termed *karyotyping*, revealing only large rearrangements and changes in chromosome number. Later, elongated prometaphase chromosomes were stained with specialized dyes enabling the iden-

tification of individual chromosomes by characteristic banding patterns. Thus, more subtle variations down to a size of 3 Mb including translocations, deletions, duplications, insertions and inversions could be detected. Using *fluorescence in situ hybridization* (FISH), specific target sequences could be fluorescently labelled, revealing their presence and relative location under the microscope. This enabled the detection of variants down to a size of a few kbp.

Later, hybridization-based methods such as *array-based comparative genomic hybridization* (array-CGH) enabled the detection of CNVs at a higher resolution [27]. When analyzing a genome of interest in comparison to a reference sample with array-CGH, both samples are fragmented and fluorescently labelled with different tags. Then, the labelled fragments are applied to an array with a collection of DNA probes where the sample fragments only bind to matching DNA probes. Finally, the fluorescence from both samples and for every probe can be measured and compared to reveal CNVs between the two samples. Similar to array-CGH, SNP arrays use allele-specific DNA probes to detect not only CNVs but also SNVs.

2.2.2 *Next-generation sequencing*

With the advent of *next-generation sequencing* (NGS) in the mid-2000s, the older cytogenetic and hybridization-based technologies have been gradually replaced by sequencing-based approaches [36]. NGS experiments produce high volumes of sequence data for a fraction of the cost of previous sequencing technologies like Sanger sequencing. This high throughput is achieved through the fragmentation of the DNA into smaller pieces and subsequent amplification to produce many copies of each DNA fragment (see Figure 2.3a). Then, NGS is carried out in a massively parallel fashion with millions of DNA molecules being sequenced at the same time in different reaction chambers of the same sequencing machine.

Two broad categories for NGS exist that either use the ligation of labelled probes (*sequencing by ligation*, SBL) or the incorporation of labelled nucleotides using a DNA polymerase (*sequencing by synthesis*, SBS) to identify the sequence of a DNA fragment. Today, SBS accounts for the largest market share of NGS instruments [36]. Illu-

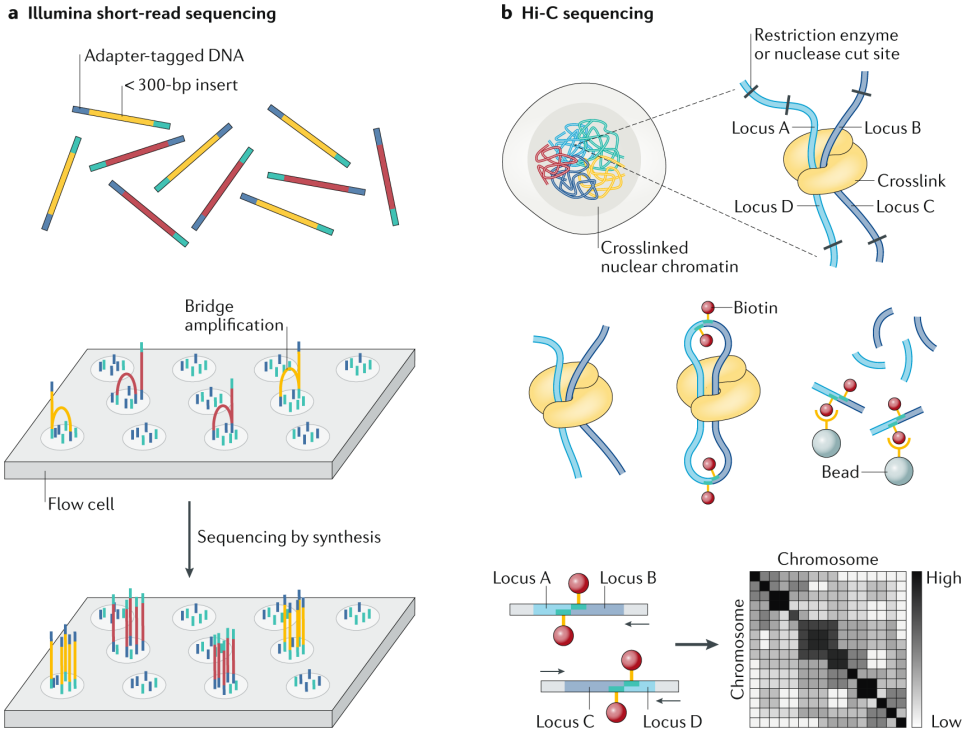


Figure 2.3: **Overview of short-read sequencing technologies.**
a | In Illumina sequencing, DNA fragments (yellow and red) are ligated to adapters (blue and aqua) that attach to oligonucleotides on the surface of the flow cell. Through a process called *bridge amplification*, numerous copies of each fragment are generated. Then, millions of fragments are sequenced at the same time by incorporating fluorescently labelled nucleotides on the fragment. Characteristic light flashes corresponding to each newly incorporated base are recorded using microscopy and enable the detection of the fragment's DNA sequence.
b | In Hi-C, nuclear chromatin is fixed with formaldehyde so that genomic loci in close spatial proximity are cross-linked. After cutting the fixed chromatin with a restriction enzyme, the resulting ends of the DNA fragments are extended with biotin-linked nucleotides and ligated to connect the two cross-linked fragments. Then, the crosslinks are removed and the fragments are split into smaller pieces. Pieces with attached biotin are pulled down with specialized beads and sequenced with Illumina paired-end sequencing. This produces read pairs connecting two genomic loci (light and dark blue). The contact frequency between pairs of loci strongly correlates with their distance in 3-D space and can be visualized in a contact map.

mina, the most prominent manufacturer of NGS machines, provides a range of different sequencing machines that all use an SBS-based technology called *cyclic reversible termination* (CRT). CRT uses labelled nucleotides with an attached blocking group that prevents further elongation after the incorporation of the nucleotide. After the incorporated nucleotide is imaged to identify its label, the blocking group and label are removed and the next incorporation cycle can begin.

NGS experiments produce millions of short DNA sequences known as *reads* that, depending on the instrument, have a length between 70 and 700 bp. Usually, every position of the sequenced genome is contained in several reads where the average number of reads covering any position is termed *sequencing coverage* or *sequencing depth*. In comparison to the earlier Sanger sequencing technology, NGS data has an elevated error rate with current Illumina instruments reaching an accuracy of approximately 99.9% [33]. Like most sequencing platforms, Illumina machines exhibit some systematic bias, such as a propensity towards substitution errors [69] and an under-representation of regions with extreme GC content [39].

Despite these errors and biases, NGS enabled, for the first time, the near-complete resolution of an individual's genome sequence and the comprehensive detection of all classes of variation. Although array-based technologies are still in use due to their high throughput and low cost, NGS possesses two major advantages: its ability to detect balanced variants as well as insertions of sequence not present in the reference genome and its considerably higher resolution down to the base-pair level.

2.2.3 *Long-read sequencing*

Recently, a number of new technologies are promising to revolutionize variant detection. Compared to traditional NGS approaches using short sequencing reads they offer both improved sensitivity and lower false discovery rates. The most versatile of these novel technologies are *long-read sequencing* technologies (also referred to as *third-generation sequencing*, TGS). As the name indicates, they produce longer reads than traditional NGS technologies which enables

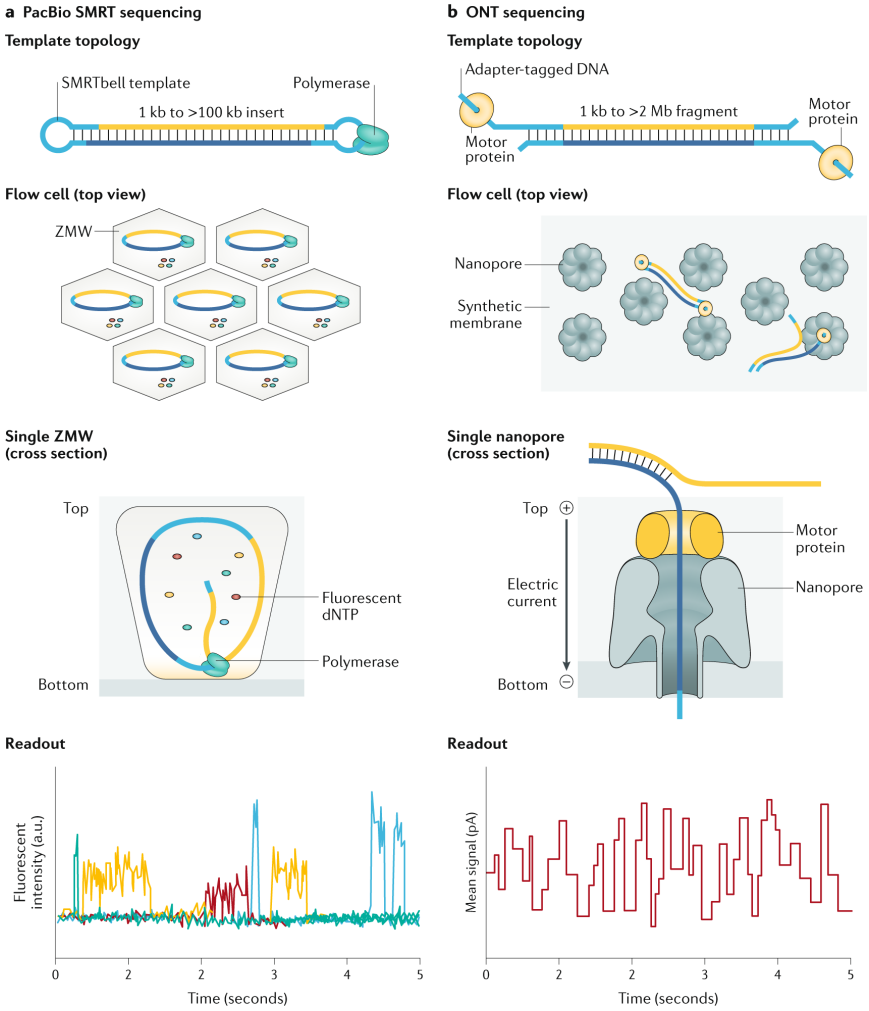


Figure 2.4: **Overview of long-read sequencing technologies.**
a | In Pacific Biosciences (PacBio) sequencing, DNA fragments (yellow for forward strand, dark blue for reverse strand) are ligated to so-called hairpin adapters (light blue) to form circularized DNA templates (SMRTbell). The actual sequencing reaction is performed in small chambers known as Zero-mode waveguides (ZMWs). A DNA polymerase molecule attached to the bottom of each ZMW incorporates fluorescently labelled nucleotides (dNTP) on the DNA template. During incorporation, fluorescent light flashes are emitted and recorded by a camera. Different colors corresponding to the different DNA bases enable the identification of each incorporated nucleotide.
b | In Oxford Nanopore Technologies (ONT) sequencing, DNA fragments are ligated to adapters (light blue) with attached motor proteins. The fragments are passed through protein nanopores embedded in a synthetic membrane. While the motor protein drives the fragment through the pore at a constant speed, an electric field is applied to the pore. The DNA fragment causes characteristic changes in the field's current that are used to identify its DNA sequence.
Reprinted by permission from Springer Nature © 2020 [65]

them to resolve long repeats, structural variants or other complex regions of the genome [36]. When applied in transcriptome studies, long reads are able to cover entire mRNA transcripts. This facilitates the discovery of alternative splicing events and the reconstruction of complete gene isoforms. Long-read sequencing approaches can be broadly categorized into two groups: synthetic long-read sequencing and single-molecule real-time (SMRT) approaches.

As the name already indicates, synthetic long-read sequencing does not produce actual long reads. Instead, it produces short reads using existing sequencing technologies that are later computationally assembled into longer fragments [94]. For this purpose, special library preparation steps have been developed. First, long sequence fragments are distributed across a large number of wells in a plate. Subsequently, the few fragments in each well are amplified, split into smaller fragments and tagged with short characteristic sequence barcodes unique to each well. Then, all fragments from all wells can be sequenced together on a traditional NGS machine. As fragments sharing the same barcode originated from the same well, the barcodes can be used to split the data again and computationally reassemble the original long sequence fragments.

In contrast to synthetic long-read sequencing, SMRT sequencing does not rely on traditional NGS technology but follows a completely different approach. SMRT sequencing instruments analyze a single DNA molecule without prior amplification in real-time, i.e. without the cyclic addition and removal of chemical reagents necessary in NGS approaches. Two commercial SMRT solutions exist to date: SMRT sequencing by Pacific Biosciences (PacBio) and nanopore sequencing by Oxford Nanopore Technologies (ONT). PacBio machines possess millions of small reaction chambers (*Zero-mode waveguides*, ZMWs) with a stationary DNA polymerase molecule fixed to the bottom (see Figure 2.4a) [25]. To initiate sequencing, a single-stranded DNA template is added to the chamber together with fluorescently labelled nucleotides. When the DNA polymerase incorporates complementary nucleotides on the DNA template, fluorescent light flashes are emitted and recorded by a camera revealing the identity of each incorporated nucleotide. The PacBio technology uses circularized DNA templates that allow the DNA polymerase to traverse the template several times. This enables PacBio machines to produce two

different data types: Firstly, the relatively inaccurate raw read sequences termed *continuous long reads* (CLR) from individual passes of the circular template can be used. Secondly, a template can be sequenced multiple times to correct sequencing errors and generate a more accurate *circular consensus sequence* (CCS).

SMRT sequencing by ONT uses a very different approach. Small protein pores (*nanopores*) in a synthetic membrane are utilized that directly sense the composition of the DNA molecule (see Figure 2.4b) [15]. This is made possible through the application of an electric field to the pore and the constant measurement of the field's current. To initiate sequencing, a single-stranded DNA molecule is passed through the pore causing characteristic changes in the current. By analyzing these changes, the DNA sequence can be determined.

Both SMRT sequencing technologies have the same two drawbacks. Firstly, the relatively high error rates of approximately 5-15% complicate many downstream applications [65, 95]. Errors in PacBio reads are randomly distributed and can be easily corrected with a sufficiently high sequencing coverage. Furthermore, the generation of a CCS from multiple sequencing rounds of the same template can correct most errors and reduce the error rate to below 1% [97]. The Nanopore platform, in contrast, suffers from systematic errors in homopolymer regions that are harder to correct than random errors [44]. In both platforms, the majority of errors are indels and error rates steadily decreased in recent years with the introduction of new instrument generations, chemistry updates and software improvements.

The second major drawback of the SMRT sequencing platforms is their high cost compared to short-read sequencing. Although the costs significantly decreased over the last years with the release of new instrument generations, further reductions are necessary for more widespread adoption of both technologies [65].

2.2.4 *Chromosome conformation capture*

Beside long-read sequencing, several other new technologies and protocols enable the detection and characterization of genomic variants. They often complement the existing short-read and long-read

sequencing approaches with their sensitivity for variant classes or sizes that are frequently missed by sequencing approaches.

One of these protocols is called *Hi-C*. Together with other *chromosome conformation capture* (3C) techniques it analyzes the three-dimensional organization of genomic DNA in the nucleus [99]. The folding of chromosomes can bring loci that are distant on the linear genome into close proximity in 3-D space. This spatial organization is closely linked to the biological function of the genome and its investigation with 3C techniques has enabled several important research findings. Most prominently, it has been revealed that the genome is organized into *topologically associating domains* (TADs) which are characterized by a higher interaction frequency between loci in the same TAD relative to loci in different TADs [23].

As the first step of most 3C methods, the chromatin is fixed with formaldehyde so that genomic loci in close spatial proximity are cross-linked (see Figure 2.3b) [99]. Then, the fixed chromatin is cut with a restriction enzyme. The resulting ends of the DNA fragments are ligated to connect the two cross-linked fragments that were originally in close spatial proximity in the nucleus. Then, these connected fragments from interacting loci are quantified with subsequent steps that are specific to the concrete 3C technique.

In the Hi-C protocol, the quantification of interacting loci is performed using paired-end short-read sequencing [60]. This produces read pairs with highly variable genomic distances between the reads reflecting the spatial organization of the chromosome. After alignment of the read pairs to a reference genome, a contact matrix can be constructed from the number of ligations products/read pairs between every combination of genomic regions.

2.2.5 *Optical mapping*

Another method to complement traditional sequencing-based technologies is *optical mapping*. First introduced in 1993, it employs restriction enzymes to produce low-resolution maps of large DNA fragments [80]. A restriction enzyme cuts the DNA only at sites with a specific sequence pattern (*restriction sites*). Mapping the locations

of restriction sites on a DNA fragment therefore produces a unique barcode pattern for the fragment.

Modern optical mapping approaches, such as the *Bionano Genomics* platform, attach fluorescent labels at restriction sites instead of cutting [53]. Then, they stretch the DNA fragments with sizes of up to 1 Mbp into small channels and take images of them. The images reveal the locations of the fluorescently labelled sites and can be transformed into digital representations (barcodes) of the label patterns. Finally, the barcodes can be aligned to each other or to a reference map to produce genome-wide maps.

Optical mapping is widely used in de novo genome assembly because it provides a long-range scaffold for other datasets, for instance from short-read or long-read sequencing [46, 51]. Optical maps can also be compared to detect SVs, such as deletions or insertions, that can cause deviations in the distances between adjacent labels [9, 53]. Unlike sequencing-based approaches, however, optical mapping provides neither base-pair resolution nor the genomic sequence of the fragment.

2.3 FROM SEQUENCING READS TO GENOMIC VARIANTS

Since the widespread adoption of NGS approaches, the volume of sequencing data being generated has seen an unprecedented growth. Sequencing data most commonly consists of sequencing reads, i.e. strings of nucleotide bases that are relatively short when compared to the length of the genome. To analyze these sequencing reads, new bioinformatics methods, tools and protocols were required. Most prominently, read alignment and genome assembly evolved as the two main avenues for the analysis of sequencing reads.

2.3.1 *Read alignment and genome assembly*

Read mapping describes the task of locating the original position of a given read on an existing reference genome. After identifying this position, a sequence alignment can be derived in the next step. The alignment associates each base of a query sequence with a base from a target sequence in such a way that the associated bases are

related or share a common ancestry. In the case of read alignment, the aim is to associate each read base to the reference base it was sequenced from. To find an optimal alignment, alignment algorithms attempt to maximize the number of associations between similar bases (*matches*) and to minimize the number of associations between differing bases (*mismatches*). To account for insertions or deletions in one of the sequences, gaps can be inserted between bases. Thus, read alignment enables a direct comparison between the sequenced genome and the reference genome at a particular genomic locus. The read alignments contain characteristic signatures of genomic variation that can be analyzed by computational tools.

Genome assembly, in contrast, is performed when a suitable reference genome is either lacking or shall not be used in order to conduct a reference-independent analysis. It denotes the task of using overlapping portions of sequencing reads to reconstruct larger units that are denoted *contigs*. The three main goals of genome assembly are to produce a representation of the genome that is a) accurate, b) complete and c) partitioned in as few fragments as possible. Consequently, the ideal assembly has a low error rate, covers the entire genome and consists of large fragments corresponding to full chromosomes.

However, a genome assembly on its own is merely a set of nucleotide sequences. To draw valuable conclusions from the assembly, an annotation of its elements, such as genes and regulatory regions, is needed. The annotation can be performed explicitly using specialized tools or implicitly by comparing the assembly with an existing, already annotated, genome sequence. For species with an existing high-quality reference genome, such as human or mouse, a new genome assembly is routinely aligned to this reference genome. From the resulting genome-genome alignment, SVs can be detected in a similar fashion as from read alignments.

2.3.2 *Computational variant detection*

Read alignment and genome assembly are only intermediate steps towards the characterization of genomic variants. After alignment, the reads need to be compared with the reference genome to extract, filter and interpret signatures of the different variant classes. In

a similar fashion, a new genome assembly can be compared to another assembly or reference genome to identify differences and rearrangements. Over the last years, more than a hundred different tools have been developed for this task of variant detection [73].

Due to their simple structure, SNVs and small indels were the first classes of variation to be comprehensively analyzed using NGS. Their detection has since become a routine task in genetics and genomics studies and is usually divided into two separate steps: Variant calling and genotype calling. While variant calling aims to identify positions where the read bases differ from the reference sequence, genotype calling estimates the zygosity of the variant as well as the exact bases in the maternal and paternal chromosome. Early approaches for both steps used fixed cutoffs to determine when to call a variant or a certain genotype. Recent methods, however, employ probabilistic models to incorporate uncertainty as well as additional information [72]. Results from the 1000 Genomes project which motivated the development of several tools and best-practice workflows show that state-of-the-art methods such as the *Genome Analysis Toolkit* combine high sensitivity and specificity [17, 21]. However, several challenges, for instance in genomic regions with poor mappability, remain.

While NGS has enabled the comprehensive identification of SNVs and small indels, SVs are much harder to detect. One reason is that SVs encompass a diverse range of modifications. While SNVs are simple base pair substitutions, the term *SV* summarizes many different phenomena, such as deletions, insertions, inversions and duplications.

A wide variety of SV calling tools have been developed for different purposes [73]. Most of these callers discover SVs from short paired-end reads. After the reads have been aligned to a reference genome, they examine the alignments for characteristic signatures. There are three common conceptual approaches for SV detection from sequencing reads [2]:

1. **Read depth approaches** analyze the alignment depth across the genome. They search for regions with an elevated read depth caused by duplications and regions with a reduced

depth caused by deletions. Read depth approaches are not able to detect balanced SVs, such as inversions or translocations.

2. **Read pair approaches** analyze the relative position and orientation of mapped read pairs. Almost all SV classes can be detected from their characteristic mapping signatures. Read pairs mapping too far apart, for instance, indicate deletions while those mapping too close indicate insertions. Pairs with discordant mapping orientations are indicative of inversions.

3. **Split-read approaches** analyze reads that have been split and whose segments have been independently mapped to the reference to produce a better overall alignment. Particularly reads from rearranged regions cannot be mapped linearly to the reference and have to be split up. Similar to read pair approaches, the relative distances between read segments as well as their orientations yield information on virtually all classes of underlying SVs.

Over the years, numerous SV callers have been published employing either one or multiple of these conceptual approaches. A recent benchmarking study counted 79 different tools and compared 69 of them on different real and simulated datasets [49]. The authors found that each tool excels in the detection of particular SV classes and sizes but that no single method outperforms the others in all settings.

After the recent introduction of third-generation sequencing technologies, a number of SV callers for long-read datasets have been developed. In the next chapter, we will give a detailed overview of them and will introduce our own method for SV detection from long error-prone sequencing reads.

3

STRUCTURAL VARIANT DETECTION FROM LONG SEQUENCING READS

Despite ongoing efforts, the discovery of SVs from short-read sequencing data remains challenging. Studies have shown that short-read methods suffer from poor sensitivity, particularly for small SVs shorter than 1 kbp [12, 43]. In contrast to SNPs where discovery and sequence resolution can be performed simultaneously within a single sequencing read, SVs are discovered mainly indirectly from signatures in the read alignments. These signatures can be indirect evidence in favor of certain SV classes but are often unable to fully characterize the SV. The main limitation is that most SVs are larger than a single short sequencing read. Furthermore, the accurate detection of SVs is hampered by the big diversity of SV classes, their association with repeat regions and biases in the sequencing technology [10, 42, 98].

Long-read, single-molecule sequencing technologies like those offered by Pacific Biosciences or Oxford Nanopore Technologies are able to overcome many of these challenges by producing reads that are orders of magnitude longer than ordinary NGS reads. Despite the higher error rate and sequencing cost, they offer many advantages for the detection of SVs [19]. The long reads can be mapped with greater accuracy which enables the sequencing of repetitive and low-complexity regions [11, 66]. Unlike with short reads, SVs are often spanned by a single long read. This enables the direct detection and full characterization of the SVs. Consequently, several studies confirmed that a substantial number of SVs that are missed by short-read approaches can be identified with long reads [13, 68]. Yet, available software tools still do not fully exploit the possibilities.

In this chapter, we present *SVIM*, a tool for the sensitive detection and precise characterization of SVs from long-read data. *SVIM* consists of four components for the collection, clustering, combination and genotyping of SV signatures from read alignments. It distin-

guishes six different variant classes including similar types, such as insertions, tandem and interspersed duplications.

3.1 CURRENT METHODS FOR SV DETECTION FROM LONG READS

While short-read sequencing has been applied to the task of SV detection for almost two decades, third-generation sequencing is a more recent development. Nevertheless, several SV callers have been developed for long reads [49].

The first tool designed for the analysis of PacBio data, *PBHoney*, was published in 2014 and implements two different variant identification approaches [26]. The first approach, *PBHoney-Spots*, exploits the stochastic nature of the errors in PacBio reads. It scans read alignments (usually produced by the read aligner *BLASR*) and recognizes SVs by an increase in error and a subsequent decrease in error along the reference sequence. The second approach, *PBHoney-Tails*, analyzes the soft-clipped (i.e. unmapped) read tails from a *BLASR* alignment. It extracts such tails from the *BLASR* output and realigns them to the reference. Then, SVs are detected by clustering the resulting piece-alignments based on their location and orientation.

In 2016, the assembly-based pipeline *SMRT-SV* was published. It first covers the genome with overlapping windows of 60 kbp. Then, it scans PacBio alignments for SV signatures, such as spanned deletions, spanned insertions and soft-clipped read tails, and places additional windows around them [43]. From each window, aligned reads are assembled, polished and realigned to the reference genome. Based on these alignments, SVs are called.

Three years later in 2019, a follow-up study introduced *SMRT-SV2* which follows a similar approach but implements updates for handling data from more recent PacBio machines [3]. Most importantly, it uses an updated version of the aligner *BLASR* and *canu* instead of the *Celera Assembler*. Because *SMRT-SV2* and *SMRT-SV* follow an assembly-based approach, they not only generate SV calls but also assembly contigs containing the SVs. This comes at the cost of a substantially longer runtime and an increased demand in computational resources.

SMRT Link, the official graphical user interface that PacBio ships together with its machines to configure, monitor and analyze sequencing runs also includes a structural variant caller named *pbsv* [8]. It distinguishes five basic SV classes: deletions, duplications, insertions, inversions and translocations. Due to its availability on all PacBio machines, *pbsv* is widely used although the method has not been published in a scientific journal and its source code has not been made open-source. *pbsv* and the other tools introduced above have been developed specifically for PacBio data. Correspondingly, other methods, such as *NanoSV* and *NanoVar*, exist that are specifically designed for Nanopore data [88, 93].

Sniffles, the currently most popular SV caller for long reads was published in 2018. It uses signatures from split-read alignments, high-mismatch regions and a coverage analysis to identify SVs in PacBio or Nanopore data [82]. To overcome the high error rate in the reads, *Sniffles* evaluates candidate SVs based on features such as their size, position and breakpoint consistency. Beside the five basic SV classes deletions, (tandem) duplications, insertions, inversions and translocations it also detects inverted duplications.

In contrast to the previous general methods which enable the detection of several SV types, other more targeted approaches have been published as well. *npInv* and *rMETL*, for instance, are designed specifically for the detection of inversions and mobile element insertions, respectively [45, 84].

Third-generation sequencing approaches have enabled considerable advances towards the accurate detection of the full spectrum of structural variation present in the human population. Nevertheless, available methods do not fully exploit the possibilities. Most importantly, existing methods still suffer from a high error rate [20, 63]. On the one hand, they fail to detect many true SVs (*false negatives*) while, on the other hand, many of the detected SVs are wrong (*false positives*). Both of these error types complicate downstream analysis of the callsets and may make it necessary to analyze orthogonal datasets to confirm and complement the SV calls. Another problem are the inaccurate genotypes estimated for the called SVs. As a consequence, additional genotyping steps with specialized tools or orthogonal datasets are often required to obtain more accurate genotype calls. Furthermore, existing methods either do not distinguish

between tandem and interspersed duplications or they detect tandem duplications only. This leaves interspersed duplications caused by mechanisms such as mobile element insertions unexplored.

3.2 CHALLENGES IN SV DETECTION FROM LONG READS

To detect SVs, most SV calling methods analyze reads that were previously aligned to a reference genome by a read aligner, such as *NGMLR* or *minimap2* [56, 82]. A read aligner attempts to find the best alignment of each read to the reference genome. Simply put, it searches for an alignment with as many consecutive matches and as few mismatches or gaps as possible. In genomic regions that are free of any variant, the reads match perfectly to the reference genome and can be aligned linearly and at their full length yielding only a single alignment segment for each read. In regions harboring variation, however, no perfect alignment exists and discordancies in the read alignment arise. To accommodate smaller differences between a read and the reference, the read aligner can introduce mismatches or gaps (representing inserted and deleted bases) into the alignment. Around larger and more complex variants, such as SVs, the read aligner can also split the read into multiple segments and align each segment independently to the reference. These so-called split alignments allow the aligner to produce good alignments even when there are major rearrangements between the reference genome and the individual genome that the read has been sampled from. Due to split alignments, the BAM file can contain multiple alignment records for a single read, i.e. one record for each aligned read segment.

In Figure 3.1, we schematically visualize the discordant read alignments caused by different classes of SVs. It shows that discordancies in a read alignment can be contained either within alignment segments in the form of mismatches and gaps (*intra-alignment discordancies*) or between alignment segments in the form of discordantly mapped split alignments (*inter-alignment discordancies*). The figure also depicts how some SV classes lead to reads that can be aligned in multiple different ways. Reads containing a tandem duplication (fifth column), for instance, can be either linearly aligned with reference

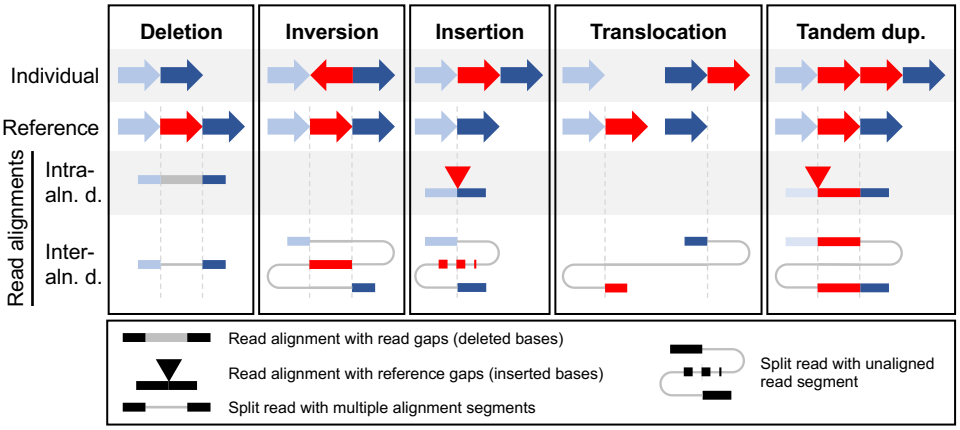
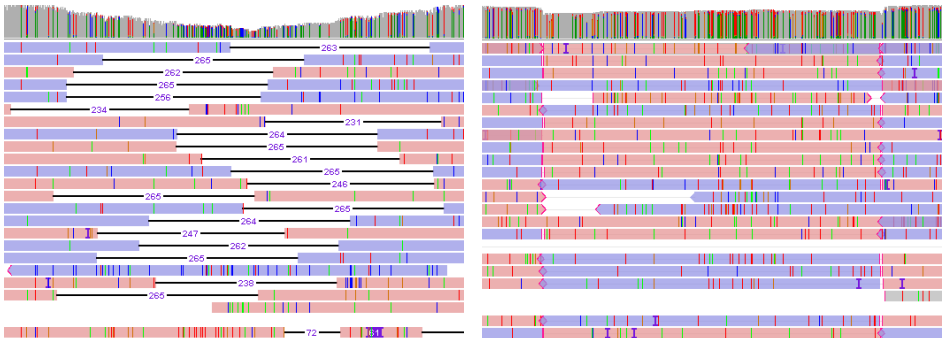


Figure 3.1: **Discordant read alignments across 5 different classes of SVs.** Structural variants between an individual genome (first row) and a reference genome (second row) lead to discordant alignments of reads from the individual genome to the reference. The discordancies can be contained either within a continuously aligned read segment (intra-alignment discordancies, third row) or between independently aligned segments of a read (inter-alignment discordancies, fourth row).
Abbreviations: aln. - alignment, d. - discordancies, dup. - duplication



(a) **Intra-alignment discordancies** (b) **Inter-alignment discordancies**

Figure 3.2: **Discordant alignments of real reads.** Shown are views from a genome browser that visualizes the read alignments as horizontal bars along the reference genome.
a | Long reads are aligned to a genomic region on the forward (red) or reverse strand (blue). Most read alignments show gaps between 231 and 265 bp in length caused by a homozygous deletion. The precise lengths and positions of the gaps vary substantially due to sequencing errors in the reads and alignment ambiguities.
b | To align long reads to this genomic region, many reads had to be split by the alignment algorithm. Most reads are split into three segments and the segment in the middle is aligned to a different strand than the other two segments (visualized by the color). These discordancies indicate the presence of a homozygous inversion.

gaps for the tandem copy (third row) or split into two segments with overlapping positions on the reference (fourth row).

When assessing alignments of real long-read datasets, numerous discordancies can be observed. Most of them are small gaps or mismatches (visualized by colored vertical marks in Figure 3.2) caused by sequencing errors in the reads. Larger discordancies, such as long gaps (Figure 3.2a) or discordant split alignments (Figure 3.2b), can point to the presence of SVs. There is no clear line between small discordancies and those larger ones indicative of SVs. Most SV callers including *SVIM* use a user-configurable threshold defining the minimum size of an SV and ignore all discordancies smaller than this threshold (see Section C.1 in the Appendix).

Another challenge presented by real data is the heterogeneity of the read alignments (see Figure 3.2a). Due to the repetitive nature of the human genome, the same read sequence can often give rise to different equally good alignments. Sequencing errors in the reads further exacerbate the problem by pushing the alignment in a direction where the similarity between the partially erroneous read bases and the reference is maximized. Together, this can lead to very different alignments of reads that were originally sequenced from the same haplotype.

3.3 FOUR STEPS TOWARDS MORE ACCURATE SV DETECTION FROM LONG READS

In this section, we introduce our computational method *SVIM* (*Structural Variant Identification Method*). It analyzes alignments of long reads in BAM format [57] to detect six different classes of SVs. In particular, *SVIM* searches discordant alignments in order to extract SV signatures from them. We define SV signatures as pieces of evidence pointing to the presence of an SV between the sequenced genome and the reference. More formally, an SV signature can be viewed as a quadruple $S = (S.type, S.chrom, S.start, S.end)$ where $S.type$ is one of six different signature types defined below. $S.chrom$ is the chromosome where the SV is located on and $S.start$ and $S.end$ define the genomic start and end position.

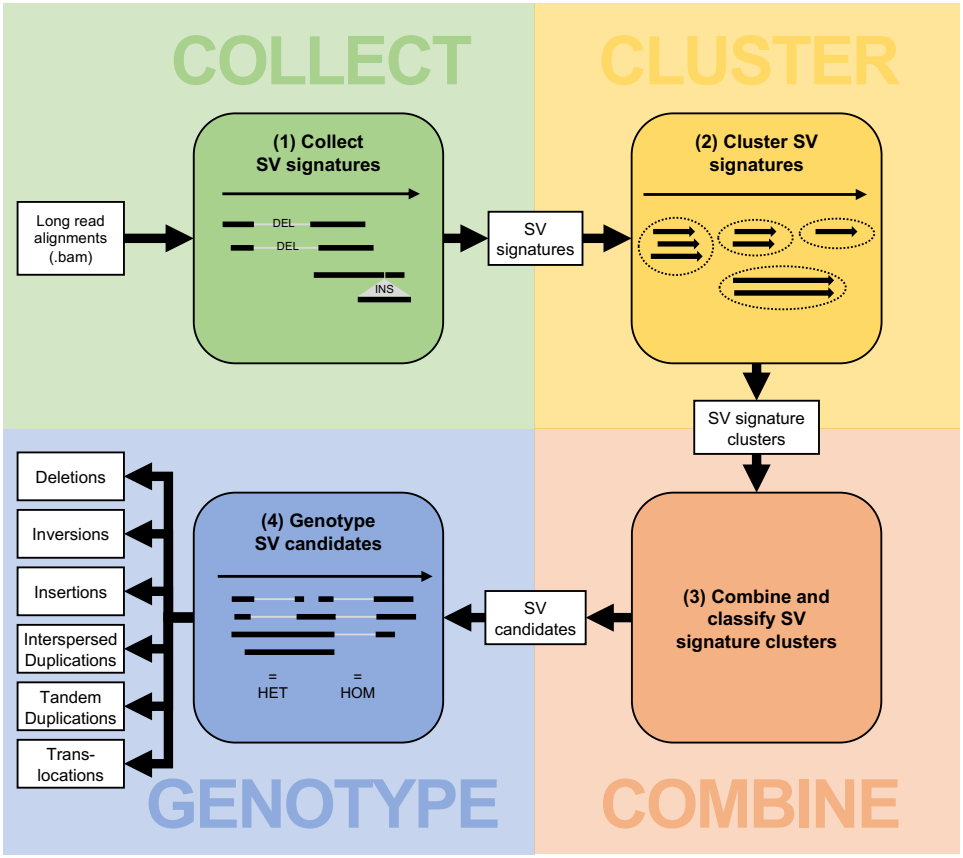


Figure 3.3: **The *SVIM* workflow.** (1) Signatures for SVs are collected from the input read alignments. *SVIM* collects them from within alignments (intra-alignment discordancies) and between alignments (inter-alignment discordancies). (2) Collected signatures are clustered based on their genomic position and span. (3) Different signature clusters are combined to distinguish six different classes of SV candidates: deletions, insertions, inversions, interspersed duplications, tandem duplications and translocations. (4) Read alignments spanning each SV candidate are analyzed to determine the candidate's genotype.

SVIM implements a pipeline of four consecutive components (see Figure 3.3). Firstly, SV signatures are collected from each individual read in the input BAM file (*COLLECT*). Secondly, the detected signatures are clustered using a hierarchical clustering approach and a novel distance metric for SV signatures (*CLUSTER*). Thirdly, the signature clusters are analyzed, combined and classified into six classes of SVs (*COMBINE*). Finally, the SV candidates are genotyped using read alignments in the genomic neighborhood (*GENOTYPE*). Below, we explain the four components in greater detail.

SVIM has been implemented in Python and its source code is available at `github.com/eldariont/svim`. It can be easily installed

via bioconda or the Python Package Index (PyPI) [38]. As input, *SVIM* expects either raw or already aligned reads (in FASTA/FASTQ or BAM format, respectively) as well as a reference genome (in FASTA format).

3.3.1 *COLLECT: Collection of SV signatures from individual reads*

SVIM extracts six different types of SV signatures from the BAM file by analyzing one read at a time. Firstly, the individual alignment segments of each read are scanned for intra-alignment discordancies. This type of discordancy can be retrieved from the *CIGAR* string, a special field in the BAM record. In simplified terms, the *CIGAR* string defines the precise read alignment as a string of matches, mismatches and gaps between the read and the reference. Long gaps in the read or in the reference represent regions that have been deleted from or inserted into the sequenced genome, respectively. They are collected as *deleted region* (DEL) and *inserted region* (INS) signatures, respectively.

Secondly, the alignment segments of each read are scanned for inter-alignment discordancies, i.e. discordant relative alignment positions and orientations among alignment segments. This type of discordancy arises when a read is split to enable a better alignment of its segments to the reference. The termination of one alignment segment and the continuation of the alignment at another genomic position can indicate several different SV classes. Therefore, *SVIM* classifies this type of discordancy in a heuristic fashion to collect the correct type of SV signature. The heuristic procedure that is used to classify inter-alignment discordancies and produces different types of SV signatures is shown as a simplified decision tree in Figure 3.4. Initially, the alignment segments of each read are sorted by their position on the read such that alignment segments involving the first read bases come before segments involving bases later in the read. Then, each pair of adjacent alignment segments, starting with the first and second segment, is compared according to the criteria shown in Figure 3.4. The decision tree has six possible outcomes. Five of them, representing different SV signature types, are explicitly shown in the diagram: (1) *deleted region* (DEL), (2) *inserted region*

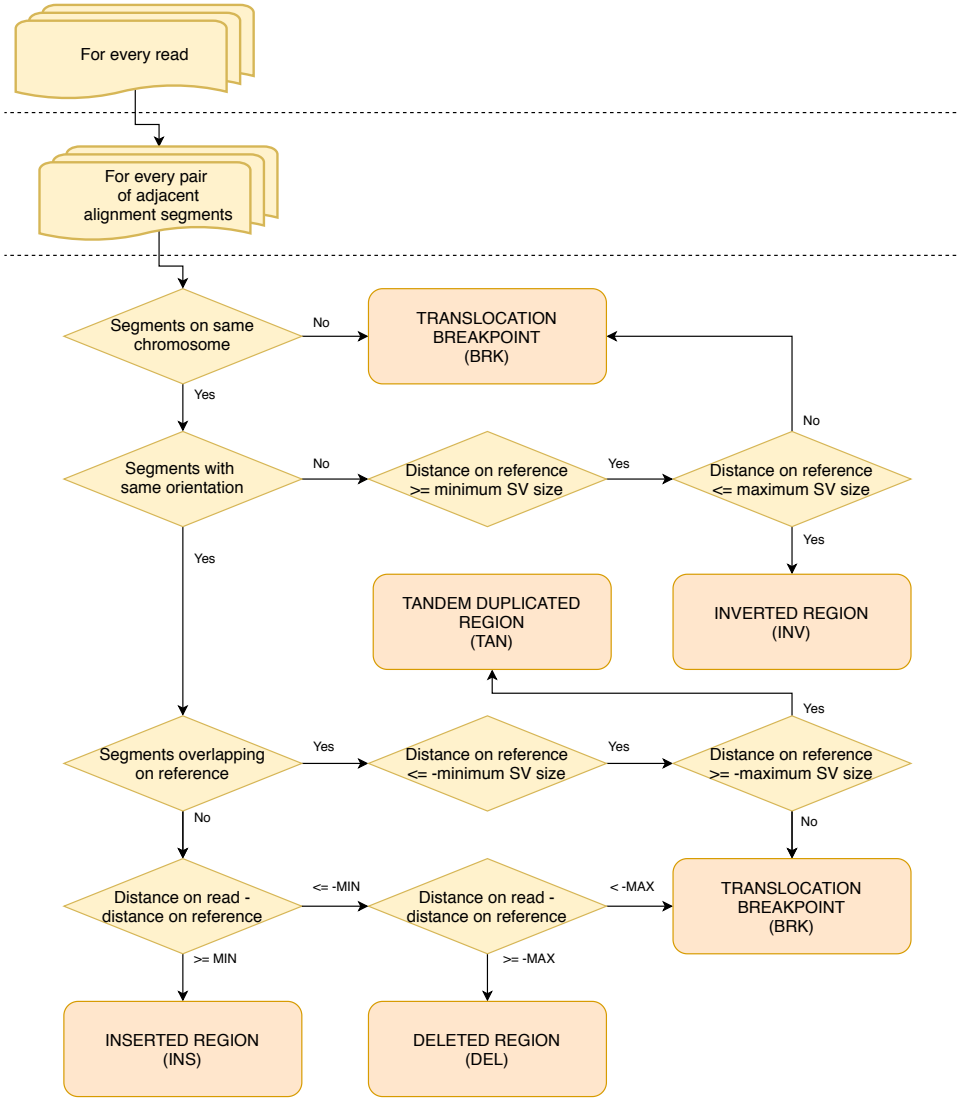


Figure 3.4: **Heuristic decision tree for detecting and categorizing inter-alignment discordances.** The decision tree is entered at the top node. The two boxes at the top represent for-loops such that the lower portion of the diagram is traversed for every pair of adjacent alignment segments in every read. Diamond-shaped boxes represent criteria on the alignment segments and their outgoing edges represent decisions. The decision tree has five explicit sink nodes (rounded rectangles) representing the different types of SV signatures. For clarity, the sixth sink node and all outgoing edges leading to it are not shown explicitly. It captures all remaining cases from which no SV signature is collected.
Abbreviations: MIN - minimum SV size (user-modifiable parameter, see Section C.1, by default 40 bp), MAX - maximum SV size (user-modifiable parameter, see Section C.1, by default 100 kbp)

(INS), (3) *inverted region* (INV), (4) *tandem duplicated region* (TAN) and (5) *translocation breakpoint* (BRK). The sixth outcome of the tree is not shown for the purpose of clarity. It captures all remaining cases from which no SV signature is collected.

After completing this heuristic procedure on a given read, a post-processing step is performed to collect a sixth SV signature type, *interspersed duplicated region* (INT). Reads covering interspersed duplications are often represented by split reads whose middle alignment segment has been aligned at a distance from its adjacent segments (see Figure 3.7). Such reads give rise to two characteristic translocation breakpoints connecting the insertion locus with the start and end of the origin region. Therefore, the translocation breakpoint signatures (BRK) collected for a given read are scanned for this characteristic pattern and if detected, an interspersed duplicated region signature (INT) is collected.

All in all, the first component of the *SVIM* pipeline collects 6 different types of SV signatures: DEL, INS, INV, TAN, BRK and INT. Some of these evidence types (e.g. INV) indicate one particular SV class (inversion). Others could indicate several possible SV classes. An INS, for instance, can indicate either a duplication or a simple insertion.

3.3.2 *CLUSTER: Clustering of SV signatures*

The collection of signatures from the read alignments is only the first step to accurately detect SVs. Because most genomic regions are covered by multiple reads, several signatures of the same variant can be gathered from different reads. In the next step, signatures from multiple reads need to be merged and criteria have to be found to distinguish groups of true signatures from signatures that are merely artifacts caused by sequencing or alignment errors. To achieve this, *SVIM* combines a hierarchical clustering approach with a novel distance metric for SV signatures. The aim is to merge signatures of the same SV even if their positions vary (as in Figure 3.2a). At the same time, signatures from separate SVs need to be kept separate even if the SVs are in close proximity to each other.

The only distance metric that can, to our knowledge, be applied to genomic intervals like SV signatures is the Gowda-Diday distance [37]. It combines (a) the distance between two intervals, (b) their span difference and (c) their degree of overlap into a single numeric distance value. However, the Gowda-Diday distance is not well suited for our type of data derived from error-prone long reads. Due to the abundance of sequencing errors, we sometimes observe little to no overlap between signatures even if they originate from the same SV (see Figure 3.2a). However, signatures from the same SV often possess similar positions and spans.

Therefore, we introduce *span-position distance* as a novel distance metric for SV signatures (see Figure 3.5). For two SV signatures x and y with different type $x.type \neq y.type$ or different chromosome $x.chrom \neq y.chrom$, the span-position distance is set to a very high value. For two SV signatures x and y on the same chromosome and with the same type, however, the span-position distance SPD consists of two components SD and PD: $SPD(x,y) = SD(x,y) + PD(x,y)$. SD is the relative difference in span between both signatures ($SD \in [0,1)$). It is defined as $\frac{|x.span - y.span|}{max(x.span, y.span)}$ where $x.span = x.end - x.start$ and $y.span = y.end - y.start$. PD is the difference in position between both signatures scaled by a user-defined scaling constant. It is defined as $\frac{|x.center - y.center|}{N}$ where $x.center = \frac{x.start + x.end}{2}$ and $y.center = \frac{y.start + y.end}{2}$. N is a user-defined scaling constant which regulates the relative importance of SD and PD (see Section C.1 in the Appendix). In our analyses, setting $N = 900$ returned the best results. Intuitively, this setting means that two signatures that are 900 bp apart ($PD = \frac{900}{900} = 1$) but have the same span ($SD = 0$) would have the same $SPD = 1$ as two signatures with extremely different spans ($SD \approx 1$) but the same position ($PD = \frac{0}{900}$). It is possible to show that the span-position distance fulfills all conditions of a metric (see proof in Section A of the Appendix).

Before performing the actual clustering, we first form coarse partitions of signatures in close proximity. The partitions are formed in an iterative fashion while traversing each chromosome and its signatures from start to end. When a signature is located close to the previous signature, it is added to the same partition. When the distance between a signature and its predecessor exceeds a user-configurable threshold (see Section C.1 in the Appendix), however,

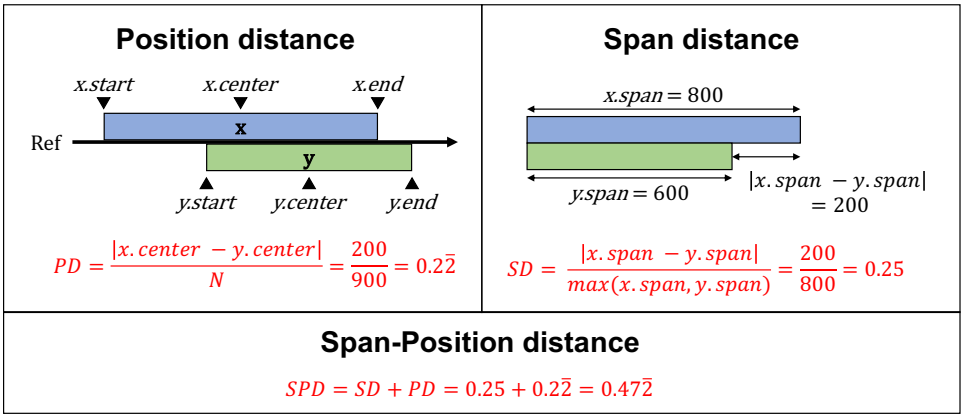


Figure 3.5: **Computing the span-position distance.** This diagram schematically visualizes the computation of span-position distance as the sum of span distance and position distance. The blue and green boxes represent two SV signatures x and y of the same type on the same chromosome of the reference genome. The center positions of the signatures are 200 bp apart yielding a position distance of $0.2\bar{2}$. The span of the smaller signature y is 25% smaller than the span of the larger signature x yielding a span distance of 0.25. This gives a span-position distance of $0.47\bar{2}$ between the two signatures.

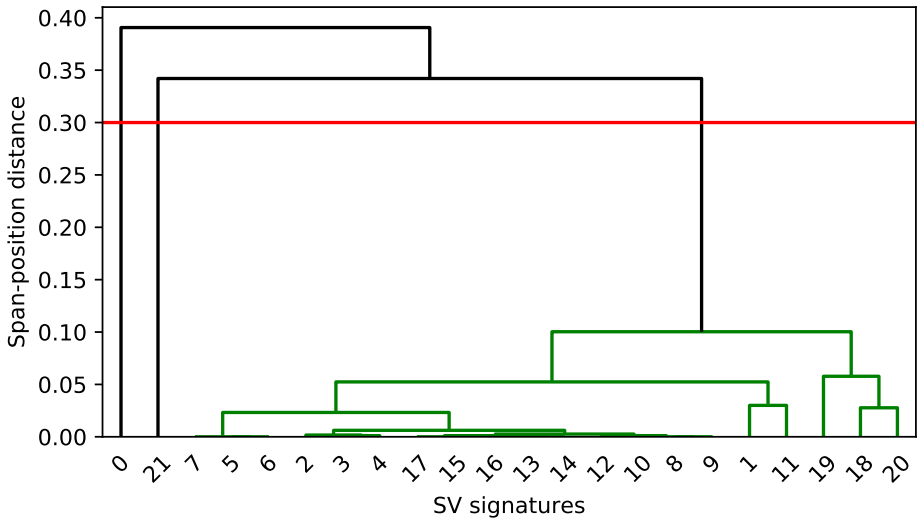


Figure 3.6: **A clustering dendrogram visualizing the tree structure produced by a hierarchical clustering process.** Shown is the clustering dendrogram from the hierarchical clustering of 22 SV signatures (x-axis). Each U-shaped link connects two signature clusters. The height of the link visualizes the average span-position distance between the members of the two clusters (y-axis). A cut (red line) is applied at a user-configurable distance threshold to yield three final clusters: two clusters with a single signature each (0 and 21, respectively) and a large cluster (in green) containing the remaining 20 signatures.

the previous partition is closed and the signature is added to a new partition.

After all signatures have been distributed to coarse partitions, the actual signature clustering is performed on each partition using an hierarchical agglomerative clustering approach. Initially, each signature in a partition starts in its own cluster. In every clustering iteration, the pairwise span-position distances between all pairs of clusters are computed and the two most similar clusters (with the lowest distance) are merged. To compute distances between clusters with multiple signatures, we use the average of all signature-to-signature distances between the two clusters (average linkage). Thus, two clusters are merged in each iteration until finally all signatures have been merged into a single big cluster.

The tree structure generated by the hierarchical merging of clusters can be visualized in a dendrogram (see Figure 3.6). We cut the cluster dendrogram at a user-configurable distance threshold (by default 0.3, see Section C.1 in the Appendix) yielding the final set of signature clusters. Choosing a lower distance threshold would produce many small clusters while a higher threshold results in fewer but larger clusters. Each final signature cluster represents a group of SV signatures that can be jointly assumed to express the same SV in the genome under investigation.

Finally, *SVIM* computes a score $S \in (0, 100]$ for each cluster based on three features:

1. The number $n \in \mathbb{N}$ of signatures in the cluster.

2. A score $s_p \in [0, 1]$ based on the standard deviation s_{pos} of the genomic positions of the signatures in the cluster normalized by their average span.
   $s_p = 1 - min(1, s_{pos} / \overline{span})$

3. A score $s_s \in [0, 1]$ based on the standard deviation s_{span} of the genomic spans of the signatures in the cluster normalized by their average span.
   $s_s = 1 - min(1, s_{span} / \overline{span})$

The three features are combined into a total score $S \in (0, 100]$ with the following formula:

$$S = max(80, n) * (1 + \frac{1}{8} * s_p + \frac{1}{8} * s_s)$$

The score formula puts the main emphasis on the number n of signatures in the cluster but takes at most 80 signatures into account. Clusters with very consistent genomic positions and spans and consequently high standard deviation scores can earn a score bonus of up to 25% ($\frac{1}{8} + \frac{1}{8} = \frac{1}{4}$) relative to the number n of signatures in the cluster. The maximum score that can be reached for clusters with $n \geq 80$ and $s_p = s_s = 1$ is 100. Thus, we obtain a score to discern trustworthy signature clusters from artifacts, such as sequencing or alignment artifacts. Trustworthy events are characterized by many supporting signatures that exhibit a high concordance of their genomic positions and spans.

3.3.3 *COMBINE: Combination and classification of SVs into six SV classes*

The third component in the workflow analyzes and combines the SV signature clusters to classify events into the final six SV classes: deletions, inversions, insertions, tandem duplications, interspersed duplications and translocations. Three types of signature clusters (INV, DEL and TAN) are directly reported as inversions, deletions and tandem duplications, respectively. For the other three types of signature clusters (INS, INT and BRK) a separate analysis is performed.

The main motivation for this analysis is that interspersed duplications can be represented by three different types of SV signatures (see Figure 3.7):

1. Interspersed duplicated regions (INT) from reads that cover the entire duplication

2. Inserted regions (INS) that mark only the insertion location of the additional copy

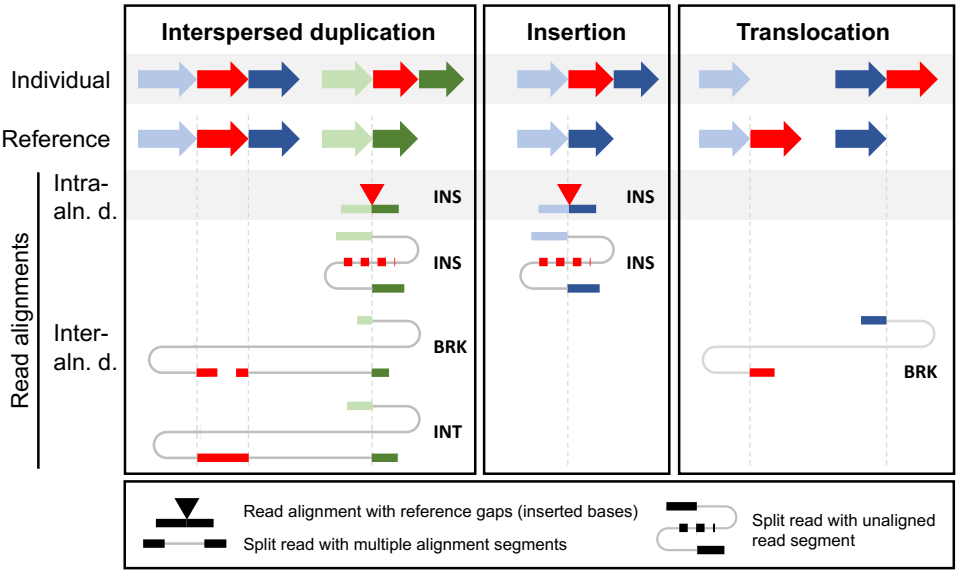


Figure 3.7: **Discordant read alignments from interspersed duplications, insertions and translocations.** Structural variants between an individual genome (first row) and a reference genome (second row) lead to discordant alignments of reads from the individual genome to the reference. Interspersed duplications (left column) can lead to discordancies that are very similar to those from insertions (middle column) and translocations (right column). In particular, signatures of inserted regions (INS) and translocation breakpoints (BRK) require further analysis to determine the correct SV class.
Abbreviations: aln. - alignment, d. - discordancies

3. Translocation breakpoints (BRK) from reads that cover only parts of the duplication

The last two signature types, INS and BRK, can also indicate the presence of insertions and translocations, respectively. Which type of signature can be found around an interspersed duplication is determined by the sequence of each read and decisions made by the read aligner. This again illustrates the problem posed by alignment ambiguities and overlaps between different SV classes.

To classify SVs correctly, *SVIM* makes the following distinctions:

- Interspersed duplication signature clusters (INT) are reported as interspersed duplications.
- Inserted region signature clusters (INS) that are connected to other genomic regions by matching translocation breakpoints (BRK) are reported as interspersed duplications.

- The remaining inserted region signature clusters (INS) are reported as simple insertions.
- The remaining translocation breakpoints (BRK) are reported as translocations.

3.3.4 *GENOTYPE: Genotyping of SV candidates*

The last stage in the *SVIM* workflow estimates the genotype of each SV candidate using read alignments in close proximity. First, all read alignments overlapping a window extended by 1 kbp upstream and downstream around a given SV are retrieved. Then, two read sets are created: 1) the set V of reads supporting the variant and 2) the set R of reads supporting the reference allele. The set of variant reads V contains the reads that gave rise to the SV candidate, i.e. those that were clustered together in the CLUSTER stage to form the SV candidate. The set of reference reads R is compiled as follows.

Each retrieved read alignment that is not a member of V is analyzed with respect to the concrete SV type. If the alignment overlaps the SV breakpoints in any way that conflicts with the hypothesis of the read containing the SV, the read is added to the set of reference reads R. For insertions and duplications, such a conflict arises when the read is aligned over the insertion location without any gaps. For deletions and inversions, a conflict arises when a read alignment extends far into the presumably deleted or inverted region.

Finally, the fraction $F = \frac{|V|}{|R|+|V|}$ of supporting reads is computed and compared against thresholds that can be modified by the user (see Section C.1 in the Appendix). If the support fraction is high (by default $F \geq 0.8$), the SV is called homozygous. If it lies around 50% (by default $0.8 > F \geq 0.2$), the SV is called heterozygous. And if the support fraction is very low (by default $F < 0.2$), the variant is called homozygous reference and filtered out. Homozygous reference SVs are supported by only a few reads while the majority of reads in the region supports the reference allele. Although they could be real SVs, most of them are artifacts caused by spurious read alignments.

```
#CHROM POS ID REF ALT QUAL FILTER INFO FORMAT HG002

1 33005341 svim.INS.741 N <INS> 48 PASS SVTYPE=INS;END=33005341;
SVLEN=224;SUPPORT=39;STD_SPAN=1.17;STD_POS=0.77 GT:DP:AD 1/1:39:0,39

                                                          read depth
1 33156762 svim.DEL.430 N <DEL> 21 PASS SVTYPE=DEL;END=33156938;   reference reads
                                                             variant reads
SVLEN=-176;SUPPORT=17;STD_SPAN=0.24;STD_POS=2.06 GT:DP:AD 0/1:29:12,17
                                                                  genotype

1 33516612 svim.DUP_INT.1 N <DUP:INT> 14 PASS SVTYPE=DUP:INT;END=33519096;
SVLEN=2484;SUPPORT=21;STD_SPAN=2.22;STD_POS=1.11 GT:DP:AD 0/1:32:11,21

1 197756789 svim.INV.19 N <INV> 22 PASS SVTYPE=INV;END=197757984;
SUPPORT=19;STD_SPAN=0.48;STD_POS=0.23 GT:DP:AD 0/1:34:15,19

                                                                  copy number
1 236876893 svim.DUP_TANDEM.41 N <DUP:TANDEM> 13 PASS SVTYPE=DUP:TANDEM;     |
END=236878365;SVLEN=1472;SUPPORT=11;STD_SPAN=0.6;STD_POS=0.3 GT:CN:DP:AD ./.:3:.:.,.

1 248572008 svim.BND.9276 N [1:248796343[N 12 PASS SVTYPE=BND;SUPPORT=10;
STD_POS1=17;STD_POS2=. GT:DP:AD ./.:.:.,.
```

Figure 3.8: **Examples of SV calls in VCF.** This example contains six VCF records of six different SV types (red). The first line starting with "#" is a header line naming the 10 tab-separated columns of the file. The first and second fields give the start chromosome and position of the SV (orange). Some SV classes also require an end position (also orange) and the length of the SV, i.e. the length difference between reference and variant allele (blue). For each SV, the support, i.e. the number of supporting reads, and a quality score (both green) are given. The last field (purple) contains the genotype, the number of reads covering the SV (the read depth) and the number of reads supporting the reference and variant allele. For tandem duplications, the last field also contains the estimated copy number.

3.4 REPRESENTATION OF SVS IN THE VARIANT CALL FORMAT

SV calls can be stored in different file formats. The most naive approach would be to use non-standardized tabular text files that store the different properties of SVs (e.g. chromosome, start and end position, SV class) in separate columns. Alternatively, the calls could be stored as genomic intervals in Browser Extensible Data (BED) format. The BED format is standardized but because it was not designed for storing variant calls it lacks specific fields required to define variants precisely. To facilitate the combination and comparison of variant callsets from different sources, several standardized formats have been developed, such as the Genome Variation Format (GVF) and the Variant Call Format (VCF) [18, 78].

The VCF has been developed for the 1000 Genomes project and has since been adopted as the mainstream format for calls of short

and large genomic variants [18]. VCF files consist of a header section containing an arbitrary amount of meta-information and a data section containing the actual variants. In the data section, each variant is encoded in a single line (see Figure 3.8). Each line consists of 8 mandatory fields plus optional fields to store the genotype information of an arbitrary number of samples. The eight mandatory fields are:

- *CHROM*: chromosome or contig name
- *POS*: start position of the SV on CHROM
- *ID*: SV identifier
- *REF*: reference sequence or N for undefined
- *ALT*: variant sequence or symbolic allele, such as `<DEL>`
- *QUAL*: quality score
- *FILTER*: PASS for good calls or list of failed filters
- *INFO*: semicolon-separated list of fields containing additional information

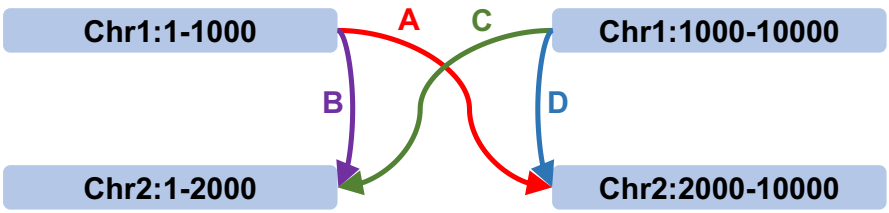
The VCF allows two different ways of defining SVs: Firstly, variants can be characterized using sequence alleles, such as `REF=ATTGGTAGTAGC, ALT=A`, to define an 11 bp-deletion of the bases `TTGGTAGTAGC` after `A`. Alternatively, variants can be characterized using symbolic alleles, such as `REF=N, ALT=<DEL>`, to define a deletion. Because symbolic alleles lack important information such as the length and end position of the SV, this information needs to be given in the *INFO* fields *SVLEN* and *END*.

Each SV class is defined by a slightly different set of coordinates and fields. Deletions and inversions are characterized by a chromosome (CHROM), a start position (POS) and an end position (INFO/END). Insertions are characterized by a chromosome (CHROM), an insertion location (POS) and the insertion length (INFO/SVLEN). Duplications are characterized by a chromosome (CHROM), a start position (POS) and an end position (INFO/END). Tandem duplications additionally have a copy number (CN).

Translocations can be expressed in a VCF file using the breakend (BND) notation. This notation represents a translocation as a set of *novel adjacencies*, i.e. connections between two genomic loci in the sequenced genome that are distant in the reference genome. Each adjacency $N = (B_1, B_2)$ joins two breakends $B_1 = (P_1, O_1)$ and $B_2 = (P_2, O_2)$ which are defined as genomic positions P with associated strand orientations $O \in \{forward, reverse\}$. Because each breakend in a novel adjacency has an associated orientation, there are four possible combinations of orientations $\{forward, reverse\} \times \{forward, reverse\}$ in an adjacency (see Figure 3.9).

SVIM prints all detected SVs into a single output file in VCF. It characterizes the SVs using symbolic alleles and the fields defined above. For each detected translocation, it outputs two VCF records in BND notation - one for each breakend in the novel adjacency.

After we have described our own computational method for the detection of SVs from long-read alignments, the same principles will be adapted in the next chapter for the detection of SVs from genome assemblies.



BND	Orientations (O1, O2)	CHROM	POS	REF	ALT
A	(forward, forward)	Chr1	1000	N	N[Chr2:2000[
B	(forward, reverse)	Chr1	1000	N	N]Chr2:2000]
C	(reverse, reverse)	Chr1	1000	N	]Chr2:2000]N
D	(reverse, forward)	Chr1	1000	N	[Chr2:2000[N

Figure 3.9: **Schematic view of the four possible combinations of breakends in a novel adjacency.** Shown are chromosomes 1 and 2 which are broken at positions chr1:1000 and chr2:2000, respectively. There are four possibilities A to D for connecting the breakpoint at chr1:1000 with the breakpoint at chr2:2000. The table below contains the four VCF fields CHROM, POS, REF and ALT for each combination.

4

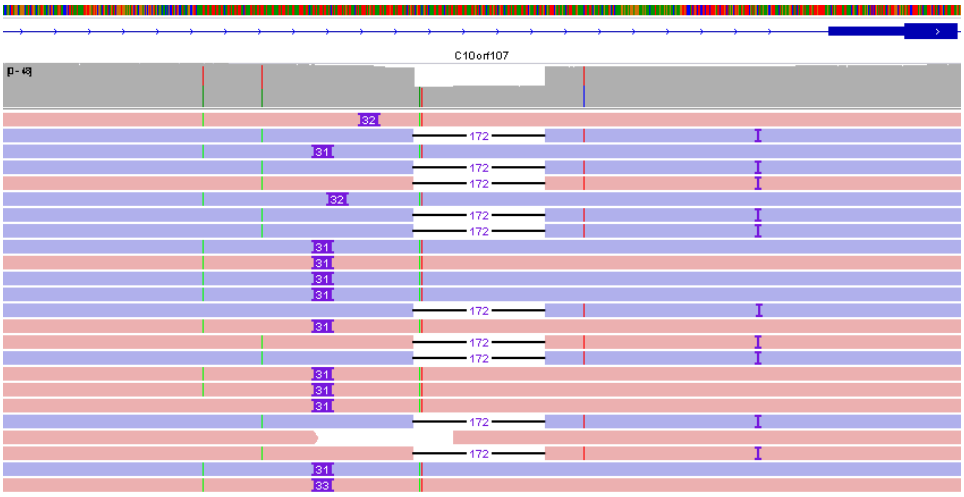
STRUCTURAL VARIANT DETECTION FROM GENOME ASSEMBLIES

The alignment of sequencing reads to a reference genome followed by variant calling is only one way of detecting genomic variants in a sample (see Figure 1.1). Another approach is to assemble the sequencing reads into larger units and to compare the resulting assembly contigs to a reference genome. In this chapter, we first give more detailed background on genome assembly in diploid organisms. Then, we introduce a modified version of the *SVIM* pipeline for the detection of SVs from haploid and diploid genome assemblies.

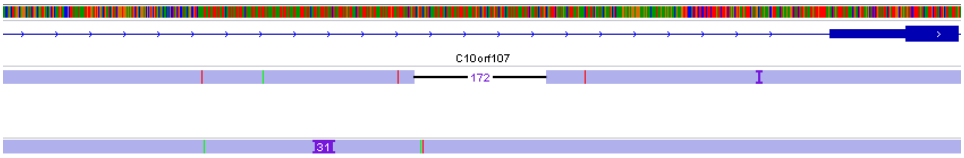
4.1 ADVANTAGES OF GENOME ASSEMBLY FOR SV CALLING

Genome assembly possesses several advantages over read alignment. Most importantly, the assembly process uses only overlaps between the sequencing reads and is therefore independent of the reference genome. Read alignment, in contrast, suffers from biases towards sequences matching the reference and away from those with substantial differences [22]. Consequently, reads that carry larger variants or novel sequence are harder to map to the reference than reads similar to the reference [31]. Another problem arises from the considerable genomic difference between different human populations. Because the linear reference genome can only represent one of numerous human haplotypes, samples from distant populations suffer from poor mappability.

To avoid these biases, genome assembly can be a good alternative. However, it usually requires a higher read coverage and substantially more computational resources than read alignment. Additionally, read alignment simplifies the detection and reporting of genomic variants because the reference genome offers a standardized coordinate-system. Genomic variants can be located by numbered reference bases and are expressed in terms of differences to the



(a) **Read alignments**



(b) **Genome assembly alignments**

Figure 4.1: **Alignments of sequencing reads versus genome assemblies.** Both sequencing reads (panel a) and contigs from a genome assembly (panel b) can be aligned to a reference genome.
a | Several long reads are aligned to a reference genome on the forward (red) or reverse strand (blue). Approximately half of the reads belong to one parental haplotype containing a 172 bp deletion while the other half belong to the other parental haplotype containing an insertion of approximately 31 bp instead.
b | A diploid genome assembly consisting of two sets of contigs is aligned to the same genomic region. One contig (upper alignment) represents the haplotype carrying the deletion while the other contig (lower alignment) represents the haplotype with the insertion.

reference sequence. Genome assembly, in contrast, produces a set of nucleotide sequences (contigs) without any inherent ordering or structure. Therefore, it is common practice to align the assembly contigs to a reference genome to obtain information on similarities and differences. Conceptually, such a genome-genome alignment is very similar to a read alignment (see Figure 4.1). But because the assembly contigs are much longer than a single read and usually span both similar and divergent regions, the impact of the mapping bias is greatly reduced.

4.2 HAPLOID AND DIPLOID GENOME ASSEMBLY

Like many other species, humans are diploid organisms, i.e. their cells possess two sets of chromosomes that are inherited from the mother and the father, respectively. The two copies (or haplotypes) of each chromosome usually differ to a certain degree so that genomic variants fall into two categories: homozygous variants present in both haplotypes and heterozygous variants present in only one haplotype. Genome sequencing approaches generate a mixture of sequencing reads from the two copies of each chromosome. When the reads are aligned to a reference genome, the two groups of reads become visible in regions harboring heterozygous variation (see Figure 4.1a). Then, approximately half of the reads deviate from the reference while the other half does not. In regions with homozygous variation or regions without variation, all the reads agree in either matching or deviating from the reference.

In the realm of genome assembly, however, the diploid nature of the human genome had to be disregarded for a long time due to technical challenges [65]. Despite numerous differences, the chromosomal haplotypes inherited from the two parents are highly similar and contain long stretches of identical sequence. This makes it very difficult to separate sequencing reads from the two haplotypes during the genome assembly process. The separation of the two haplotypes is further complicated by the presence of sequencing errors in the reads. Therefore, until recently, the majority of genome assembly algorithms produced only one (haploid) set of chromosomes, effectively collapsing the two copies of each chromosome into one haploid sequence [65].

Over the last few years, advances in sequencing technology have enabled the diploid assembly of species with even low heterozygosity, such as human. Several diploid assembly tools have been developed, each using different sequencing technologies to separate the two haplotypes: *FALCON-Unzip* uses the phasing information contained in PacBio reads [14]. *FALCON-Phase* and *DipAsm* combine PacBio data with Hi-C while a pipeline by Porubsky et al. combines PacBio and Strand-Seq [30, 52, 75]. Another method, *TrioCanu*, uses parental short-read data to separate long reads from the two haplotypes [48].

4.3 CURRENT METHODS FOR SV DETECTION FROM GENOME ASSEMBLIES

Despite the increased number of high-quality genome assemblies there are only a few tools so far that detect SVs from genome assemblies: In 2015, *AsmVar* was the first tool to discover and genotype SVs in genome assemblies on a population scale [62]. When applied to de novo assemblies of 10 Danish trios and seven other human individuals, it detected more than 3 million SVs between 1 bp and 50 kbp in length. *AsmVar* categorizes variants into five broad classes: deletions, insertions, inversions, simultaneous gaps and translocations. Another method called *Assemblytics* is a simple web application for the detection of SVs in genome assemblies [71]. It produces an interactive visualization of the results and discovers three types of SVs: simple indels, indels in tandem repeats and indels in repeat regions. *SyRi*, the most recent tool, works in two phases: it first searches for structural differences in the assemblies, such as inversions, translocations or duplications. Then, it detects smaller sequence differences like SNVs and indels [34].

All three tools are designed for haploid genome assemblies. As input, they expect two sets of genome sequences: a haploid query assembly and a haploid reference assembly. In their initial step, the tools align the query sequences to the reference sequences using an alignment software: *AsmVar* uses *LAST* while both *Assemblytics* and *SyRi* use *nucmer* for this step. Then, the alignments are analyzed to detect SVs.

For the detection of SVs in diploid genome assemblies, only one SV caller is available to date: The *DipCall* pipeline analyzes *minimap2* alignments to detect SVs from diploid genome assemblies [58]. After filtering the alignments based on mapping quality and alignment length, variants are called separately for each haplotype and later merged into a set of SVs with genotypes. *DipCall* produces two output files: a VCF file containing the SVs and a BED file containing confident genomic regions. *DipCall* detects only two classes of SVs: deletions and insertions. To our knowledge, there is still no tool available for the detection and genotyping of other SV classes, such as inversions and duplications, from diploid genome assemblies.

4.4 ADAPTING THE SVIM PIPELINE FOR ACCURATE SV DETECTION FROM GENOME ASSEMBLIES

In this section, we introduce our computational method *SVIM-asm* (*Structural Variant Identification Method for Assemblies*). Although its workflow is similar to that of *SVIM*, several adaptions have been made to consider the unique properties of assembly alignments as opposed to read alignments (see Figure 4.2). *SVIM-asm* implements two different pipelines for haploid and diploid genome assemblies, respectively. Like *SVIM*, it has been implemented in Python, can be easily installed via bioconda and its source code is available at `github.com/eldariont/svim-asm`.

Diploid genome assemblies consist of two sets of contigs, one for each chromosomal haplotype. Consequently, *SVIM-asm* expects two input BAM alignment files as input for this type of assembly. In the first step of the pipeline (COLLECT), SV signatures are collected separately for each haplotype from individual contig alignments with the methodology described in Section 3.3.1. In the second step (PAIR), signatures from opposite haplotypes are compared and paired up if sufficiently similar. Paired signatures from the two haplotypes are merged into homozygous SV candidates while variants without a partner on the other haplotype are called as heterozygous SV candidates (GENOTYPE). Finally, the genotyped SVs are written out in VCF as members of one of six SV classes (OUTPUT).

In contrast to their diploid counterparts, haploid assemblies are comprised of only a single set of contigs. For diploid organisms, this set often represents a mixture of the two haplotypes. Due to the missing second haplotype, it is not possible to estimate genotypes from haploid genome assemblies which simplifies the pipeline considerably. After the same first step (COLLECT) is applied to the assembly alignments, the PAIR and GENOTYPE steps are skipped for haploid assemblies and the detected SV signatures can be written out immediately (OUTPUT).

While some parts of the *SVIM-asm* pipeline are similar to parts already described in Section 3.3, the PAIR step has been added and the GENOTYPE stage has been modified. In the PAIR step, similar signatures from opposite haplotypes are matched. For this purpose, a clustering approach similar to that in *SVIM* (see Sec-

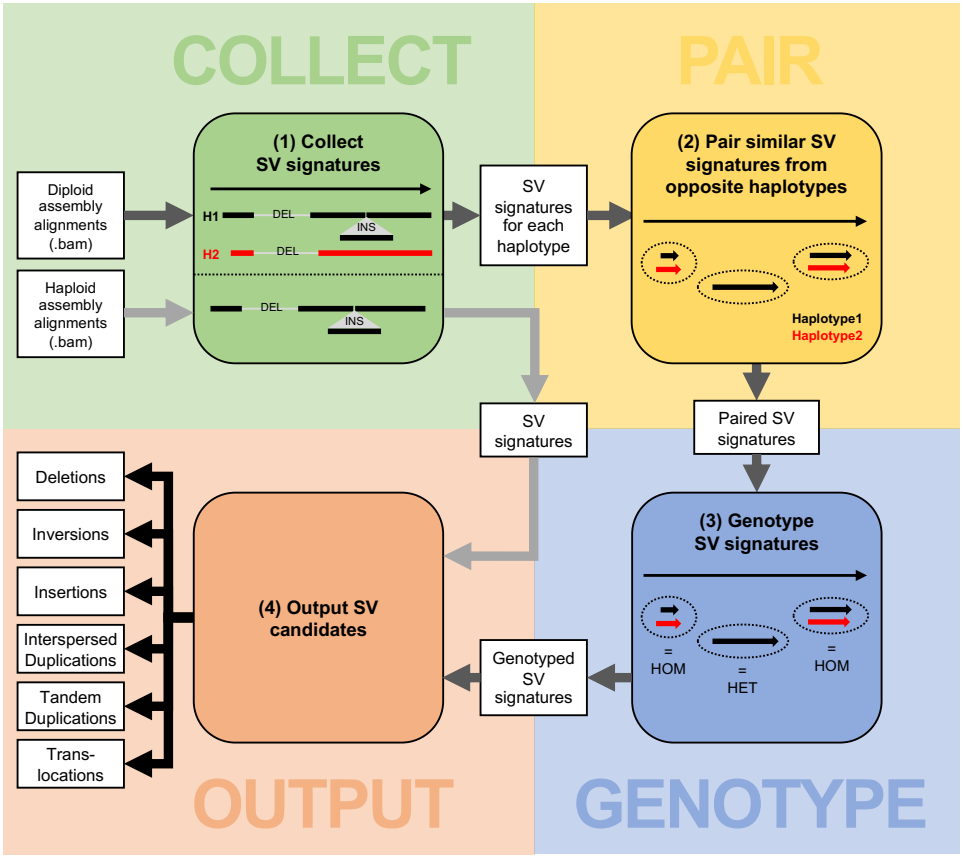


Figure 4.2: **The *SVIM-asm* workflow.** Signatures for SVs are collected from the input assembly alignments (COLLECT). Depending on the ploidy of the assembly, signatures are collected from a single haplotype (haploid assembly) or a pair of haplotypes (diploid assembly). For diploid assemblies, SV signatures from the two haplotypes are compared and similar signatures are paired up based on the edit distance between their haplotype sequences (PAIR). Paired SV signatures represent homozygous (HOM) SVs while isolated SV signatures from only one of the haplotypes represent heterozygous (HET) SVs (GENOTYPE). Finally, six different classes of SV candidates are written out in VCF: deletions, insertions, inversions, interspersed duplications, tandem duplications and translocations (OUTPUT). SV signatures from haploid assemblies skip steps PAIR and GENOTYPE and are written out directly after the COLLECT step.

tion 3.3.2) is applied on the signatures from both haplotypes. Firstly, coarse partitions of signatures in close proximity are formed. Then, the haplotype sequences for all signatures in a given partition are generated. The haplotype sequence $hap(R, S)$ of an SV signature S and a reference genome sequence R is the nucleotide sequence formed by applying the genomic rearrangement defined by S to R. If $s = (DEL, chr1, 10100, 10200)$, for instance, then $hap(GRCh37, s)$ is

the nucleotide sequence that forms when the bases 10100 through 10200 in chromosome 1 are removed from version 37 of the human reference genome. Now, for every pair $x, y \in S$ of signatures in a given partition, the edit distance (Levenshtein distance) $E(hap(R,x), hap(R,y))$ between the haplotype sequences of x and y is computed. Edit distance is defined as the minimum number of operations (deletion, insertion or substitution of one character) required to transform one haplotype sequence into the other. To compute the edit distance between two signatures, it is sufficient to generate $hap(R,x)$ and $hap(R,y)$ for the genomic context around x and y and not the entire genome. In our implementation, we generate both haplotype sequences for the genomic interval $[min(x.start, y.start) - 100, max(x.end, y.end) + 100]$ and use the library *edlib* for the computation of the edit distance [86]. To prevent that signatures from the same haplotype are matched, we enforce a very large distance instead of the actual edit distance between signatures from the same haplotype. Based on the computed distances between their haplotype sequences, the signatures in each partition are clustered using an hierarchical agglomerative clustering approach. With a low distance threshold for cutting the dendrogram (see Figure 3.6), it is ensured that only very similar signatures (i.e. signatures with similar haplotype sequences) from different haplotypes are clustered together.

Based on these clusters, the genotype estimation carried out in the GENOTYPE step becomes trivial. Clusters of two signatures (from opposite haplotypes) represent homozygous variants while clusters of only a single signature represent heterozygous variants. Clusters of more than two signatures cannot form due to the large distance enforced between signatures from the same haplotype.

Together, *SVIM* and *SVIM-asm* form a versatile toolset for the detection of SVs from different long-read datasets. It can be applied to raw long reads generated with PacBio or Nanopore technology as well as haploid or diploid genome assemblies. In the next chapter, we will evaluate both tools using simulated and real datasets.

5

EVALUATION

In the previous chapters, we introduced two computational methods, *SVIM* and *SVIM-asm*, for the detection of SVs from long reads and genome assemblies, respectively. To ensure that these two methods work as intended and produce accurate results, a comprehensive evaluation on different datasets is necessary. In this chapter, we evaluate *SVIM* and *SVIM-asm* on both simulated and real datasets and compare them to existing state-of-the-art tools. For each of the two methods, we first describe the datasets, metrics and methods used for the evaluation. Then, we show the results of the evaluation.

5.1 EVALUATION ON LONG-READ ALIGNMENTS

We compared our tool, *SVIM* (v1.4.1), to two other SV detection methods: *Sniffles* (v1.0.11) and *pbsv* (v.2.3.0) [8, 82]. We chose these two tools because they are versatile computational methods for the detection of several SV classes from long-read sequencing data. Furthermore, they are actively maintained and widely used across the community [20, 63]. While *pbsv* is designed for PacBio data only, *SVIM* and *Sniffles* support both PacBio and Oxford Nanopore data.

We did not compare against short-read SV callers because they have previously been shown to exhibit lower recall than methods relying on long reads [13, 82]. We also did not compare against *SMRT-SV2* because it is not a stand-alone tool. Instead it is a software pipeline applying several existing tools in succession. Moreover, it detects only three SV classes (deletions, insertions and inversions) and is computationally much more demanding than pure alignment-based tools.

We performed comprehensive benchmarks on two types of data. Firstly, we generated a simulated genome from which we sampled in-silico PacBio sequencing reads with known SVs. This provided us with a complete set of fully characterized SVs for evaluation.

Secondly, we used three publicly available sequencing datasets from the latest PacBio and Nanopore sequencers.

For *Sniffles* and *SVIM*, we aligned the reads with *minimap2* (v2.17-r941). For *pbsv*, reads were aligned with *pbmm2* (v1.2.1), a software wrapper around *minimap2* developed by PacBio specifically for their sequencing platform. As reference genome, we used *hg19* for the simulated datasets and *GRCh37* with decoy sequences (*hs37d5*) for the real datasets, respectively.

5.1.1 *Evaluation metrics*

We computed *precision*, *recall* and *F1 score* of the three software tools by comparing their SV calls (*comparison calls*) with a truth set of SVs (*base calls*) for the given dataset. The calls of a given SV caller can fall into three categories:

- *True positives* (TP) are comparison calls that are also contained in the truth set. These are SVs that were correctly called by the tool.
- *False positives* (FP) are comparison calls that are not contained in the truth set. These are SVs that were incorrectly called by the tool.
- *False negatives* (FN) are base calls that are not contained in the set of comparison calls. These are SVs that were missed by the tool.

The Precision $Prec = \frac{TP}{TP+FP}$ is defined as the fraction of comparison calls that are also contained in the truth set. It therefore represents the fraction of correct comparison calls over all comparison calls. Recall (also known as sensitivity or true-positive rate) $Rec = \frac{TP}{TP+FN}$ is defined as the fraction of SVs in the truth set that are also contained in the comparison callset. It therefore represents the fraction of detected base calls over all base calls. The F1 score $F1 = \frac{2*Prec*Rec}{Prec+Rec}$ is the harmonic mean of precision and recall. It can be used to measure the general SV detection performance because it combines both precision and recall into a single value.

To compare the callsets and compute precision, recall and F1 score, we used the tool *Truvari* (v1.3.4) [32]. It matches a comparison call

with a base call if both have the same type, are less than 1 kbp apart and have a difference in span of less than 30% (i.e. $\frac{|span_1 - span_2|}{max(span_1, span_2)} < 0.3$).

As expected, recall and precision reached by a tool can change substantially with the parameters given to it. For SV calling, thresholds on the score, support or confidence of calls have the biggest impact because they determine which calls are reported and which are filtered out. More relaxed thresholds (those yielding more SVs) increase the recall but decrease the precision while stricter cutoffs achieve the opposite. Consequently, we ran all three tools with different settings for their most important confidence threshold: For *SVIM* we applied different score cutoffs (1 to 60). *Sniffles* was run with different settings of the *min_support* parameter (1 to 60). For *pbsv*, we varied the *call-min-reads-one-sample* and *call-min-reads-all-samples* parameters (both 1 to 60). In the precision-recall plots below, we visualize the performance of the tools by plotting each parameter setting as a distinct point.

Beside the confidence threshold, we used the default setting for all other tool parameters if possible. Only a few parameter settings had to be changed in order to ensure that the three tools ran with comparable settings. The complete commands used to call SVs in the context of this evaluation are displayed in Section C.2 of the Appendix.

5.1.2 *Simulated datasets*

To generate the simulated datasets, we simulated 400 homozygous SVs by altering the sequence of chromosomes 21 and 22 in the hg19 reference genome. More precisely, we implanted 100 deletions, 100 inversions, 100 tandem duplications and 100 interspersed duplications with the R package *RSVSim* (v1.24.0) [4]. The package estimates the distribution of SV sizes from real datasets and takes the association of SVs to various kinds of repeats into account. The resulting genome contained SVs between 50 bp and 100 kbp in size. Subsequently, reads were simulated from this genome with the tool *SimLoRD* (v1.0.4) to generate 5 different datasets with coverages of

5x, 15x, 30x, 45x and 60x [91]. *SimLoRD* imitates the error model of SMRT reads to simulate realistic PacBio reads.

To simulate heterozygous SVs, we adapted the previously described approach only slightly. Instead of sampling all reads from the altered reference genome, half of the reads were sampled from the original reference genome. Consequently, reads from the original reference genome and the altered genome each amounted to 50% of the total coverage.

The comparison between different tools was complicated by the fact that each tool defines and detects slightly different SV classes. *pbsv* is able to detect deletions, insertions, inversions, tandem duplications and translocation breakpoints. *Sniffles* is additionally capable of identifying inverted duplications. Because only *SVIM* distinguishes between insertions and interspersed duplications, we compared the tools on four common basic SV classes in the simulated datasets: deletions, insertions, inversions and tandem duplications. To obtain insertion calls from *SVIM* for the following benchmarks, we merged the calls of interspersed duplications with regular insertion calls.

The simulated datasets made it possible to evaluate the SV calling performance of the three methods in different settings. They allowed us to investigate strengths and weaknesses of the tools and enabled us to quantify:

- how well the tools detect SVs of four different classes: deletions, inversions, insertions and tandem duplications,
- how well the tools perform with different levels of sequencing coverage and
- how well the tools detect heterozygous and homozygous SVs.

In Figure 5.1, our results for the detection of homozygous SVs from the 15x coverage simulated dataset are shown. The figure shows four precision-recall plots for the four different SV classes. In each plot, the precision on the y-axis is plotted against the recall on the x-axis. The three colored curves represent the three evaluated tools. Every point on a curve gives the precision and recall for a certain value of the confidence threshold. The points with the best threshold values are located in the upper right corner of the plot where both precision and recall are maximized.

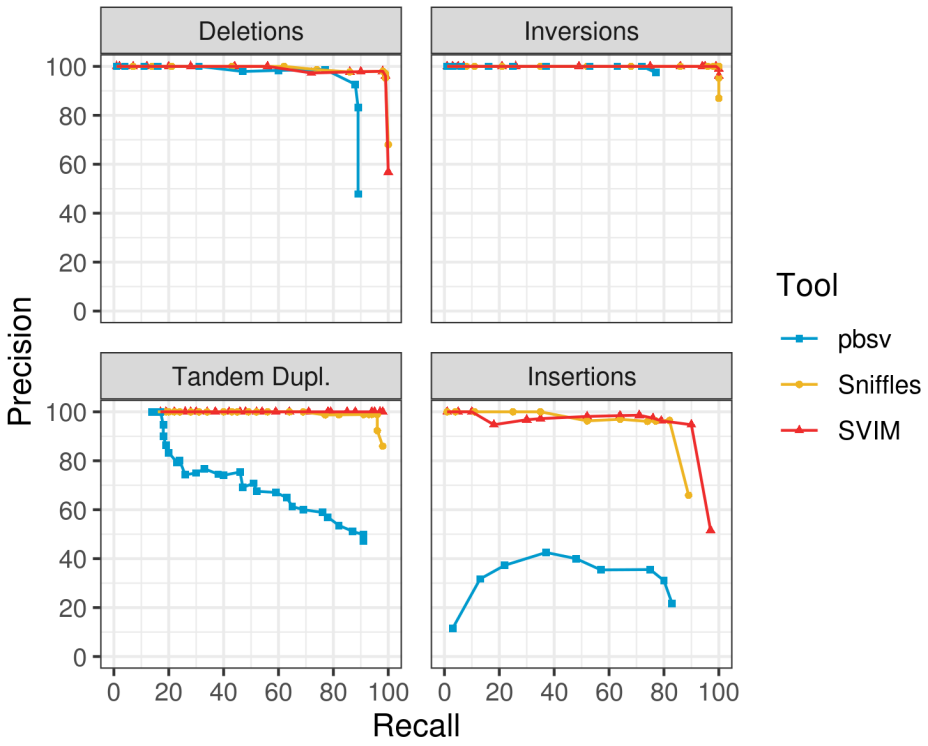


Figure 5.1: **Comparison of SV detection performance on a 15x coverage homozygous simulated dataset.** Shown are recall (x-axis) and precision (y-axis) reached by each tool for different confidence thresholds. Recall and precision were calculated requiring a maximal distance of 1 kbp and a span difference of less than 0.3 between matching variant calls and the original simulated variants. Reads were aligned using *minimap2* (*SVIM* and *Sniffles*) and *pbmm2* (*pbsv*).

The results show that *Sniffles* and *SVIM* performed substantially better than *pbsv* across all SV classes. *pbsv* struggled in particular on tandem duplications and insertions where it reached a far lower precision than the other two tools. Although *SVIM* and *Sniffles* exhibited a similar performance, *SVIM* outperformed *Sniffles* on tandem duplications and insertions where it maintained high precision even for confidence thresholds that reached a very high recall.

Instead of plotting the results for all confidence thresholds in a precision-recall plot, we can alternatively pick only the "best" threshold that yields the highest F1 score. Conceptually, this reduces each curve in Figure 5.1 to one data point and makes it easier to compare the callers in different settings. In Figure 5.2, the best F1 score reached by a caller is plotted on the y-axis against five levels of read coverage on the x-axis. The colors of the bars represent the

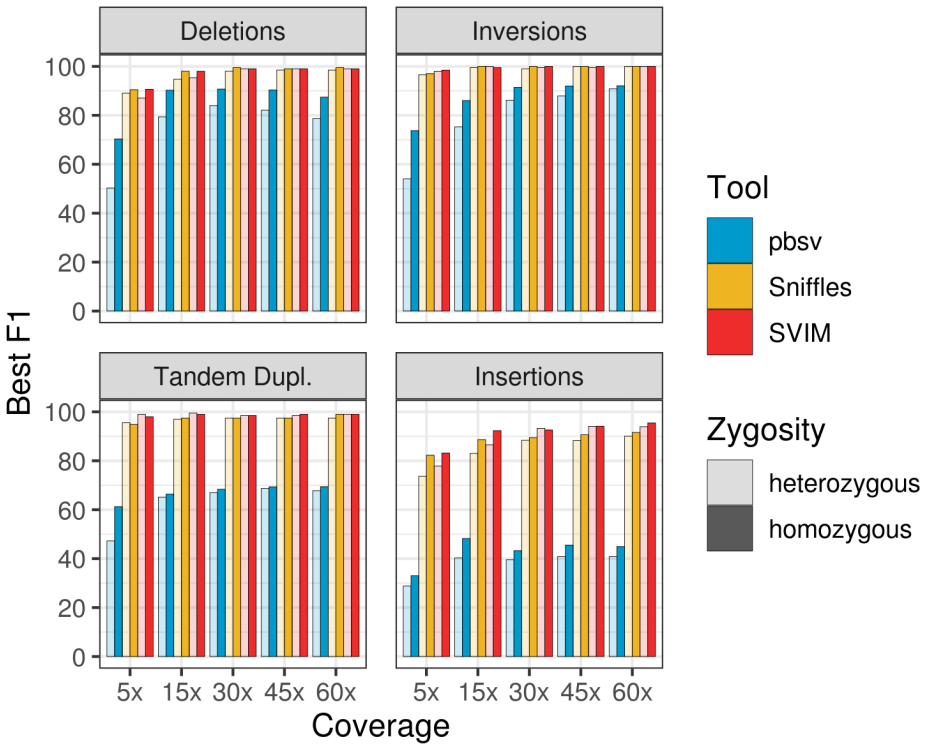


Figure 5.2: **Best SV detection performance for five different simulated coverage levels.** Shown are the best F1 scores (y-axis) reached by each tool for different read coverages between 5x and 60x (x-axis). Generally, higher coverages enabled higher F1 scores. F1 scores were calculated requiring a maximal distance of 1 kbp and a span difference of less than 0.3 between matching variant calls and the original simulated variants. Reads were aligned using *minimap2* (*SVIM* and *Sniffles*) and *pbmm2* (*pbsv*).

three tools while the color saturation represents the zygosity of the simulated SVs.

The first observation one can make is that the SV detection performance increased with higher coverages. However, as the coverage reached 30x, none of the callers substantially benefited from even more data. When comparing the three tools, the observations from the 15x dataset above were confirmed. *SVIM* achieved the best results among the three tools and was followed closely by *Sniffles*. While both were largely on a par for deletions and inversions, *SVIM* reached better F1 scores for tandem duplications and, in particular, insertions.

For the detection of homozygous deletions, inversions and tandem duplications, *SVIM* and *Sniffles* reached almost perfect results (F1 scores $> 97\%$) even for coverages as low as 15x. Insertions proved

slightly harder to detect with F1 scores of 92.3%, 88.6% and 48.2%, respectively, for *SVIM*, *Sniffles* and *pbsv* on the 15x dataset.

When comparing different zygosities, heterozygous SVs were harder to detect than homozygous SVs and all tools reached slightly lower F1 scores on heterozygous SVs. This was expected, as heterozygous SVs are supported by only a subset of reads, making them harder to distinguish from the general signal noise. For heterozygous SVs, *SVIM* again produced the best results followed closely by *Sniffles*.

To measure the influence of the input read alignments on SV calling, we also compared the previous results from alignments by *minimap2* with results from another long-read aligner, *NGMLR* (see Figure B.1 in the appendix). The results indicate that *SVIM* is very robust to the choice of the aligner. *Sniffles*, however, reached considerably worse results for calling insertions when analyzing alignments by *NGMLR*. Visual inspection of the alignments revealed a difference in the way that reads covering insertions are aligned. While *minimap2* expresses insertions mainly as long reference gaps in the CIGAR string (intra-alignment discordancies), *NGMLR* tends to split reads at insertions (inter-alignment discordancies). Because *Sniffles* does not call insertions of sequence existing somewhere else in the genome (i.e. interspersed duplications) from split alignments, it reached a lower recall with *NGMLR* on our simulated datasets.

5.1.3 *Real datasets*

Simulations cannot reproduce all aspects of real biological data. Even sequencing data from a sophisticated read simulator like *RSVSim* is different from actual long-read sequencing data. Therefore, we used real PacBio and Nanopore data in the second part of our analysis. There, we analyzed three different long-read datasets from the same individual, *HG002*. This individual is part of the Ashkenazi Jewish trio in the Personal Genomes Project (PGP) and their DNA is available as a Reference Material of the National Institute of Standards and Technology (NIST) [101].

A large variety of sequencing datasets are available for HG002 as well as a draft SV benchmark set generated by the NIST-hosted

	A	B	C
Data type	PacBio CLR	PacBio CCS	Nanopore
Seq. instrument	Sequel II	Sequel II	PromethION
Read coverage	38.7x	36.6x	50.7x
Mean read len.	14.9 kbp	12.9 kbp	8.2 kbp
Read len. N50	20.9 kbp	12.9 kbp	49.4 kbp
Source	PacBio [6]	PacBio [7]	UCSC [79, 83]

Table 5.1: **Three recently generated long-read datasets for the HG002 individual.** The datasets were compiled by the Human Pangenome Reference Consortium (see `https://github.com/human-pangenomics/HG002_Data_Freeze_v1.0` for details). *Abbreviations: Seq. - Sequencing, len. - length*

Genome in a Bottle Consortium (GIAB) [102]. To generate the benchmark set, GIAB combined 19 variant calling methods applied to four technologies: short-read (Illumina and Complete Genomics), linked-read (10x Genomics) and long-read (PacBio) sequencing as well as optical (Bionano) and electronic (Nabsys) mapping. With 12,745 isolated, sequence-resolved insertion and deletion calls larger than 50 bp it represents the most comprehensive callset of germline SVs to date. We used the benchmark set in our analysis of real long-read datasets to identify false negative and false positive SV calls and to compare the precision and recall of different SV callers. It is worth noting that *pbsv* was among the calling methods used to generate the benchmark set which could give the tool a slight advantage in the comparisons below.

For our analysis, we chose long-read datasets produced using three different sequencing technologies: PacBio CLR, PacBio CCS and Oxford Nanopore sequencing (see Table 5.1). The evaluation of the SV callers on these three most common TGS technologies gives a realistic insight into their practical performance. All datasets were generated with the most recent sequencing instrument available for each platform (PacBio Sequel II and Oxford Nanopore PromethION). They all had sequencing coverages of more than 30x, with the Oxford Nanopore dataset reaching the highest coverage of 50.7x. Of the two PacBio datasets, the CLR dataset contained longer reads with a mean read length of 14.9 kbp compared to 12.9 kbp for the CCS dataset. While the Nanopore dataset showed the lowest mean read length

(8.2 kbp) it reached by far the highest read length *N50*. The N50 measures the length of the longest read in the dataset for which it is true that 50% of all bases in the dataset are from reads longer than that read. In the Nanopore dataset, 50% of the bases were contained in reads with a length of 49.4 kbp or larger. For the two PacBio datasets, read length N50 was substantially lower with 20.9 kbp and 12.9 kbp, respectively. The high read length N50 in the Nanopore dataset was achieved with a special *ultra-long read* protocol. Due to this library preparation protocol, a large fraction of the reads in the dataset were longer than 100 kbp, amounting to a coverage of 8.5x with these reads alone.

To assess the influence of the sequencing coverage on the SV detection performance, we subsampled all three datasets to 10 different coverage levels using *samtools view* and performed SV calling on each subset.

5.1.3.1 *Dataset A - PacBio CLR*

Dataset A is a PacBio CLR dataset with 38.7x read coverage and a read length N50 of 20.9 kbp. PacBio CLR sequencing is characterized by a relatively high error rate between 8 and 13% [65]. The majority of errors are small indels with substantially more insertions than deletions [95].

We generated precision-recall curves which are shown in Figure 5.3. Because all three tools are able to detect SVs and estimate their genotype, we performed two evaluations represented by the two panels: Firstly, we evaluated the ability of the tools to detect the presence of an SV regardless of its predicted genotype (Figure 5.3a). In stark contrast to the results on the simulated data, *pbsv* outperformed both *Sniffles* and *SVIM* reaching a better balance of precision and recall than the two others. *pbsv*'s increased performance on real data can at least partly be explained by differences in the error profile compared to simulated data. Most likely, *pbsv* has been optimized for real PacBio data and its performance deteriorates on datasets with slightly different characteristics. This finding emphasizes the importance of evaluating computational methods on both simulated and real data. Behind *pbsv*, *Sniffles* and *SVIM* were on a par although

SVIM reached both a higher maximal recall and a higher maximal precision.

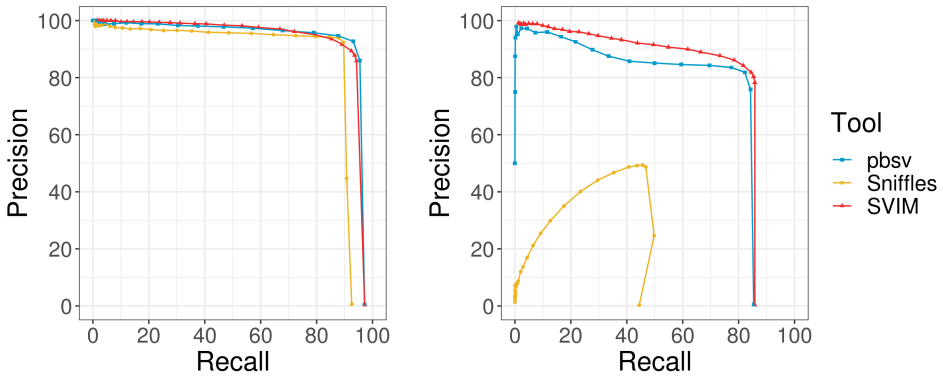
Secondly, we assessed the ability of the tools to detect the presence and exact genotype of an SV (Figure 5.3b). In this scenario, the precision and recall of all tools decreased substantially. *SVIM* yielded the best results followed by *pbsv* indicating that *SVIM* produced more reliable genotypes than *pbsv*. Compared to the previous evaluation, *Sniffles*' performance decreased the most with both precision and recall falling below 50%. This indicates that *Sniffles* had great difficulties predicting correct genotypes.

To investigate how the tools cope with lower levels of PacBio CLR data, we subsampled the dataset into ten different subsets containing between 10% and 90% of the total number of reads. In Figure 5.4, we plot the best F1 score (y-axis) reached by each tool against the subsampling fraction (x-axis). As for the simulated data, we generally observed increasing F1 scores from higher sequencing depth but also a saturation of performance for higher coverages. When we assessed variant calls and ignored genotypes (Figure 5.4a), *pbsv* outperformed the other methods except for the lowest coverage level. *SVIM* and *Sniffles* reached similar results except for the 30-50% subsamples where *SVIM* was slightly superior. When genotypes were evaluated, F1 scores dropped for all tools while *SVIM* overtook *pbsv* on high coverages (Figure 5.4b). At the same time, *Sniffles* produced considerably worse results than the others with F1 scores remaining below 50%.

5.1.3.2 *Dataset B - PacBio CCS*

Dataset B is a PacBio CCS dataset with 36.6x read coverage and a read length N50 of 12.9 kbp. Because PacBio CCS sequencing generates a consensus sequence from multiple sequencing rounds of the same circular fragment, it is characterized by a relatively low error rate of less than 1% and shorter read lengths than PacBio CLR [97]. The majority of remaining errors are indels in homopolymer regions.

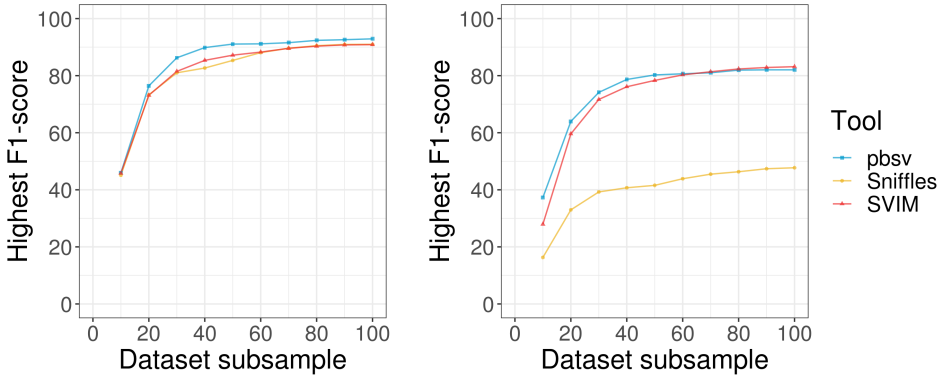
For this dataset, we performed the same analyses as for dataset A. Figure 5.5 shows the precision-recall curves for calls and genotyped calls, respectively. In the first setting, the differences between the tools were smaller than from PacBio CLR data, presumably because



(a) **Calls only**: Ability of the tools to detect the presence of SVs regardless of their genotype

(b) **Calls with genotype**: Ability of the tools to detect the presence of SVs and their correct genotype

Figure 5.3: **Precision-recall curves for three SV callers on the 38.7x PacBio CLR dataset.** Shown are recall (x-axis) and precision (y-axis) reached by each tool for different confidence thresholds. Precision and recall were calculated requiring a distance of less than 1 kbp and a span difference of less than 0.3 between matching variant calls and the gold standard variants. Reads were aligned using *minimap2* (*SVIM* and *Sniffles*) and *pbmm2* (*pbsv*).



(a) **Calls only**: Ability of the tools to detect the presence of SVs regardless of their genotype

(b) **Calls with genotype**: Ability of the tools to detect the presence of SVs and their correct genotype

Figure 5.4: **Best SV detection performance for ten different subsamples of the 38.7x PacBio CLR dataset.** Shown are the best F1 scores (y-axis) reached by each tool for different subsamples between 10% and 100% of the full 38.7x coverage (x-axis). F1 scores were calculated requiring a distance of less than 1 kbp and a span difference of less than 0.3 between matching variant calls and the gold standard variants. Reads were aligned using *minimap2* (*SVIM* and *Sniffles*) and *pbmm2* (*pbsv*).

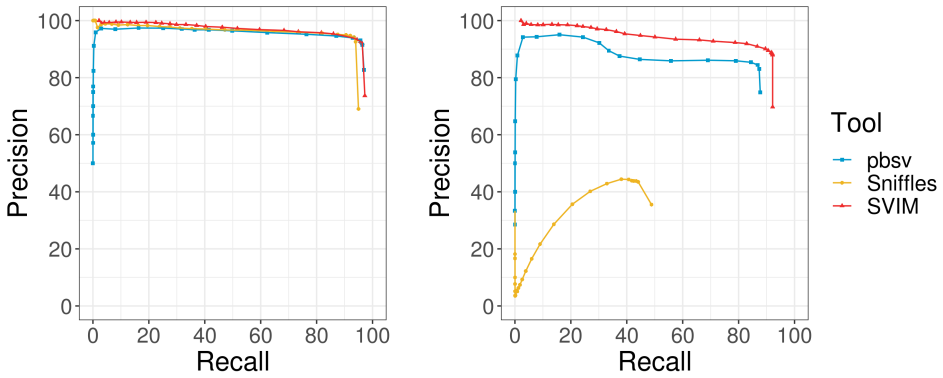
of the lower amount of noise in the PacBio CCS data (Figure 5.5a). All three tools reached similar precision and recall for their optimal confidence threshold. However, *SVIM* and *pbsv* reached a higher maximal recall than *Sniffles* while *SVIM* and *Sniffles* reached a higher maximal precision than *pbsv*. When taking genotypes into account, *SVIM* outperformed the other tools by reaching a precision and recall of approximately 90% compared to 85% and 44% for *pbsv* and *Sniffles*, respectively (Figure 5.5b).

We again assessed the best F1 score reached by each tool on different subsamples of the dataset and observed a swift saturation of performance even for relatively low coverages (Figure 5.6). When ignoring genotypes, differences between the tools were more pronounced for lower coverages (Figure 5.6a). The performance of *Sniffles* degraded faster than that of the other tools for decreasing read coverages while *pbsv* gained a slight advantage over *SVIM* for coverages below 15x (50% subsample). When evaluating genotyped calls, *SVIM* outperformed the two others with the exception of the two lowest coverage levels below 10x (30% subsample) where *pbsv* was able to predict more reliable genotypes (Figure 5.6b). Across all coverage levels, *Sniffles* produced poor results when genotypes were considered with F1 scores below 50%.

5.1.3.3 *Dataset C - Nanopore*

Dataset C is an Oxford Nanopore dataset with 50.7x read coverage and a read length N50 of 49.4 kbp. Oxford Nanopore sequencing is characterized by a relatively high error rate, between 10 and 20% [95]. The majority of errors are small indels that occur particularly often in homopolymers. In contrast to PacBio sequencing, there are substantially more deletions than insertions.

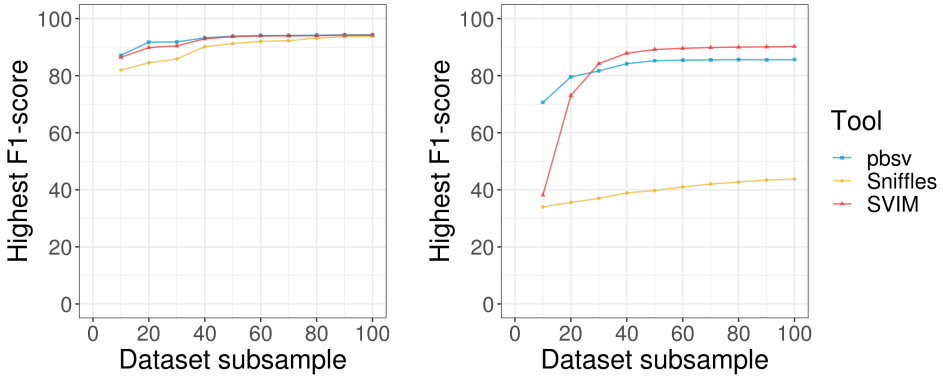
As *pbsv* is a development of PacBio and has been targeted and optimized for their own PacBio SMRT platform, we restricted our analysis to *Sniffles* and *SVIM* on this Nanopore dataset. The precision-recall curves for calls and genotyped calls, respectively, are shown in Figure 5.7. In both settings, *SVIM* consistently reached a higher precision and recall than *Sniffles*. Similar to the other datasets, *Sniffles* failed to predict accurate genotypes and was outperformed by



(a) **Calls only**: Ability of the tools to detect the presence of SVs regardless of their genotype

(b) **Calls with genotype**: Ability of the tools to detect the presence of SVs and their correct genotype

Figure 5.5: **Precision-recall curves for three SV callers on the 36.6x PacBio CCS dataset.** Shown are recall (x-axis) and precision (y-axis) reached by each tool for different confidence thresholds. Precision and recall were calculated requiring a distance of less than 1 kbp and a span difference of less than 0.3 between matching variant calls and the gold standard variants. Reads were aligned using *minimap2* (*SVIM* and *Sniffles*) and *pbmm2* (*pbsv*).



(a) **Calls only**: Ability of the tools to detect the presence of SVs regardless of their genotype

(b) **Calls with genotype**: Ability of the tools to detect the presence of SVs and their correct genotype

Figure 5.6: **Best SV detection performance for ten different subsamples of the 36.6x PacBio CCS dataset.** Shown are the best F1 scores (y-axis) reached by each tool for different subsamples between 10% and 100% of the full 36.6x coverage (x-axis). F1 scores were calculated requiring a distance of less than 1 kbp and a span difference of less than 0.3 between matching variant calls and the gold standard variants. Reads were aligned using *minimap2* (*SVIM* and *Sniffles*) and *pbmm2* (*pbsv*).

a particularly wide margin when genotypes were considered (Figure 5.7b).

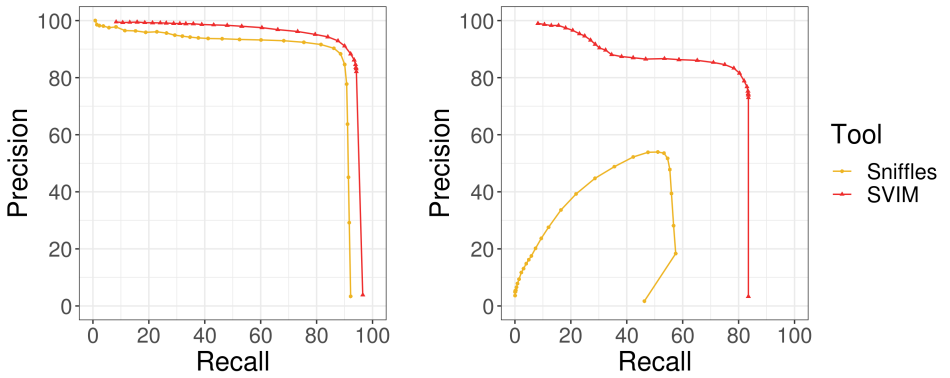
These trends were persistent across all coverage levels as is shown in Figure 5.8. *SVIM* yielded F1 scores that were slightly higher for simple calls and considerably higher for genotyped calls.

5.1.3.4 *Comparison of different sequencing technologies*

Beyond a comparison of the different SV callers, the three sequencing datasets enabled us to assess the different TGS technologies and their suitability for SV calling and genotyping. Above, we already analyzed the SV calling performance with varying sequencing coverages and observed that the performance generally improves with increasing coverage. Now, we compare different sequencing technologies in terms of the SV detection performance they enable.

In Figure 5.9, we plot the alignment coverage on the x-axis against the best F1 score achieved by *SVIM* on the y-axis (for similar plots using calls from *Sniffles* and *pbsv* see Figure B.2 and B.3 in the Appendix). The line color represents the sequencing technologies PacBio CLR, PacBio CCS and Oxford Nanopore. The left panel of the figure shows the results for simple calls ignoring their genotype. Again, we observed that the F1 score increased with increasing coverage. However, the different technologies did not follow the same curve. The two less accurate technologies, PacBio CLR and Oxford Nanopore, achieved substantially lower F1 scores than the more accurate PacBio CCS sequencing for the same alignment coverage. Understandably, an elevated error rate on the sequence level seems to result in a higher amount of erroneous SV calls while more accurate read sequences make it easier for the SV callers to distinguish the signal from the noise. Although PacBio CLR and Oxford Nanopore reached a similar level of performance, PacBio CLR performed slightly better than Oxford Nanopore for all but the lowest coverages.

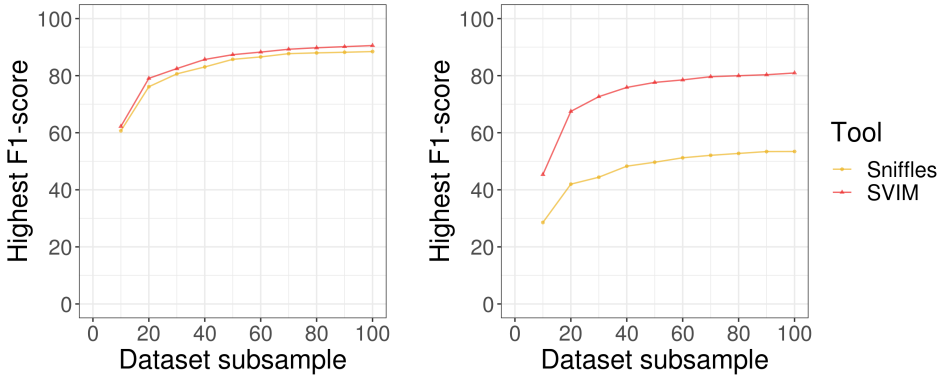
From the PacBio CCS data, a high SV calling performance could be reached even using a relatively low coverage. A coverage of only 10x was sufficient to produce F1 scores above 90%. PacBio CLR and Oxford Nanopore, in contrast, required a coverage of 24x and 36x, respectively, to reach the same level of performance. At 28x coverage, PacBio CCS enabled F1 scores above 94% which was far beyond



(a) **Calls only**: Ability of the tools to detect the presence of SVs regardless of their genotype

(b) **Calls with genotype**: Ability of the tools to detect the presence of SVs and their correct genotype

Figure 5.7: **Precision-recall curves for three SV callers on the 50.7x Oxford Nanopore dataset.** Shown are recall (x-axis) and precision (y-axis) reached by each tool for different confidence thresholds. Precision and recall were calculated requiring a distance of less than 1 kbp and a span difference of less than 0.3 between matching variant calls and the gold standard variants. Reads were aligned using *minimap2* for both *SVIM* and *Sniffles*.



(a) **Calls only**: Ability of the tools to detect the presence of SVs regardless of their genotype

(b) **Calls with genotype**: Ability of the tools to detect the presence of SVs and their correct genotype

Figure 5.8: **Best SV detection performance for ten different subsamples of the 50.7x Oxford Nanopore dataset.** Shown are the best F1 scores (y-axis) reached by each tool for different subsamples between 10% and 100% of the full 50.7x coverage (x-axis). F1 scores were calculated requiring a distance of less than 1 kbp and a span difference of less than 0.3 between matching variant calls and the gold standard variants. Reads were aligned using *minimap2* for both *SVIM* and *Sniffles*.

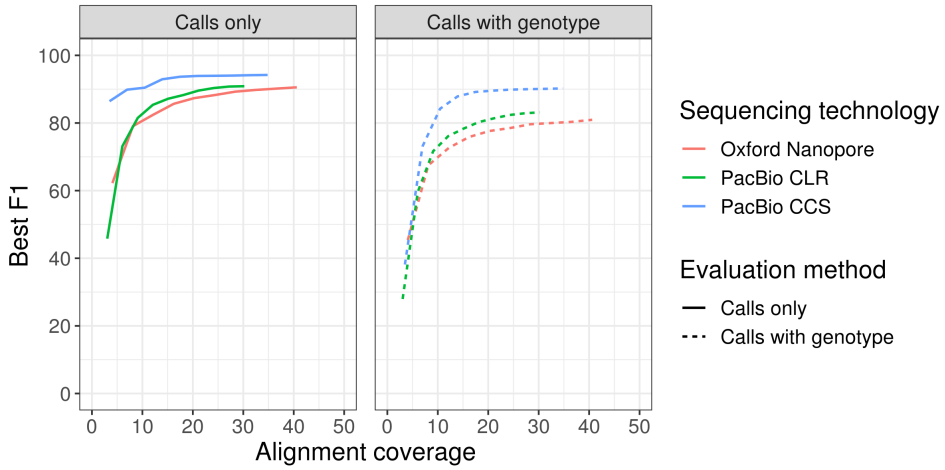


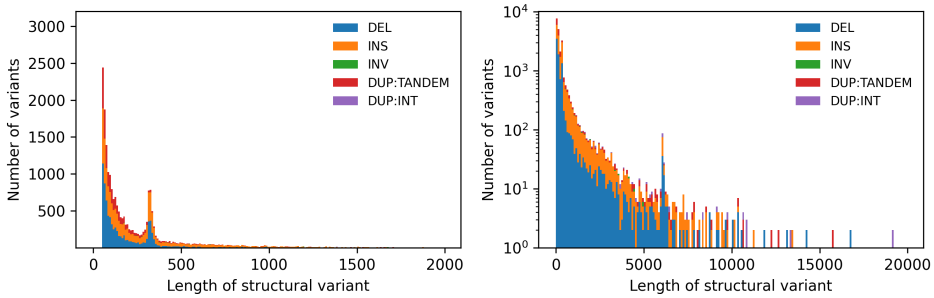
Figure 5.9: **Best SV detection performance reached by *SVIM* on sequencing datasets of different coverage levels from different technologies.** Plotted are the best F1 scores achieved by *SVIM* (y-axis) against the alignment coverage (x-axis) of the sequencing dataset (represented by the line color). Results for simple and genotyped calls are visualized in the left and right panel by solid and dashed lines, respectively. F1 scores were calculated requiring a distance of less than 1 kbp and a span difference of less than 0.3 between matching variant calls and the original simulated variants. Reads were aligned using *minimap2*.

the maximum performance reached by PacBio CLR and Oxford Nanopore (maximum F1 scores of 90.9% and 90.6%, respectively). When considering genotypes (right panel of Figure 5.9), all previous observations were confirmed although the general SV detection performance dropped slightly. PacBio CCS still produced the best results with a maximum F1 score of 90.2% compared to 83.2% and 80.9% for PacBio CLR and Oxford Nanopore, respectively.

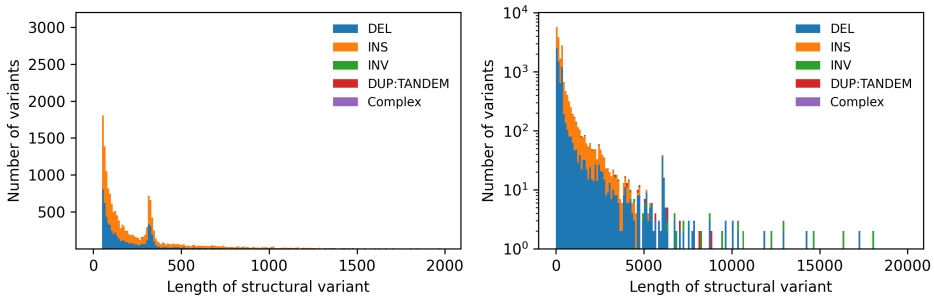
5.1.3.5 *Analysis of SV lengths and classes*

After evaluating the performance of different SV callers and sequencing technologies, we assessed the sizes and classes of SVs detected in the HG002 individual. For this analysis, we used the PacBio CCS dataset (dataset B) because it had achieved the best SV detection performance in the benchmarks above (for similar results for the PacBio CLR and Oxford Nanopore datasets see Figure B.4 and B.5 in the Appendix).

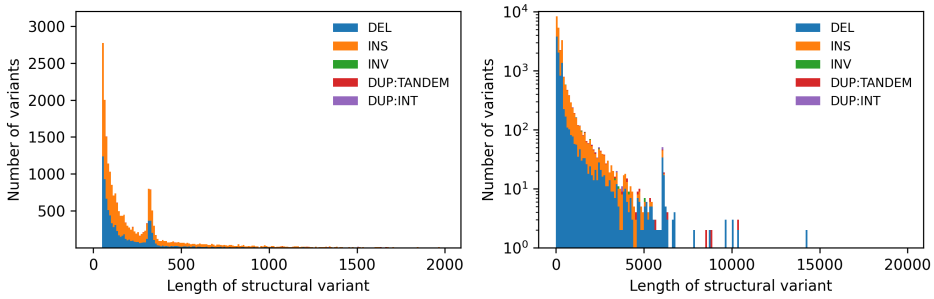
In Figure 5.10, the frequency of SVs from different classes is plotted against their length. We observed a characteristic size distribution



(a) **pbsv**



(b) **Sniffles**



(c) **SVIM**

Figure 5.10: **Size distribution of SVs detected in the 36.6x PacBio CCS dataset.** Shown are stacked histograms of SV classes represented by different colors. In the left column, SV sizes up to 2 kbp are plotted with a bin size of 10 bp. In the right column, SV sizes up to 20 kbp are plotted on a logarithmic y-axis with a bin size of 100 bp. The top, middle and bottom panels, visualize callsets by *pbsv*, *Sniffles* and *SVIM*, respectively. All callsets were generated with a confidence threshold of 5. To simplify the comparison, SV class names have been harmonized between the tools.

that has already been described in several earlier studies [20, 43, 88]. Regardless of the SV calling method, the number of detected SVs decreased exponentially with increasing size and the vast majority of SVs were small. Two prominent peaks could be observed in all callsets: one at approximately 300 bp corresponding to *Alu* elements and another one at approximately 6 kbp corresponding to *LINE1* elements. Alu and LINE1 elements are both mobile elements that spread across the genome through interspersed duplication. In HG002, they were responsible for a large number of deletions and insertions depending on whether the duplication occurred in the reference genome or the genome of HG002.

In total, *pbsv*, *Sniffles* and *SVIM* detected 23,389, 18,499 and 24,111 SVs, respectively (see Figure 5.11, panel "all"). Approximately half of these SVs (12,808, 9,686 and 13,825, respectively) were smaller than 200 bp (panel "tiny"). Another third (8,163, 6,848 and 8,343, respectively) had a size between 200 bp and 1 kbp (panel "small"). Only few SVs fell into the larger size categories (panels "medium", "large" and "huge").

All detected SVs reached a cumulative length of 16.6 (*pbsv*), 642.6 (*Sniffles*) and 14.5 Mbp (*SVIM*), respectively. The considerably larger value for *Sniffles* was mainly caused by a few very large but most likely spurious SV calls. While all SVs greater than 100 kbp detected by *pbsv* and *SVIM* amounted to 2.0 and 3.6 Mbp, respectively, the 7 inversions, 18 deletions and 4 tandem duplications in this size range detected by *Sniffles* had an unrealistic total size of 630.5 Mbp.

Generally, the size and class distributions from the three methods were very similar although slight differences could be observed. All three classified the majority of SVs as deletions and insertions. The most prominent difference was the higher number of tandem duplications detected by *pbsv* (3,942 compared to 64 and 128 for *Sniffles* and *SVIM*, respectively). Unlike the other two approaches, *pbsv* compares the sequences of insertions to the neighboring reference regions. If the sequences match, the insertion is called as a tandem duplication instead. All duplications, by definition, can also be represented as insertions. This ambiguity is one of the difficulties of SV calling making it hard to compare callsets from different sequencing technologies or algorithms. Among the other detected SVs, 45 (*pbsv*), 107 (*Sniffles*) and 21 (*SVIM*) were inversions. Furthermore,

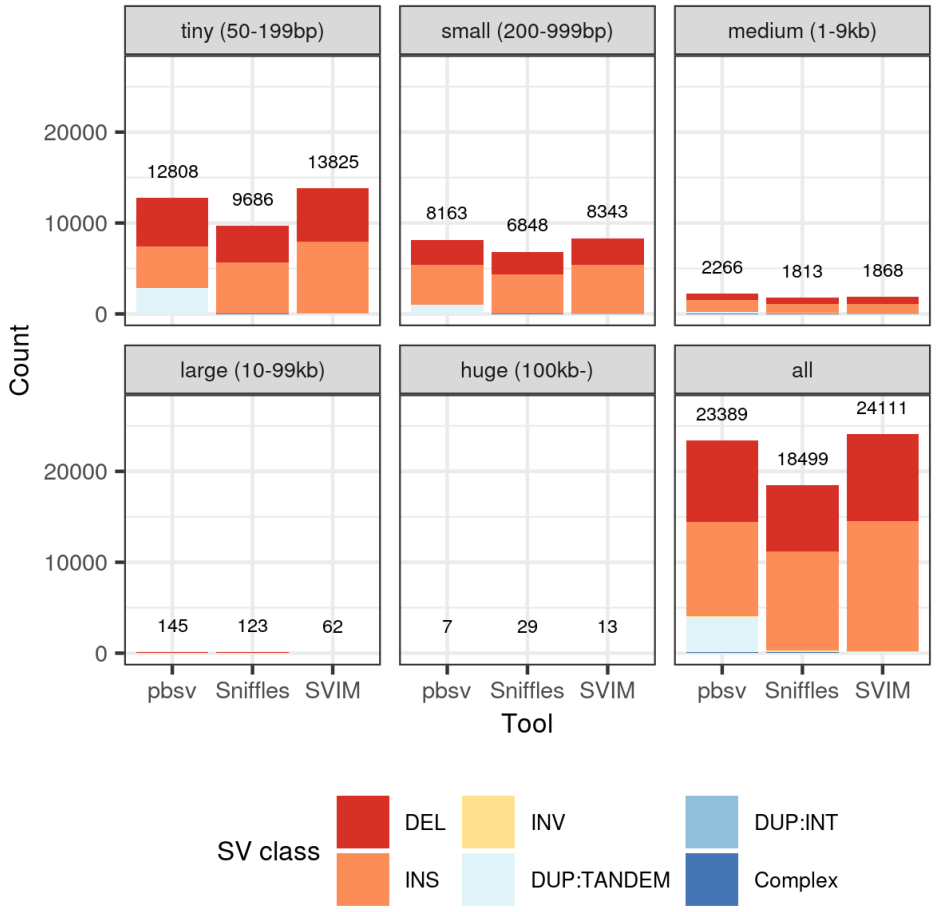


Figure 5.11: **Number of SV calls from the 36.6x PacBio CCS dataset stratified into five size classes.** Shown is a stacked bar plot of SV classes represented by different colors. Each panel represents one size class and visualizes the number of calls in that size range called by *pbsv*, *Sniffles* and *SVIM*, respectively. The bottom right panel shows the counts for all SV calls regardless of size. All callsets were generated with a confidence threshold of 5.

SVIM detected 24 interspersed duplications while *Sniffles* called 101 complex SVs.

5.1.3.6 *Comparison of runtime and memory consumption*

To compare the runtime and memory consumption of *pbsv*, *Sniffles* and *SVIM*, we ran the tools on the same PacBio CCS dataset (Dataset B with 36.6x coverage). *Sniffles* and *SVIM* were given input alignments produced by *minimap2* while *pbsv* was given *pbmm2* alignments. Three measurements were taken with GNU time (v1.9) on a machine with an AMD EPYC 7601 CPU (128 cores, 2.7 GHz) and

1 TB of memory. Only the runtime of the SV detection was measured not considering the time required for producing the alignments.

All three tools analyzed the entire dataset in under 3 hours (see Table 5.2). *Sniffles* was the fastest tool taking only 51 minutes. *pbsv* and *SVIM* took considerably longer with 109 and 129 minutes, respectively. When comparing memory consumption, *SVIM* turned out to require the least memory with only 1,248 MB. *Sniffles* and *pbsv* consumed substantially more with 2,075 MB and 6,133 MB, respectively.

We also measured the runtimes of the individual components in the *SVIM* pipeline (see Table 5.3). The two components requiring the most time were the COLLECT (1,654s) and GENOTYPE (5,922s) stages. Together, they consumed more than 98% of the total runtime. Both of them read and analyze alignments from the input BAM file. This is an IO-intensive process limited by the speed of the hard disk. While the COLLECT component sequentially reads records from the BAM file, the GENOTYPE component needs to fetch reads from the genomic neighborhood of each SV candidate. This targeted reading is slower because it requires the file reader to jump around in the BAM file. Without the genotyping, *SVIM* analyzes the dataset in less than 30 min or only 23.2% of the runtime with genotyping.

5.2 EVALUATION ON GENOME ASSEMBLIES

We compared our tool, *SVIM-asm* (v0.1.1), to the *DipCall* pipeline (v0.1) [58]. Both tools are designed for reference-based variant calling on diploid genome assemblies.

For the evaluation we chose two publicly available diploid genome assemblies of the HG002 individual from Wenger et al. (Assembly A) and Garg et al. (Assembly B) (see Section C.3 in the appendix) [30, 97]. Assembly A was generated using the assembler *Canu* on a trio-binned PacBio CCS dataset (29.7x coverage). Assembly B, in contrast, combined the same PacBio CCS dataset with Hi-C data (28.5x coverage) for scaffolding using the assembly pipeline *DipAsm*. For both assemblies, we aligned the genome fragments separately for each haplotype using *minimap2* (v2.17-r941) and produced genotyped SV calls using *SVIM-asm* and *DipCall*, respectively [56]. For

Table 5.2: **Runtime and memory consumption on the 36.6x PacBio CCS dataset.** The runtime of each tool for SV calling was measured, i.e. excluding the prior read alignment step. *CPU time* denotes the sum of *User time* and *System time* while *Wall clock time* measures the real time passed. The *Maximum memory* represents the maximum resident set size. Three measurements were taken for each value using GNU time (v1.9) and the average of the three measurements is reported.

Tool	Threads	CPU time (min)	Wall clock time (min)	Maximum memory (MB)
pbsv	1	109	109	6133
Sniffles	1	51	51	2075
SVIM	1	129	129	1248

Table 5.3: **Runtime of the *SVIM* components on the 36.6x PacBio CCS dataset.** The runtime of each component was measured based on the time stamps printed in the log file. Three measurements were taken from separate runs on the same dataset and the average of the three measurements is reported.

Component	Wall clock time (sec)	% of total
COLLECT	1654	21.4%
CLUSTER	93	1.2%
COMBINE	27	0.3%
GENOTYPE	5922	76.8%
Final Output & Plotting	20	0.3%

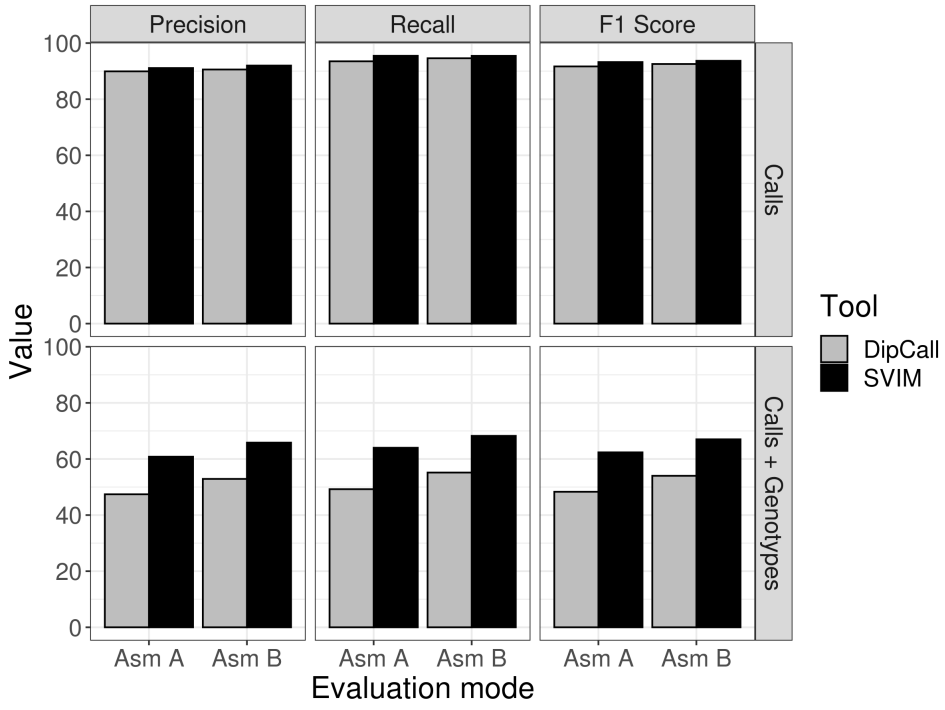
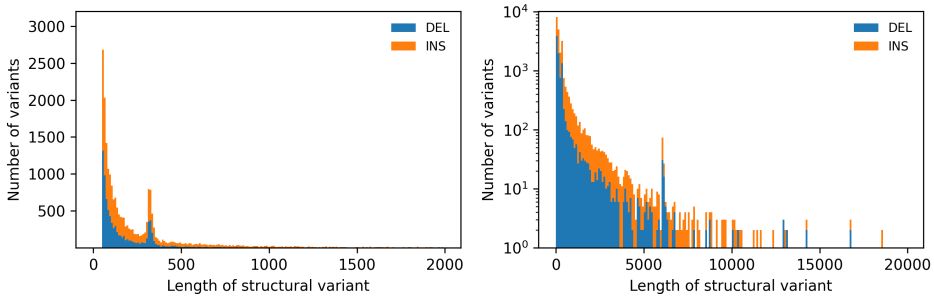


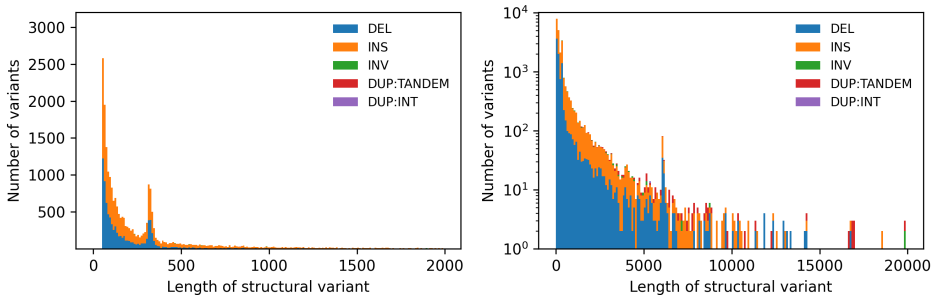
Figure 5.12: **Comparison of SV detection performance of *DipCall* and *SVIM-asm* on two diploid genome assemblies.** Plotted are the precision, recall and F1 score (y-axis) measured by comparing against the GIAB SV benchmark set. Results are shown for two genome assemblies (x axis): Asm A by Wenger et al. and Asm B by Garg et al. Two evaluation modes are distinguished: 1) Evaluating the ability of the tools to detect the presence of SVs regardless of their genotype (Calls) and 2) evaluating the ability of the tools to detect the presence of SVs and their correct genotype (Calls + Genotypes).

HG002, a comprehensive callset of germline SVs is available from the GIAB consortium (see Section 5.1.3) which we used to compute the precision, recall and F1 score of the two SV callers (using the same definitions and methods as in our previous analyses, see Section 5.1.1).

We observed that both methods reached F1 scores above 90% when only the variant calls were evaluated (see Figure 5.12, upper panels). *SVIM-asm* performed slightly better than *DipCall* with F1 scores of 93.2% (Assembly A) and 93.7% (Assembly B) compared to 91.7% and 92.5%, respectively. When additionally requiring matching genotypes, the values for both tools decreased considerably and *SVIM-asm* outperformed *DipCall* by a wide margin (see Figure 5.12, lower panels). While *SVIM-asm* reached F1 scores of 62.4% and



(a) **DipCall**



(b) **SVIM-asm**

Figure 5.13: **Size distribution of SVs identified in Assembly A.** Shown is a stacked histogram of SV classes represented by different colors. In the left panels, SV sizes up to 2 kbp are plotted with a bin size of 10 bp. In the right panel, SV sizes up to 20 kbp are plotted on a logarithmic y-axis with a bin size of 100 bp. The top and bottom panels, visualize callsets by *DipCall* and *SVIM-asm*, respectively.

67.0%, *DipCall* only reached 48.3% and 54.0%, respectively. When measuring precision and recall across variant lengths, we observed that *SVIM-asm* reached a higher recall particularly for large deletions and insertions (see Figures B.8 and B.9 in the Appendix).

When we analyzed the size distribution of SVs detected from Assembly A (see Figure 5.13), we observed a close resemblance to the distributions retrieved from long-read alignments (see Figure 5.10). Again, we found an exponential decrease in the number of SVs with increasing size and two characteristic peaks at 300 bp and 6 kbp. The distributions from *DipCall* (Figure 5.13a) and *SVIM-asm* (Figure 5.13b) were highly similar indicating that both tools produce similar callsets. The same observations could be made also on Assembly B (see Figure B.10 in the Appendix).

In total, *DipCall* and *SVIM-asm* detected 23,321 and 24,170 SVs (excluding translocations), respectively, from Assembly A (see Fig-

ure 5.14, panel "all"). This is again similar to the number of calls from long-read alignments. Approximately half of the SVs (13,214 and 13,023, respectively) were smaller than 200 bp (panel "tiny"). Another third (7,878 and 8,290, respectively) had a size between 200 bp and 1 kbp (panel "small"). Only few SVs fell into the larger size categories (panels "medium", "large" and "huge"). As for long-read alignments, the majority of SVs were categorized as deletions and insertions.

Unlike *DipCall*, *SVIM-asm* is able to detect inversions, translocation breakpoints and two types of duplications. From the Assemblies A and B it detected 90 / 73 inversions, 124 / 101 tandem duplications, 2 / 4 interspersed duplications and 3110 / 4028 translocation breakpoints, respectively.

All in all, our evaluations demonstrated that *SVIM* and *SVIM-asm* produce accurate results on diverse input datasets and outperform existing methods on the detection of genotyped SVs. In the next chapter, we will apply *SVIM* to investigate the structural variants and novel adjacencies in a set of highly rearranged patient genomes.

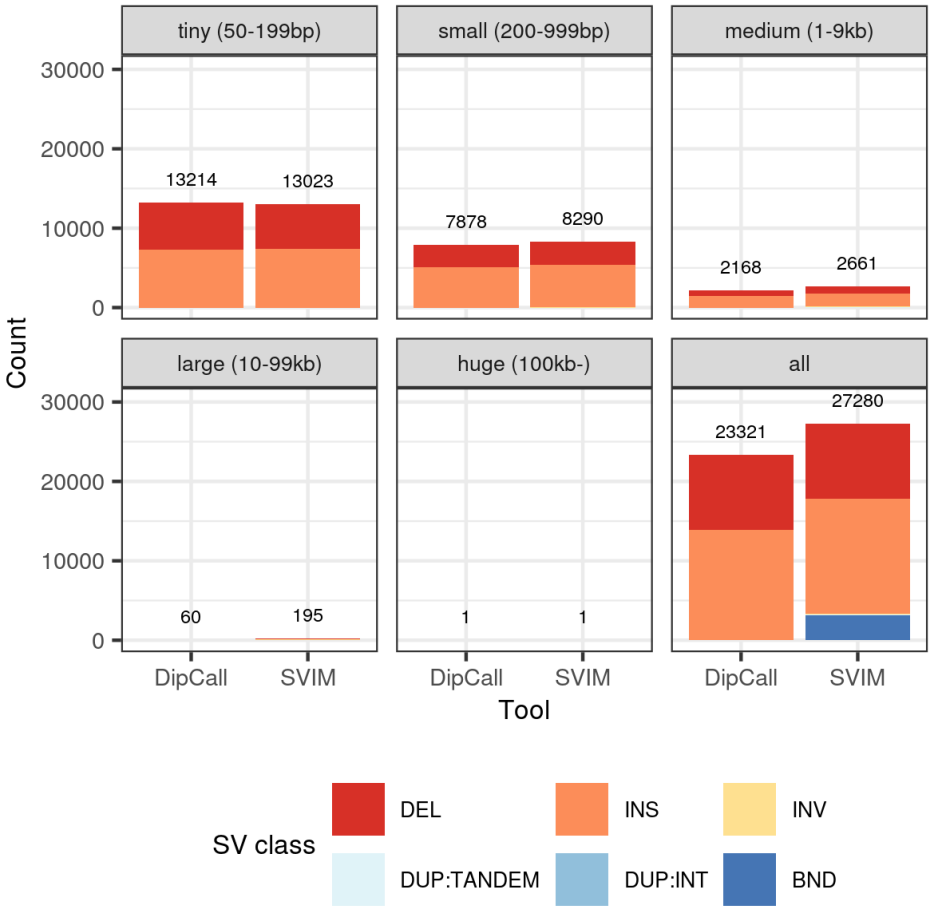


Figure 5.14: **Number of SV calls from Assembly A grouped into five size classes.** Shown is a stacked bar plot of SV classes represented by different colors. Each panel represents one size class and visualizes the number of calls in that size range called by *DipCall* and *SVIM-asm*. The bottom right panel shows the counts for all SV calls regardless of size. As translocation breakpoints (BND) do not have a size, they are included only in this bottom right panel.

6

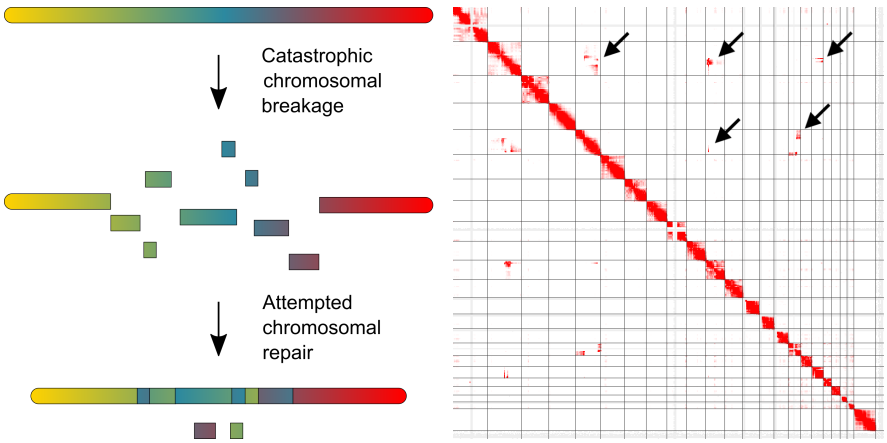
STRUCTURAL VARIANT DETECTION IN HIGHLY REARRANGED CHROMOSOMES

It is generally believed that human germline SVs form in isolated events involving a range of functional mechanisms (see Section 2.1.2.1). But three recently discovered molecular phenomena summarized under the term *chromoanagenesis* now challenge this assumption by introducing several rearrangements into the genome in a single event.

In this chapter, we describe the application of our SV detection method *SVIM* in the context of a larger research project investigating the mechanisms and consequences of chromoanagenesis. As part of this project, we analyzed PacBio data from a cohort of patients with highly rearranged genomes and detected both canonical SVs and novel adjacencies between distant genomic locations. To obtain a high-confidence set of novel adjacencies, we employed a multi-step filtering process and validated the final calls using orthogonal Hi-C data. The validation confirmed that the final callset was highly accurate and well-suited for downstream analysis, such as the complete reconstruction of rearranged chromosomes or the investigation of the molecular mechanisms behind chromoanagenesis.

6.1 THREE FORMS OF CHROMOANAGENESIS

The term chromoanagenesis encompasses three different phenomena which were described in the years 2011 and 2012 for the first time. The most prominent of them is called *chromothripsis* (from Greek *thripsis* for shattering). It describes the acquisition of a large number of chromosomal rearrangements in a single catastrophic event (see Figure 6.1) [67]. The acquired rearrangements are often complex and clustered in a limited number of genomic regions. Although the mechanisms causing chromothripsis are still under debate, the double-strand breaks forming the basis for such chromosomal re-



(a) **Chromothripsis mechanism** (b) **Hi-C map of a chromothriptic genome**

Figure 6.1: **Chromothripsis leads to a large number of chromosomal rearrangements through a single catastrophic event.**
a | Schematic view of a chromosome undergoing chromothripsis. A single catastrophic event shatters the chromosome into numerous fragments. When the cell attempts to repair the damage by reconnecting the fragments, multiple rearrangements are introduced and some fragments can get lost.
b | Hi-C map showing the 3-D contacts in a genome that underwent chromothripsis involving multiple chromosomes. Rows and columns represent the chromosomes while the color intensity visualizes the strength of interaction between two loci. Beside the expected intra-chromosomal contacts along the diagonal, several contacts between different chromosomes can be observed (black arrows) that indicate large inter-chromosomal rearrangements.

arrangements can occur by several mechanisms including aberrant DNA replication, ionizing radiation or the entry of DNA cutting enzymes into the cell nucleus [28, 74]. The DNA fragments resulting from the breaks are later reassembled by error-prone repair mechanisms, such as non-homologous end joining. Mistakes made by the repair mechanism can lead to the complex rearrangements that are the signature of chromothripsis.

Chromothripsis has been first reported in a patient with chronic lymphocytic leukemia and later confirmed to occur commonly in various types of human cancer, such as melanomas, sarcomas and gliomas [28, 89]. In cancer, chromothripsis can cause several tumor-promoting changes through one event which contrasts with the conventional theory of tumor progression through gradual changes.

Shortly after the first reports on chromothripsis in cancer, other studies described similar rearrangements in the germline causing congenital defects and genomic disorders [67]. While some of these cases showed very similar signatures to those of cancer chromothripsis, others exhibited pronounced differences [47, 61]. Unlike chromothripsis in cancer, which is mainly characterized by inversions and translocations, these latter rearrangements also comprised extensive duplication and triplication. These and other differences suggest that some germline rearrangements might be caused by other mechanisms than those active in chromothripsis. In particular, replicative processes and repair mechanisms, such as break-induced replication (BIR) might be involved. Instead of shattering and subsequent reassembly of chromosomes, these germline rearrangements might arise through replicative repair mechanisms which is why the term *chromoanasynthesis* (from Greek *anasynthesis* for reconstitution) has been suggested for such rearrangements [61].

Beside chromothripsis and chromoanasynthesis, a third related phenomenon known as *chromoplexy* (from Greek *pleko* for twisting) has been described [5]. Similar to chromothripsis, it is caused by double-strand breaks that are repaired by error-prone repair mechanisms. However, chromoplexy is characterized by a lower number of breakpoints from multiple chromosomes. Unlike in chromothripsis, the breakpoints are not clustered but distributed across the genome.

All three phenomena, chromothripsis, chromoplexy and chromoanasynthesis have in the literature been grouped under the umbrella term of *chromoanagenesis* (from Greek *anagenesis* for rebirth). Due to its recent discovery, chromoanagenesis is under active investigation by research groups worldwide and many questions remain unanswered [100].

6.2 DETECTION OF CANONICAL SVS IN A PATIENT COHORT USING SVIM

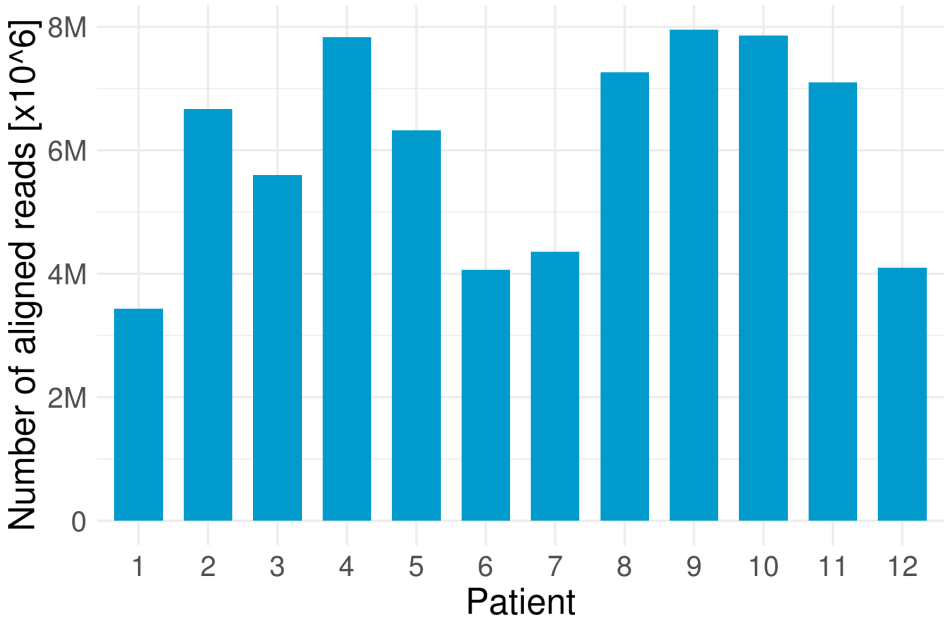
In a joint effort with collaborators from several research institutions, a cohort of 12 patients (in the following referred to as patients 1 through 12) with massive genomic germline rearrangements was collected. From the patients, lymphoblastoid or (in the case of patient 2)

fibroblast cell lines were obtained and analyzed using a comprehensive set of technologies. For each patient, three sequencing datasets were generated using Illumina sequencing, PacBio CLR sequencing and Hi-C. Ten of the twelve PacBio CLR datasets were generated on a PacBio Sequel II machine. Only patients 6 and 7 were sequenced on the older PacBio Sequel I. For most patients, only one sequencing run with a single SMRT cell was performed. The exceptions were patients 6 (11 SMRT cells), 7 (11 SMRT cells) and 11 (2 SMRT cells).

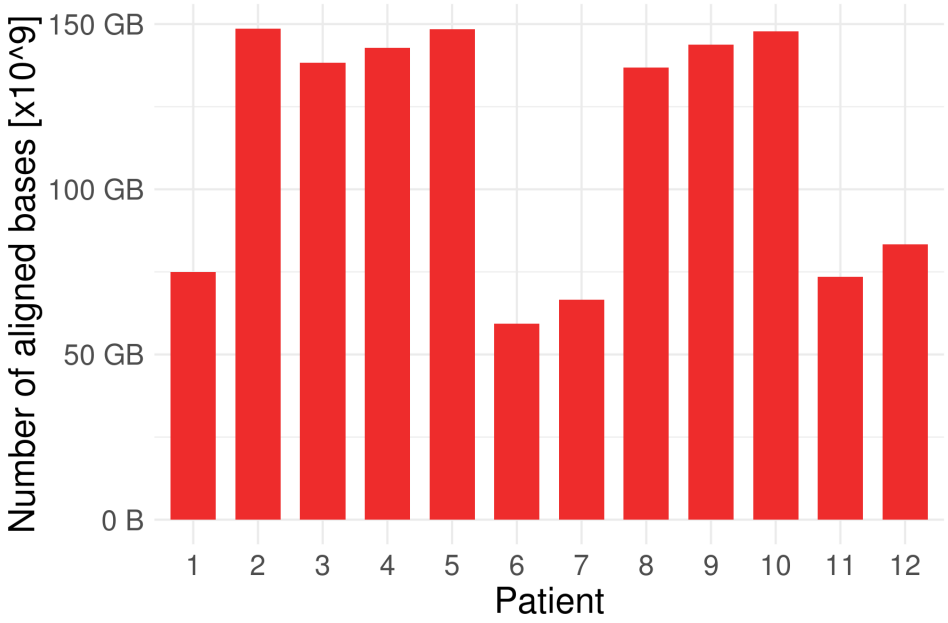
As the first step in our analysis, we aligned all datasets to the GRCh37 human reference genome. The amount of reads and bases sequenced varied greatly between the datasets. Therefore, the number of aligned reads and aligned bases also showed a high variance (see Figure 6.2). The number of aligned reads varied between 3,433,547 for patient 1 and 7,954,431 for patient 9. And while only 59.3 Gbp could be aligned for patient 6, the dataset for patient 2 comprised 148.6 aligned Gbp. Consequently, the alignment coverage of the datasets varied between 19.1x (patient 6) and 47.9x (patient 2) when assuming a genome size of approximately 3.1 Gbp.

We also observed a high variance between read lengths in different datasets (see Figure 6.3). The median read length varied between 6,839 bp for patient 11 and 24,389 bp for patient 3. In general, read lengths exhibited a right-skewed distribution. Most reads were shorter than 40 kbp with 39,694 bp (patient 3) being the highest third quartile observed among all patients. In the right tail of the distribution, numerous outliers with lengths of up to 248 kbp were present. Consequently, the mean read lengths were generally larger than the median read lengths with values between 12,896 bp and 27,083 bp.

We analyzed the read alignments of all 12 datasets with *SVIM* (v1.4.1) and detected five classes of SVs: deletions, insertions, inversions, tandem and interspersed duplications. The numbers of SVs from each class detected in each patient are visualized in Figure 6.4. In total, between 19,666 (patient 6) and 25,050 (patient 3) SVs were detected per sample. We observed a relationship between the number of aligned bases and the number of detected SVs. The datasets with the lowest number of bases (Patients 6, 7 and 11) also yielded fewer SVs. This relationship can be at least partly explained by the filtering which was applied on the callsets to retain only SVs supported by

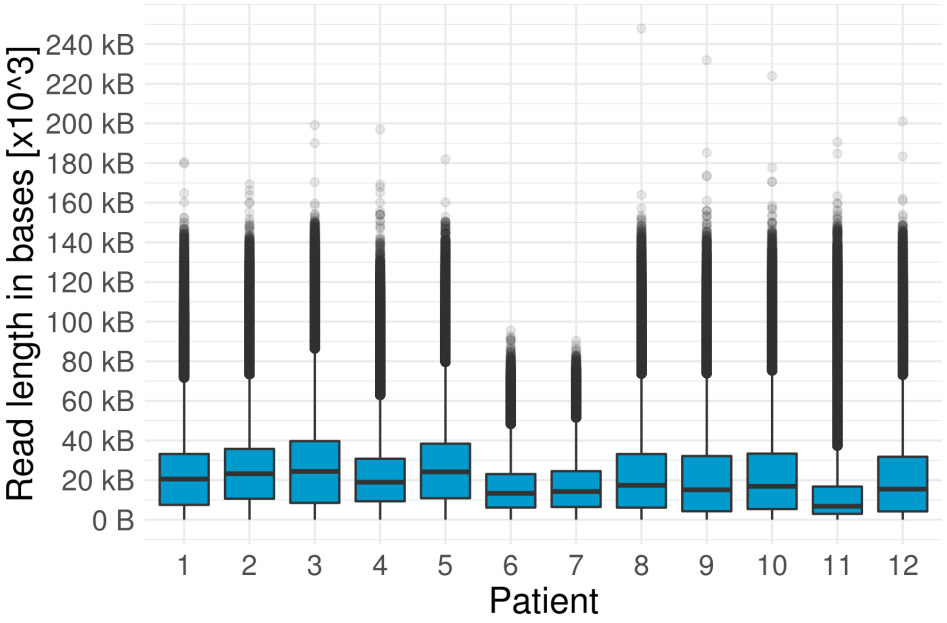


(a) **Aligned reads**

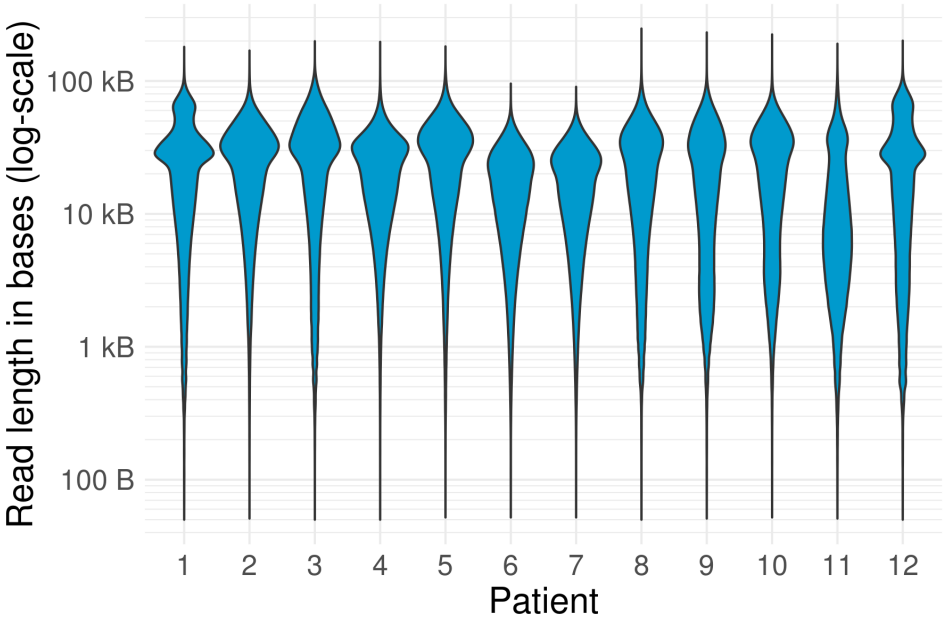


(b) **Aligned bases**

Figure 6.2: **PacBio CLR sequencing datasets from the patient cohort.** Shown is the number of aligned reads (panel a) and the number of aligned bases (panel b) for each dataset. Reads were aligned with *minimap2* on the GRCh37 human reference genome.



(a) **Boxplot on linear scale**



(b) **Violin plot on log scale**

Figure 6.3: **Distribution of read lengths in PacBio CLR sequencing datasets from the patient cohort.**
a | Each box in this boxplot visualizes the first and third quartiles (upper and lower end) and the median (middle bar in the box). The upper and lower whiskers extend to the largest/smallest value no further than 1.5 times the inter-quartile range from the upper/lower end of the box, respectively. Outliers are plotted with an alpha value of 0.1.
b | The violin plot visualizes the probability density at different values on a log-scaled y-axis.

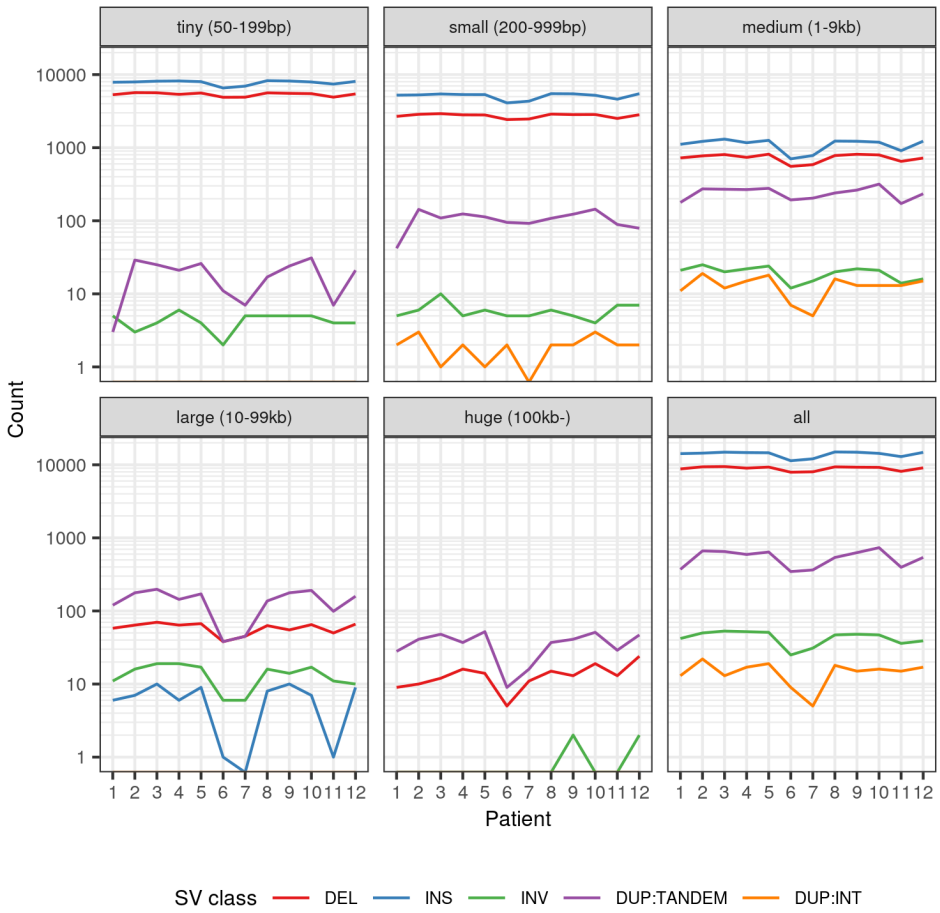


Figure 6.4: **Number of SVs detected in the patient cohort and supported by at least 5 reads.** Plotted is the number of SVs (on a log scale) detected in each patient. Five lines with different colors are shown for the different SV classes. SVs were grouped into five size classes which are represented by the first five panels. The last panel represents SVs of all size classes. DEL - deletion, INS - insertion, INV - inversions, DUP:TANDEM - tandem duplication, DUP:INT - interspersed duplication

at least 5 reads. Datasets with a deeper coverage provide a higher chance of passing this threshold.

With counts between 11,468 for patient 6 and 13,910 for patient 8, approximately half of the detected SVs were smaller than 200 bp. Another third was between 200 bp and 1 kbp in size (between 6,630 for patient 6 and 8,478 for patient 8). The most common SV class across all datasets were insertions (between 11,368 for patient 6 and 14,982 for patient 8) followed closely by deletions (between 7,918 for patient 6 and 9,449 for patient 3). All insertions together reached a cumulative size of several megabases (between 3.8 Mbp for patient 6

and 6.4 Mbp for patient 3). Deletions covered an even larger portion of the genome with a cumulative size between 9.8 Mbp for patient 6 and 33.2 Mbp for patient 12. Beside insertions and deletions, several hundred tandem duplications were detected in each dataset (between 346 for patient 6 and 734 for patient 10). Inversions (between 25 for patient 6 and 53 for patient 3) and interspersed duplications (5 for patient 7 and 22 for patient 2) were detected less frequently.

When we focus on large SVs with a size of more than 10 kbp, tandem duplications (between 47 for patient 6 and 246 for patient 3) and deletions (43 for patient 6 and 90 for patient 12) were the most frequent SV classes. Only few large insertions (between 0 for patient 7 and 10 for patients 3 and 9) were detected. The low number of large insertions can be easily explained by a lack of reads that were long enough to cover large insertions including their genomic context. In the size range above 100 kbp, no insertions were detected at all.

6.3 GENERATION AND VALIDATION OF A HIGH-CONFIDENCE SET OF NOVEL ADJACENCIES

The genomes from our patient cohort contained numerous genomic rearrangements that were indicative of chromoanagenesis. Because these rearrangements were likely acquired during a single catastrophic event they could not easily be categorized into the canonical SV classes, such as deletions or inversions. These classes describe isolated local rearrangements of the genome with the assumption that the genomic context of each SV remains largely the same. In chromoanagenesis, however, the large-scale structure of entire chromosomes is disrupted. Therefore, it is hard or even impossible to describe the rearrangements as a set of isolated local SVs because they tend to be nested and overlay each other.

Instead, a highly rearranged chromosome can be characterized as a chain of genomic fragments formed by splitting the reference genome at particular positions. The splitting positions, the orientations of the fragments and their order in the derivative chromosome are unknown but can be reconstructed from sequencing data. Below, we describe our approach of detecting and filtering novel adjacencies

between distant genomic locations from PacBio sequencing data and validating the final adjacencies with Hi-C data.

6.3.1 *Collection of novel adjacencies from PacBio alignments*

To identify the genomic fragments present in each genome and the connections between them, we aimed to collect the complete set of novel adjacencies between formerly distant genomic loci (i.e. translocation breakpoints). From PacBio data, novel adjacencies can be detected from split alignments of reads to multiple locations. Beside translocations, most other SV classes can also be expressed in terms of novel adjacencies. A deletion, for instance, creates a novel adjacency between its start and end locus. Similarly, an inversion is characterized by two novel adjacencies between its start and end locus each connecting the forward with the reverse strand.

We collected the novel adjacencies with our SV caller *SVIM* by implementing an additional operation mode (command line parameter `--all_bnds`). It collects all novel adjacencies indicated by the alignments including those from translocations, deletions, inversions, interspersed and tandem duplications. Simple insertions were not considered because the insertion of bases does not create novel adjacencies. Similar to the regular translocation breakpoint signatures collected by *SVIM*, all novel adjacencies were clustered and annotated with a confidence score (see Section 3.3.2).

6.3.2 *Filtering of novel adjacencies*

For downstream analysis, it is vital to keep the rate of false negatives and false positives in the set of adjacencies as low as possible. Even a small number of missing or erroneous adjacencies can have far-reaching consequences, e.g. in the reconstruction of rearranged chromosomes. Therefore, we inspected the *SVIM* calls using the orthogonal Hi-C data. We identified putative false positive calls and used the gained insights to develop five targeted filtering steps for the reduction of false positive calls in the set of novel adjacencies detected by *SVIM*. Although Hi-C data was used to develop and refine the filtering approach, the actual filtering steps were carried

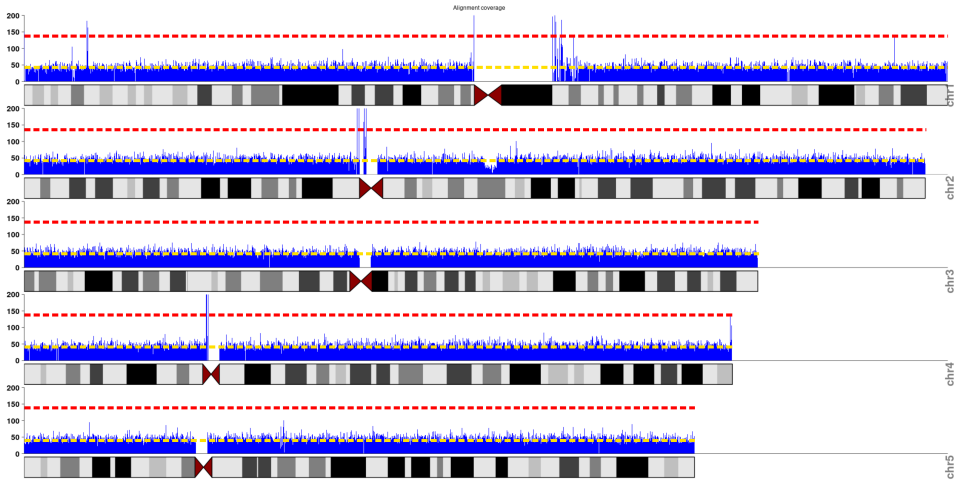


Figure 6.5: **Alignment coverage on chromosomes 1 to 5 of patient 4.** Each chromosome is represented by a grayscale ideogram with its characteristic banding pattern. Above each ideogram, blue bars visualize the average alignment coverage in non-overlapping genomic windows of 10 kbp. Yellow dashed lines represent the average alignment coverage (46x) across the whole genome. The red dashed lines represent the filtering threshold of 3 times that genome-wide average coverage.

out without the use of Hi-C data. Each of the following five steps was applied successively:

COVERAGE-BASED FILTERING Firstly, we analyzed the alignment coverage distribution across the genome by measuring the average alignment coverage in non-overlapping genomic windows of 10 kbp (see Figure 6.5). Generally, we observed a relatively uniform alignment coverage in most genomic regions. The most prominent exception were centromeric regions that contained segments with very low and very high coverages. We annotated windows with an average coverage higher than 3 times the genome-wide average coverage as *high-coverage regions*. Although there are multiple reasons for elevated coverages, they are often caused by CNVs, paralogous sequences missing from the reference genome or repetitive sequences attracting many reads [55]. Because such regions are prone to erroneous read mappings, all novel adjacencies found in high-coverage regions were filtered out.

GAP-BASED FILTERING In our cohort, we observed numerous novel adjacencies close to gaps in the reference genome. Instead

of having a proper nucleotide sequence, these gaps contain long stretches of the letter N which can represent any letter in the nucleotide alphabet [24]. Due to the lack of a sequence, alignment algorithms are not able to align reads to gap regions. Reads that cover a gap boundary therefore consist of two parts: The part outside of the gap can be aligned up to the gap whereas the part inside the gap cannot be aligned due to the lack of a reference sequence. When a region with similar sequence exists somewhere else in the genome, the latter part can sometimes be aligned to that region instead. This gives rise to a spurious novel adjacency that is only caused by the presence of the gap. Due to this problem, read alignments and particularly split alignments in the close proximity of reference gaps are unreliable. Therefore, we filtered out all novel adjacencies with a distance of less than 10 kbp to a reference gap.

DUPLICATION-BASED FILTERING Another source of unreliable read alignments in our cohort were *segmental duplications*. These duplicated genomic regions are larger than 1 kbp and have an identity of 90% or more to other regions in the genome [54]. Due to their length and similarity, segmental duplications often confuse the read alignment algorithm which leads to erroneous alignments, e.g. the alignment of reads to wrong genomic locations. In our cohort, we observed numerous novel adjacencies between related segmental duplication regions that were most likely caused by spurious split alignments. Therefore, we filtered out all novel adjacencies overlapping annotated segmental duplication regions.

SCORE-BASED FILTERING The remaining calls were filtered based on their confidence score. *SVIM* does not perform any filtering on its own so that many of the novel adjacency calls are supported by only a few reads. Most of these low-scoring calls are caused by errors in the PacBio data and the resulting misalignment of read segments or entire reads. Because the errors in PacBio data are randomly distributed, only individual reads are affected leading to novel adjacency calls with low score. Therefore, we defined score thresholds and filtered out adjacencies with scores lower than the threshold. Due to the large variance in alignment coverage across samples (see Figure 6.2) we used sample-specific score thresholds. On

average, we would expect heterozygous and homozygous variants to be supported by approximately 50% or 100% of the reads in the region, respectively. To avoid the removal of true adjacencies, however, we used a lenient threshold of 10% of the genome-wide average alignment coverage for each sample (e.g. filtering out calls with score < 5 for a sample with 50x coverage).

COHORT-BASED FILTERING When comparing calls between different samples, we observed considerable overlap. Therefore, we merged the remaining novel adjacencies from all patients and clustered similar adjacencies using a breakend distance cutoff of 1 kbp. Our analysis showed that non-unique adjacencies, i.e. those present in more than one sample, were particularly common in the repetitive genomic regions close to the centromeres and telomeres. We explain these adjacencies with systematic alignment errors that occur in multiple samples when mapping to the same reference genome. Some of the non-unique adjacencies might also reflect variants that are common in the population and therefore present in multiple patients. In our analysis, however, we were most interested in the genomic rearrangements related to chromoanagenesis which should be unique for each patient. Therefore, we filtered out adjacencies present in more than one sample.

Each of the five filtering steps removed a large number of novel adjacencies from the initial set (see Figure 6.6 with numbers for patient 4). To focus on large-scale rearrangements, the numbers below refer exclusively to long-range novel adjacencies, i.e. adjacencies between loci on different chromosomes or with a distance larger than 100 kbp on the same chromosome. The coverage-based filtering already removed between 2,835 (patient 1) and 5,709 (patient 10) of those adjacencies in high-coverage regions. Then, the gap-based filtering removed between 390 (patient 6) and 781 (patient 9) adjacencies close to gaps in the reference. Next, the duplication-based filtering removed between 959 (patient 1) and 3,434 (patient 7) adjacencies in segmental duplication regions. The score-based filtering removed the most calls with between 6,214 (patient 1) and 71,553 (patient 7). In the final cohort-based filtering step, between 42 (patient 6) and 76 (patient 12) non-unique adjacencies were removed. This left between

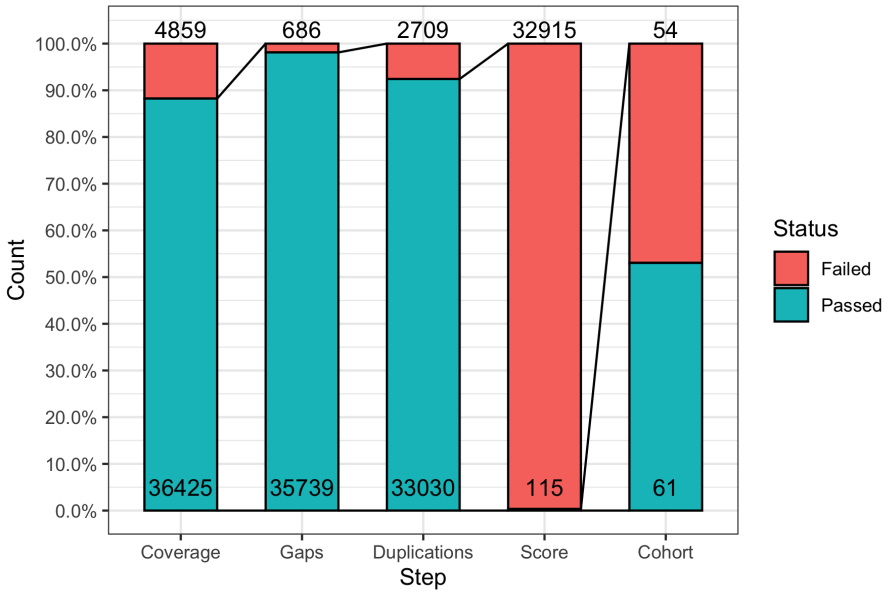


Figure 6.6: **Number of novel adjacencies in patient 4 passing and failing each filtering step.** Visualized are the five filtering steps that were successively applied to novel adjacencies. The numbers shown refer to novel adjacencies between different chromosomes or connecting two loci on the same chromosome with a distance larger than 100 kbp. The stacked colored bars visualize the fraction of adjacencies passing (aqua) and failing (red) each step. Absolute numbers are printed inside (passed) and above (failed) the bars. After the filtering, 61 high-confidence adjacencies remained for patient 4.

10 (patient 2) and 79 calls (patient 12) in the final sets of unique long-range adjacencies (see Figure 6.7).

6.3.3 *Validation of novel adjacencies using Hi-C*

To confirm that the applied filtering steps removed false positive calls instead of true positives, we investigated a subset of the filtered out adjacencies using Hi-C data. In Figure 6.8, several calls for patient 4 are overlaid on the Hi-C map of the same individual. In the Hi-C map, contact frequencies between two genomic loci (on the x-axis and y-axis, respectively) are visualized by the intensity of the red color. Pairs of distant genomic loci are located far from the diagonal and generally have a lower contact intensity than proximal loci closer to the diagonal. For true novel adjacencies between distant loci in a rearranged genome, however, sharp increases in contact intensity can

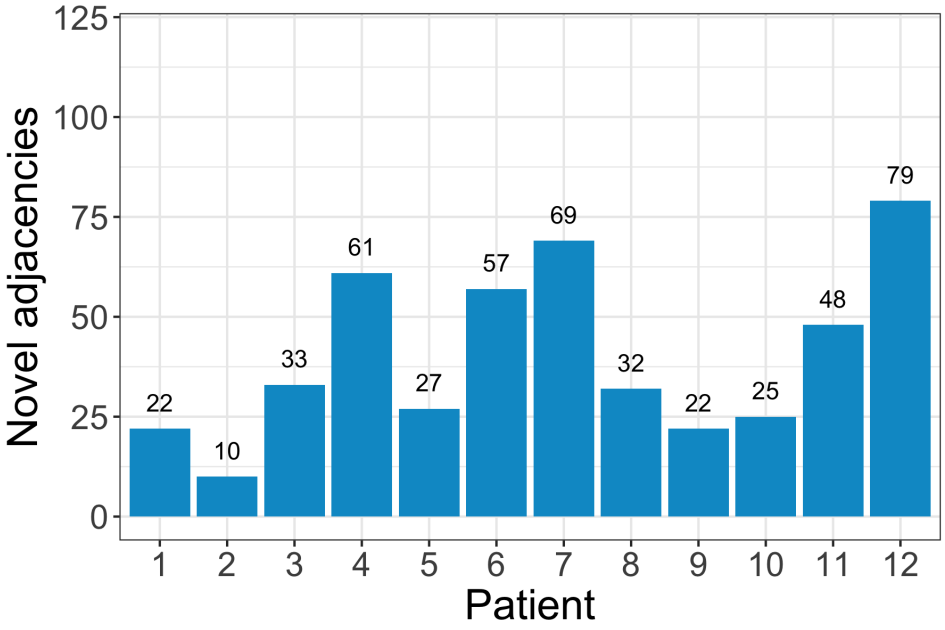
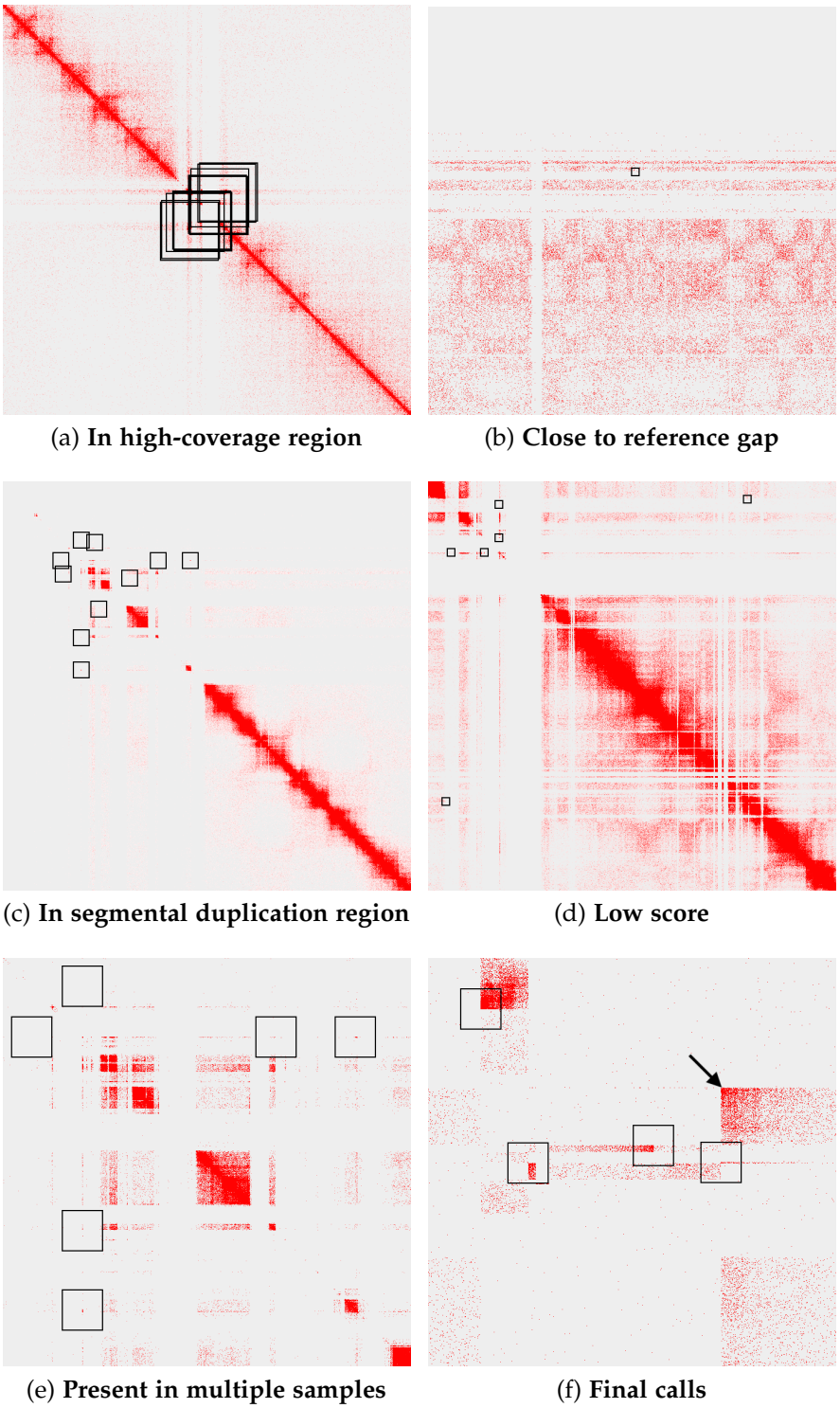


Figure 6.7: **Number of final novel adjacencies for each patient.** The barplot visualizes the number of novel adjacencies in each patient passing all five filtering steps. The numbers shown refer to novel adjacencies between different chromosomes or connecting two loci on the same chromosome with a distance larger than 100 kbp.

be observed at the breakpoint location. These prominent increases represent the proximity of two loci in 3-D space that are distant in the reference genome but close in the genome under investigation. None of the adjacencies that were filtered out due to high coverage (panel a), gaps in the reference (panel b), segmental duplication regions (panel c), low score (panel d), or presence in multiple samples (panel e) showed any substantial support by Hi-C. They were either located in regions of low mappability (gray regions with a low number of mapped Hi-C reads) or regions without any prominent increases in contact frequency. The final calls shown in panel f, in contrast, were well supported by sharp increases in contact intensity at the breakpoint locations.

Yet, not all sharp edges observed in the Hi-C map were caused by direct adjacencies. Nested rearrangements between chromosomes also led to indirect (i.e. more distant) adjacencies that could be observed in the map as sharp edges. For most applications, however, only direct adjacencies are useful. While it was hard to distinguish direct and indirect adjacencies from Hi-C data alone (see Figure 6.8,



(a) **In high-coverage region** (b) **Close to reference gap**

(c) **In segmental duplication region** (d) **Low score**

(e) **Present in multiple samples** (f) **Final calls**

Figure 6.8: **Selected parts of the Hi-C map for patient 4 around filtered and final novel adjacencies.** In the Hi-C map, contact frequencies between two genomic loci are visualized by the color intensity. Regions with low mappability, such as repeats or reference gaps, appear as gray areas. Novel adjacencies detected in the PacBio data are plotted as black rectangles centered around the precise breakpoint. In the first five panels a) to e), novel adjacencies are shown that were removed in each of the five filtering steps. The last panel f) shows final novel adjacency calls. The black arrow in panel f) points to an indirect adjacency caused by a nested rearrangement.

panel f), the indirect adjacencies were correctly not supported by the PacBio alignments illustrating how well PacBio and Hi-C complement each other.

As the final step in our analysis, we compared the final callset of long-range novel adjacencies for patient 4 with a manually curated gold standard set of breakpoints. To produce the gold standard, the complete unfiltered set of long-range novel adjacencies detected by SVIM was obtained. After adjacencies supported by less than 3 PacBio reads were removed, the remaining 631 calls were displayed on the Hi-C map of patient 4 and inspected by two expert curators. The curators examined each call for support by the Hi-C data and thus produced a gold standard set of 65 high-confidence adjacencies supported by both Hi-C and PacBio.

When we compared the final callset of 61 unique long-range adjacencies detected for patient 4 with this gold standard set, we observed a very good concordance. 56 out of 65 (recall of 86.2%) adjacencies from the gold standard set were contained in the filtered callset. When we inspected the nine adjacencies missing from the filtered callset, we found that they had been filtered out in the coverage-based filtering step (2 adjacencies), the duplication-based filtering step (2 adjacencies) and the cohort-based filtering step (5 adjacencies), respectively. A closer inspection revealed five of them as likely artifacts caused by segmental duplications. All other adjacencies were present in several other samples of the cohort. We therefore concluded that the gold standard set still contains artifacts that were successfully detected by our filtering approach.

Conversely, 56 out of 61 (precision of 91.8%) adjacencies from the filtered callset were contained in the gold standard set. We also inspected the five adjacencies missing from the gold standard set. In four out of the five cases we found good support in the PacBio data but little to no evidence by Hi-C. Two of these adjacencies were located in regions of low short-read mappability which would explain the missing support by Hi-C. For all five cases it was hard to judge whether they represented real adjacencies or artifacts.

The validation of the final callset for patient 4 confirmed that the applied filtering steps were able to produce an accurate callset of novel adjacencies for a highly rearranged genome. The callset will be a valuable resource for different applications, such as the complete

reconstruction of derived chromosomes or the investigation of chromoanagenesis mechanisms. Our analysis illustrated that PacBio and Hi-C are complementary technologies for the study of large-scale genomic rearrangements. While PacBio reads provided local information on the precise locations of and the sequence around genomic breakpoints, Hi-C provided a global but low-resolution map of the rearrangements.

In this thesis, we discussed two complementary paradigms for the investigation of genomic structural variation: While the classification of structural variation into a fixed set of canonical classes is a good framework for simpler variants, the decomposition of complex rearrangements into novel adjacencies presents itself as the more suitable approach for highly rearranged genomes. With our analysis in the context of this research project, we could demonstrate that *SVIM* can assist both in the detection of canonical SV as well as the resolution of more complex chromosomal rearrangements.

7

DISCUSSION

Structural variation is, next to single-nucleotide variation and small indels, one of the main classes of genetic variation. The large size of SVs and their strong influence on human phenotype and disease make them an important research target. However, their unique properties and the weaknesses of traditional sequencing technologies complicate the detection and characterization of SVs. Third-generation sequencing technologies, such as PacBio SMRT sequencing and ONT Nanopore sequencing, have the potential to alleviate these problems through the generation of long but relatively inaccurate reads.

In this thesis, we discussed the problem of detecting different classes of SVs in third-generation sequencing datasets. The main challenges that we met were the relatively high error rate of the data, the great diversity of SVs and the repetitive nature of the human genome. To overcome these obstacles, we introduced a novel SV detection method, *SVIM*, that employs a four-step pipeline to accurately detect and genotype SVs from long-read alignments. *SVIM* combines an hierarchical clustering approach with a novel distance metric to merge signatures of the same SV despite discrepancies caused by sequencing or alignment errors. For each SV, a confidence score is computed that facilitates the separation of true calls from artifacts. Furthermore, *SVIM* estimates the genotype of SVs from the ratio of read alignments supporting or contradicting each variant. While existing methods do not distinguish different types of duplications, *SVIM* is, to our knowledge, the first method to detect interspersed duplications from split alignments.

A comprehensive comparison of *SVIM* with two competing tools, *Sniffles* and *pbsv*, on simulated and real datasets demonstrated that our method combines a high recall with a high precision. On the simulated datasets, *SVIM* achieved the best results among the three tools followed closely by *Sniffles*. Deletions, inversions and tandem duplications could be detected best with F1 scores above 98% (homozygous) and 95% (heterozygous) even for coverages as low as

15x. Insertions proved slightly harder to detect with F1 scores above 92% (homozygous) and 86% (heterozygous). On two real PacBio datasets, *SVIM* performed best among the tools when comparing the genotyped variant calls with a gold standard set of large deletions and insertions. On low-coverage subsamples of the datasets or when variant calls without genotypes were compared, *pbsv* achieved the best results. Unlike *Sniffles* and *SVIM*, however, *pbsv* was among the methods used to compile the gold standard which could have represented a slight bias in favor of *pbsv* in our evaluation. On the real Nanopore dataset where *SVIM* was compared only to *Sniffles*, our method consistently achieved better results with a particularly wide margin on genotyped variant calls. All in all, our experiments suggest that *SVIM* outperforms *Sniffles* in most settings. Depending on the scenario, *pbsv* achieved mostly comparable but sometimes slightly better or worse results than *SVIM*. That *SVIM* performed best in our comparisons with genotyped variant calls indicates that our method estimates the most accurate genotypes of all three tools.

Yet, we would like to note that the evaluation of SV callers remains difficult in the absence of a complete and accurate truth set of SVs. Simulated datasets, like the one we generated, provide such a truth set but are often not able to emulate the full complexity of real data. For evaluations on real datasets, in contrast, only few suitable truth sets are available. The GIAB consortium combined numerous technologies and methods to generate a benchmark set for HG002 that we used in our experiments. However, that set was limited to isolated deletions and insertions leaving out other SV classes and SVs in difficult regions [102]. Furthermore, the evaluation and comparison of SV callers requires the choice of particular sequencing datasets, tool versions, tool parameters and filtering approaches. Although we made these choices with the aim of a fair comparison in mind, changes in any of these parameters could produce slightly different results.

Generally, our experiments showed that SV detection performance increases for higher coverages but saturates at a certain point where the methods do not benefit from even more data. Compared to PacBio CLR and Nanopore data, PacBio CCS enabled higher performances at lower coverages. This improved performance is enabled by the computational construction of an accurate consensus sequence

from multiple sequencing passes of the same genomic fragment. It comes at the expense of shorter read lengths and a considerably higher cost (approximately 3× the cost of CLR sequencing) [65]. In our experiments, a PacBio CCS coverage of only 10x was sufficient to produce variant calls with F1 scores above 90% (84% for calls with genotypes). PacBio CLR and Oxford Nanopore, in contrast, required much higher coverages (24x and 36x, respectively) to reach the same level of performance. This approximately threefold increase in required coverage that we observed for PacBio CLR and Nanopore data nearly balances the similar increase in cost for the PacBio CCS data. Moreover, its considerably higher accuracy makes CCS data more broadly applicable than the other TGS technologies. Unlike those, CCS data enables the accurate detection of small variants, the analysis of even more complex and repetitive genomic regions and the confident separation of haplotypes for diploid genome assembly.

Beside the analysis of long-read alignments, we also discussed the analysis of genome assemblies for SV detection. The process of de novo genome assembly is usually independent of existing reference genomes which makes the process free of any reference bias but also computationally demanding and time-consuming. To detect SVs in the sequence contigs produced by the assembly, the contigs need to be compared to another genome assembly or a reference genome. To this end, an alignment between the two assemblies is produced and analyzed with an SV caller. While several methods exist for haploid assemblies, we are aware of only one other method, *DipCall*, for the detection and genotyping of SVs from diploid genome assemblies. By adapting our method *SVIM*, we implemented a new caller for diploid assemblies, *SVIM-asm*, that detects more classes of SVs and estimates more accurate genotypes than *DipCall*. We confirmed the improved calling and genotyping performance of *SVIM-asm* on two publicly available diploid genome assemblies.

The classification of SVs into distinct classes produces a structured catalogue of differences between a macroscopically similar pair of genomes. For genomes with a disrupted structure or a large number of major rearrangements, however, the detection and classification of isolated SVs meets its limits. In this scenario, the decomposition of complex rearrangements into novel adjacencies presents itself as a suitable alternative. For a set of patients with chromoanagenesis, we

used *SVIM* to collect a comprehensive callset of novel adjacencies which was subsequently filtered. The filtering removed more than 99% of all calls which reflects the high error rate of the PacBio data. While variant calling pipelines typically impose a strict cutoff on the confidence score or the number of supporting reads, we wanted to retain the high sensitivity of our callset. Therefore, we used a low score cutoff and combined it with multiple filtering steps based on genomic features and a cross-sample analysis. Eventually, we validated the final callset using orthogonal Hi-C data and demonstrated its high sensitivity and precision. This makes our approach well-suited for applications that require a complete and accurate list of long-range genomic rearrangements, e.g. the investigation of gene regulation around genomic breakpoints or the study of breakpoint pathogenicity.

A major limitation of the approaches discussed in this thesis is their dependence on the correctness of the analyzed sequence alignments. Similar to other SV callers, *SVIM* and *SVIM-asm* are able to detect only rearrangements that are already indicated by the layout of the alignments. One problem is posed by the repetitive nature of many genomes, due to which many sequences map ambiguously or cannot be mapped confidently. These artifacts affect the sensitivity and precision of our tools but might also cause misclassification of variants, e.g. the classification of interspersed duplication from mobile elements as simple insertions.

Another limitation is that *SVIM* and *SVIM-asm* are currently unable to detect complex structural variation. Although this class of variation is much rarer than the six canonical SV classes discussed in this thesis, they are generally larger and more likely to disrupt genes or regulatory regions [16]. Therefore, we plan to extend our methods with the ability to detect complex SVs. Furthermore, we intend to add multi-threading support to *SVIM* to accelerate the processing of very large sequencing datasets.

APPENDIX

A

PROOF OF METRIC AXIOMS FOR SPAN-POSITION DISTANCE

A.1 DEFINITIONS

A metric on a set X is a function $d : X \times X \to [0, \infty)$.
For all $x, y, z \in X$, the following three axioms need to be satisfied:

1. Identity of indiscernibles: $d(x, y) = 0 \Leftrightarrow x = y$
2. Symmetry: $d(x, y) = d(y, x)$
3. Triangle inequality: $d(x, y) \leq d(x, z) + d(z, y)$

Span-position distance is a function $SPD : S \times S \to [0, \infty)$ where S is the set of SV signatures $(S.type, S.chrom, S.start, S.end)$. $S.type$ is the signature type, $S.chrom$ is the chromosome, $S.start$ is the genomic start position and $S.end$ is the genomic end position. Span-position distance is defined as the sum of two components: $SPD = SD + PD$.
SD is a function $SD : S \times S \to [0, 1)$ on the set S. It is defined as the relative difference in span between two signatures x and y: $SD = \frac{|x.span - y.span|}{max(x.span, y.span)}$ where $x.span = x.end - x.start$ and $y.span = y.end - y.start$.
PD is a function $PD : S \times S \to [0, \infty)$ on the set S. It is defined as the absolute difference between the center positions of two signatures x and y scaled by a user-defined scaling constant: $PD = \frac{|x.center - y.center|}{N}$ where $x.center = \frac{x.start + x.end}{2}$, $y.center = \frac{y.start + y.end}{2}$ and $N \in \mathbb{N}$.
In the following, we show that span-position distance satisfies the three axioms of a metric.

A.2 IDENTITY OF INDISCERNIBLES

Consider two identical SV signatures $x = y$. Because they are identical, they have the same signature type and chromosome as well as the

same start and end coordinates. Therefore $SD(x,y) = 0 \wedge PD(x,y) = 0 \Rightarrow SPD(x,y) = 0$.
Consider two indiscernible SV signatures x and y for which $SPD(x,y) = 0$. From the definition of the span-position distance for signatures on different chromosomes or with different types, it follows that x and y must have the same type $x.type = y.type$ and chromosome $x.chrom = y.chrom$. Because SPD is a sum which is 0, it follows that $SD(x,y) = 0 \wedge PD(x,y) = 0$. From the definitions of SD and PD, it follows that $x.span = y.span \Rightarrow (x.end - x.start) = (y.end - y.start)$ and $x.center = y.center \Rightarrow (x.start + x.end) = (y.start + y.end)$. It follows that $x.start = y.start$ and $x.end = y.end$. Because all four components of x and y are identical, it follows that x and y are identical.

A.3 SYMMETRY

$SPD(x,y) = SPD(y,x)$ follows from the definitions of SD and PD. The numerator of SD is the absolute value of the difference in spans which does not change if x and y are swapped. The denominator of SD is the maximum value of the spans which does not change either. Therefore, SD does not change if x and y are swapped.
The numerator of PD is the absolute value of the difference in position which does not change if x and y are swapped. The denominator of PD is a natural number independent of x and y. Because SD and PD do not change if x and y are swapped, it follows that $SPD(x,y) = SPD(y,x)$.

A.4 TRIANGLE INEQUALITY

To show that the triangle inequality is satisfied by the span-position distance SPD, we first show that the triangle inequality is satisfied by the span distance SD and the position distance PD.

A.4.1 *Span distance*

The triangle inequality for the span distance:
$SD(x,y) \leq SD(x,z) + SD(z,y)$.

It is easy to show that if two of x,y,z have the same span, one of $SD(x,y)$, $SD(x,z)$ and $SD(z,y)$ is 0 and in this case the inequality is satisfied.
Therefore, we assume that all spans are different. Next, we assume that $x.span < y.span$ (the inverse case $y.span < x.span$ is similar). Then there are 3 possibilities: $z.span < x.span < y.span$, $x.span < z.span < y.span$ and $x.span < y.span < z.span$.
In the first case $SD(x,y) = \frac{y.span - x.span}{y.span}$ and $SD(z,y) = \frac{y.span - z.span}{y.span}$ so that $SD(x,y) < SD(z,y)$. In the third case $SD(x,y) = \frac{y.span - x.span}{y.span} = 1 - \frac{x.span}{y.span}$ and $SD(x,z) = \frac{z.span - x.span}{z.span} = 1 - \frac{x.span}{z.span}$ so that $SD(x,y) < SD(x,z)$. In both cases, the strict inequality $SD(x,y) < SD(x,z) + SD(z,y)$ applies.
In the second case, $SD(x,y) = \frac{y.span - x.span}{y.span}$, $SD(x,z) = \frac{z.span - x.span}{z.span}$ and $SD(z,y) = \frac{y.span - z.span}{y.span}$.
Then,

$$
\begin{aligned}
SD(x,y) &\leq SD(x,z) + SD(z,y) \\
\Rightarrow \frac{y.span - x.span}{y.span} &\leq \frac{z.span - x.span}{z.span} + \frac{y.span - z.span}{y.span} \\
\Rightarrow \frac{y.span - x.span - y.span + z.span}{y.span} &\leq \frac{z.span - x.span}{z.span} \\
\Rightarrow \frac{z.span - x.span}{y.span} &\leq \frac{z.span - x.span}{z.span} \\
\Rightarrow \frac{1}{y.span} &\leq \frac{1}{z.span} \\
\Rightarrow z.span &\leq y.span
\end{aligned}
$$

, which is true.

A.4.2 *Position distance*

The triangle inequality for the position distance:
$PD(x,y) \leq PD(x,z) + PD(z,y)$
The position distance is defined as the absolute distance between the centers $x.center = \frac{x.start + x.end}{2}$ and $y.center = \frac{y.start + y.end}{2}$ of the signatures x and y scaled by a user-defined scaling constant N.

Because N is a constant we can ignore it in this proof of triangle inequality.
It is easy to show that if two of x,y,z have the same center, one of $PD(x,y)$, $PD(x,z)$ and $PD(z,y)$ is 0 and in this case the inequality is satisfied.
Therefore, we assume that all centers are different. Next, we assume that $x.center < y.center$ (the inverse case $y.center < x.center$ is similar). Then there are 3 possibilities: $z.center < x.center < y.center$, $x.center < z.center < y.center$ and $x.center < y.center < z.center$.
In the first case $PD(x,y) = \frac{y.center - x.center}{N}$ and $PD(z,y) = \frac{y.center - z.center}{N}$ so that $PD(x,y) < PD(z,y)$. In the third case $PD(x,y) = \frac{y.center - x.center}{N}$ and $PD(x,z) = \frac{z.center - x.center}{N}$ so that $PD(x,y) < PD(x,z)$. In both cases, the strict inequality $PD(x,y) < PD(x,z) + PD(z,y)$ applies.
In the second case, $PD(x,y) = \frac{y.center - x.center}{N}$, $PD(x,z) = \frac{z.center - x.center}{N}$ and $PD(z,y) = \frac{y.center - z.center}{N}$.
Then,

$$
\begin{aligned}
PD(x,y) &\leq PD(x,z) + PD(z,y) \\
\frac{y.center - x.center}{N} &\leq \frac{z.center - x.center}{N} + \frac{y.center - z.center}{N} \\
y.center - x.center &\leq (z.center - x.center) + (y.center - z.center) \\
y.center - x.center &\leq y.center - x.center
\end{aligned}
$$

, which is true.

A.4.3 *Span-position distance*

Above we showed that span distance and position distance satisfy the triangle inequality. It follows that span-position distance also satisfies the triangle inequality:

$$
\begin{aligned}
SD(x,y) &\leq SD(x,z) + SD(z,y) \wedge \\
PD(x,y) &\leq PD(x,z) + PD(z,y) \\
\Rightarrow SD(x,y) + PD(x,y) &\leq SD(x,z) + PD(x,z) + SD(z,y) + PD(z,y) \\
\Rightarrow SPD(x,y) &\leq SPD(x,z) + SPD(z,y)
\end{aligned}
$$

B

SUPPLEMENTARY FIGURES

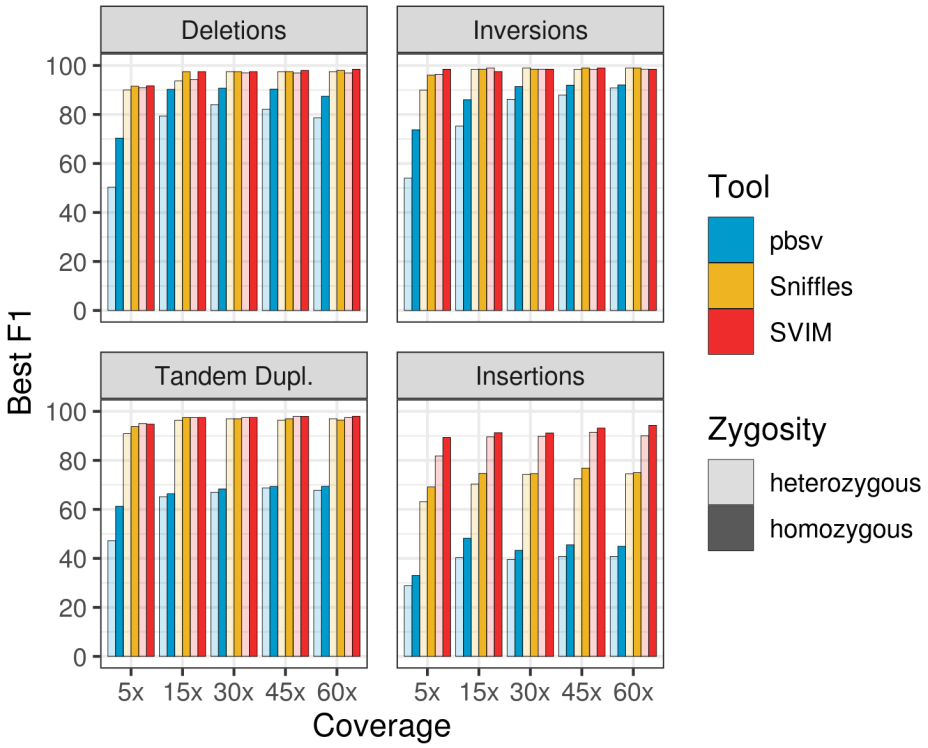


Figure B.1: **Best SV detection performance for five different simulated coverage levels (*ngmlr* alignments).** Shown are the best F1 scores (y-axis) reached by each tool for different read coverages between 5x and 60x (x-axis). Generally, higher coverages enabled higher F1 scores. F1 scores were calculated requiring a maximal distance of 1 kbp and a span difference of less than 0.3 between matching variant calls and the original simulated variants. For *Sniffles* and *SVIM*, reads were aligned using *ngmlr* while *pbsv* required input reads to be aligned with *pbmm2*.

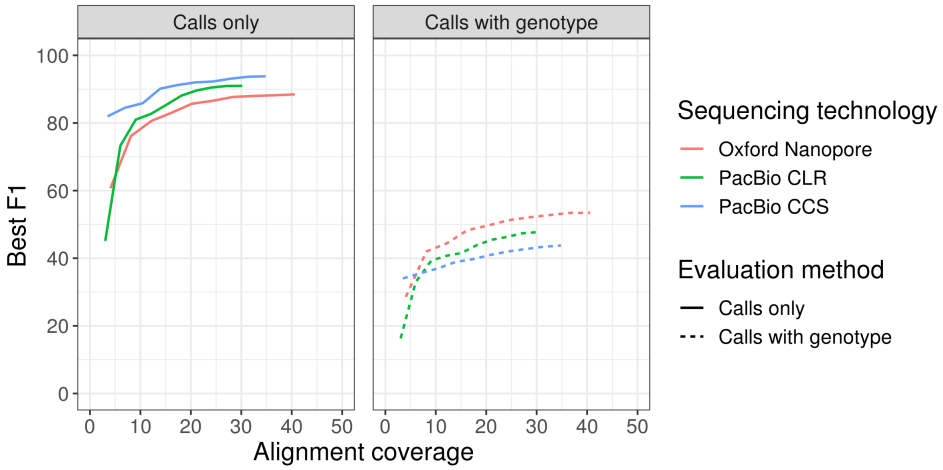


Figure B.2: **Best SV detection performance reached by *Sniffles* on sequencing datasets of different coverage levels from different technologies.** Plotted are the best F1 scores achieved by *Sniffles* (y-axis) against the alignment coverage (x-axis) of the sequencing dataset (represented by the line color). Results for simple and genotyped calls are visualized in left and right panels by solid and dashed lines, respectively. F1 scores were calculated requiring a maximal distance of 1 kbp and a span difference of less than 0.3 between matching variant calls and the original simulated variants. Reads were aligned using *minimap2*.

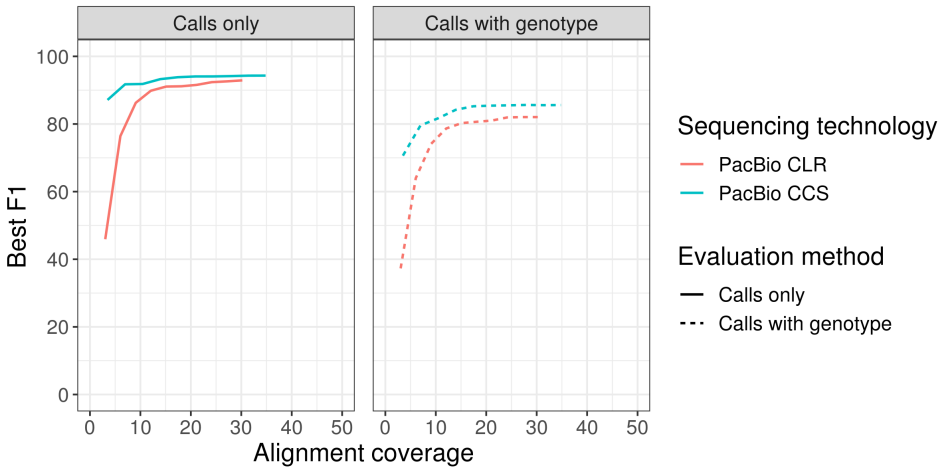
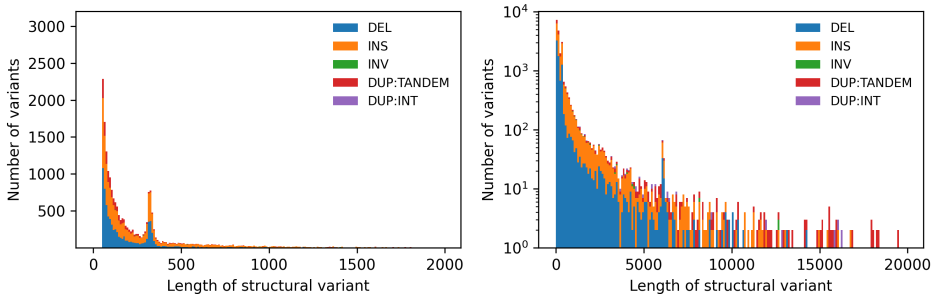
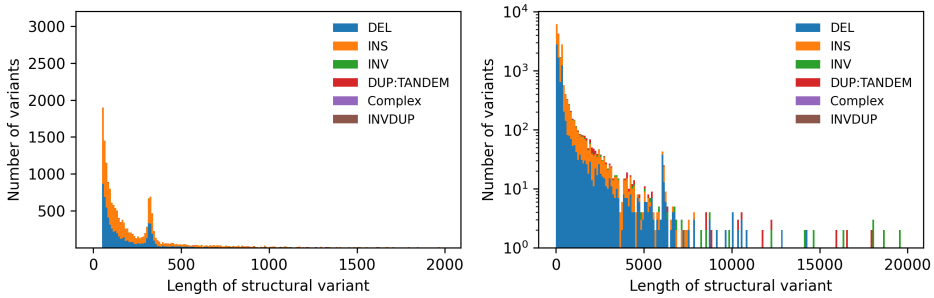


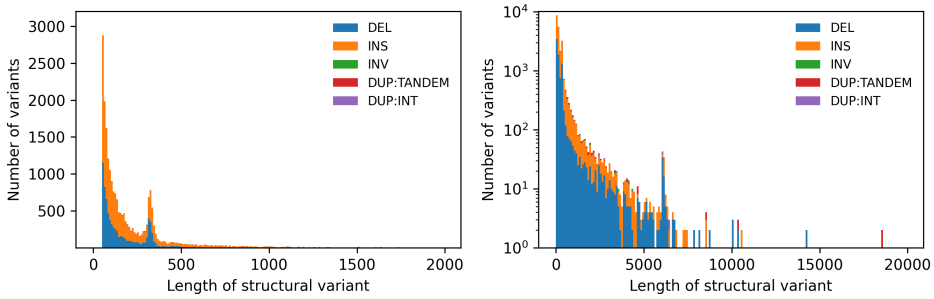
Figure B.3: **Best SV detection performance reached by *pbsv* on sequencing datasets of different coverage levels from different technologies.** Plotted are the best F1 scores achieved by *pbsv* (y-axis) against the alignment coverage (x-axis) of the sequencing dataset (represented by the line color). Results for simple and genotyped calls are visualized in left and right panels by solid and dashed lines, respectively. F1 scores were calculated requiring a maximal distance of 1 kbp and a span difference of less than 0.3 between matching variant calls and the original simulated variants. Reads were aligned using *minimap2*.



(a) **pbsv**

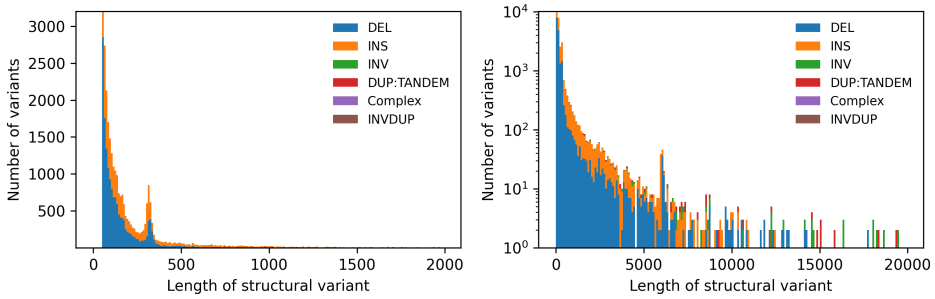


(b) **Sniffles**

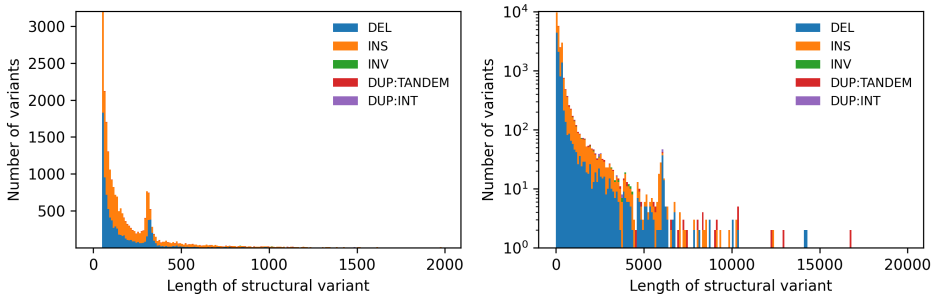


(c) **SVIM**

Figure B.4: **Size distribution of SVs detected in the 38.7x PacBio CLR dataset.** Shown are stacked histograms of SV classes represented by different colors. In the left column, SV sizes up to 2 kbp are plotted with a bin size of 10 bp. In the right column, SV sizes up to 20 kbp are plotted on a logarithmic y-axis with a bin size of 100 bp. The top, middle and bottom panels, visualize callsets by *pbsv*, *Sniffles* and *SVIM*, respectively. All callsets were generated with a confidence threshold of 5. To simplify the comparison, SV class names have been harmonized between the tools.



(a) **Sniffles**



(b) **SVIM**

Figure B.5: **Size distribution of SVs detected in the 50.7x Oxford Nanopore dataset.** Shown are stacked histograms of SV classes represented by different colors. In the left column, SV sizes up to 2 kbp are plotted with a bin size of 10 bp. In the right column, SV sizes up to 20 kbp are plotted on a logarithmic y-axis with a bin size of 100 bp. The top and bottom panels, visualize callsets by *Sniffles* and *SVIM*, respectively. All callsets were generated with a confidence threshold of 5. To simplify the comparison, SV class names have been harmonized between the tools.

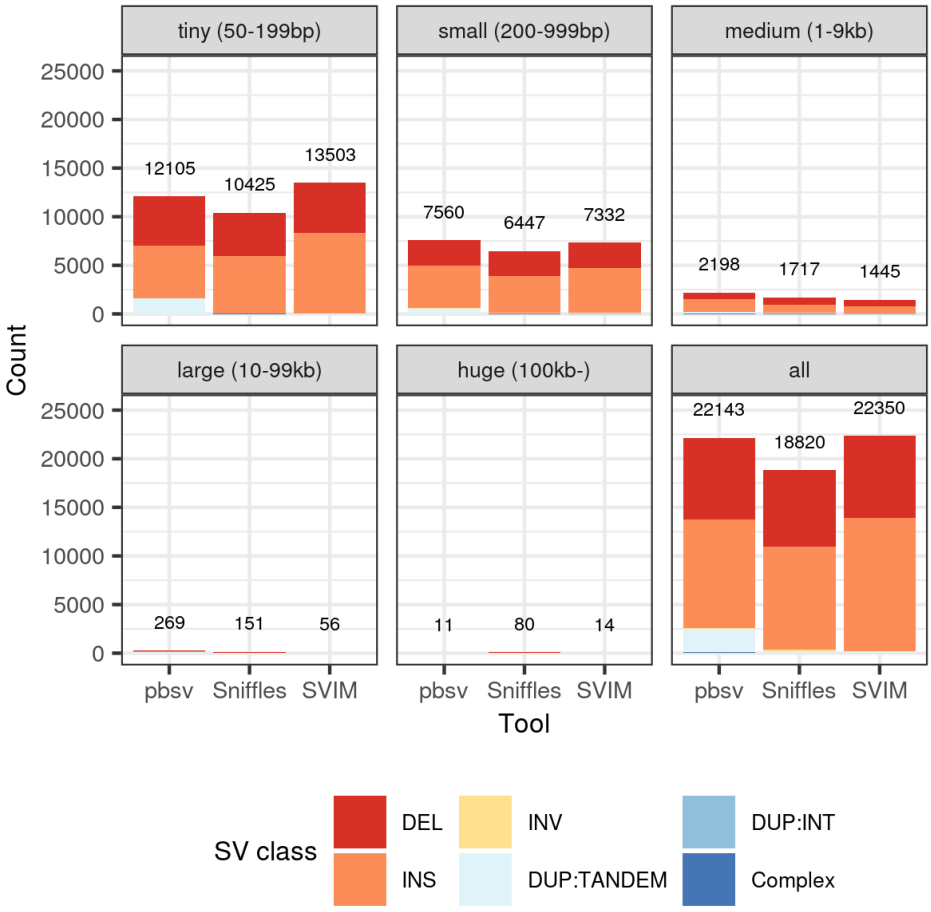


Figure B.6: **Number of SV calls from the 38.7x PacBio CLR dataset stratified into five size classes.** Shown is a stacked bar plot of SV classes represented by different colors. Each panel represents one size class and visualizes the number of calls in that size range called by *pbsv*, *Sniffles* and *SVIM*. All callsets were generated with a confidence threshold of 5. The bottom right panel shows the counts for all SV calls regardless of size.

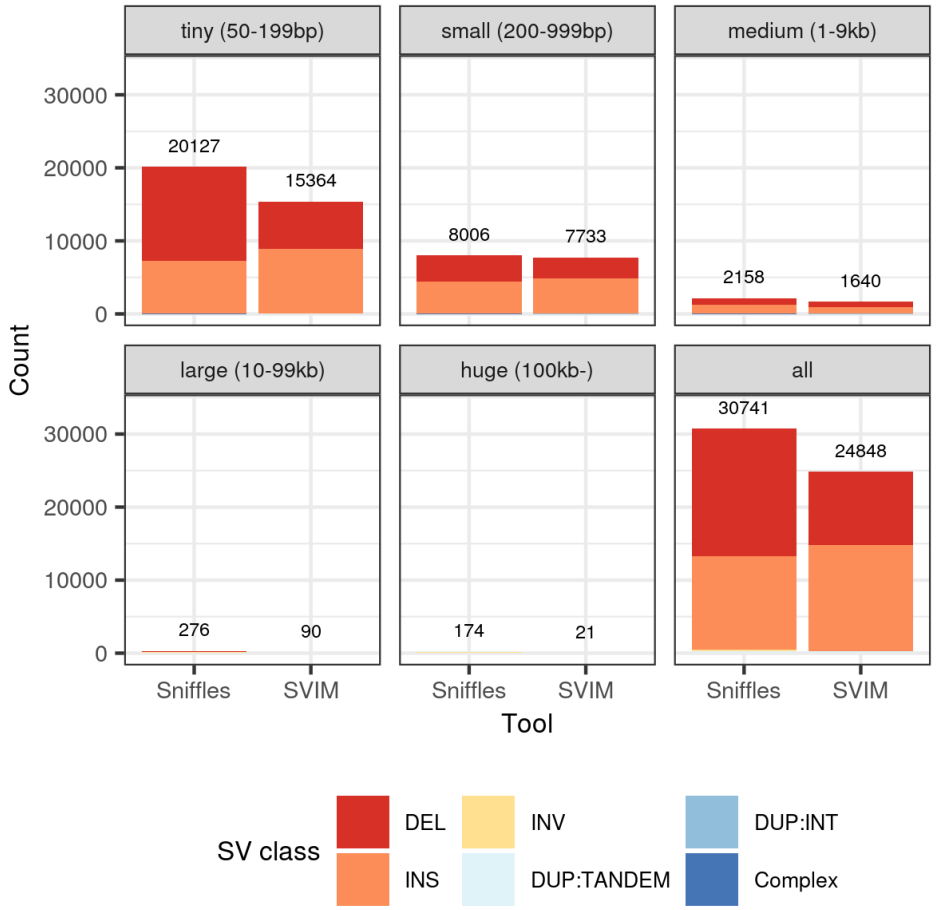
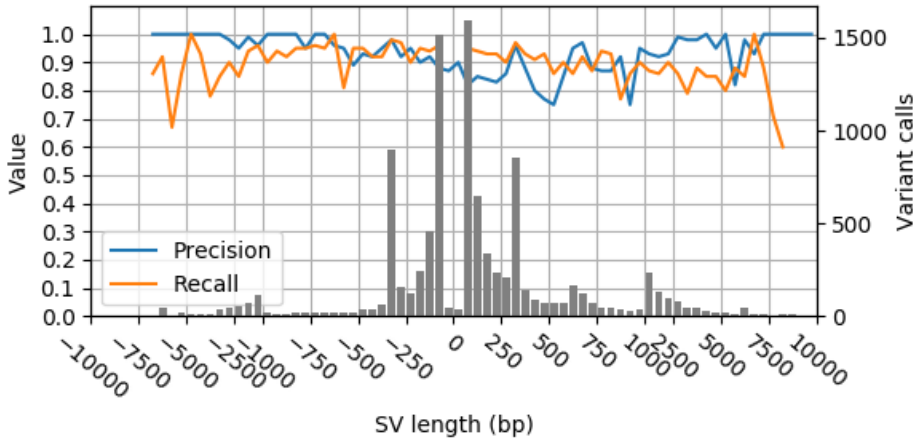
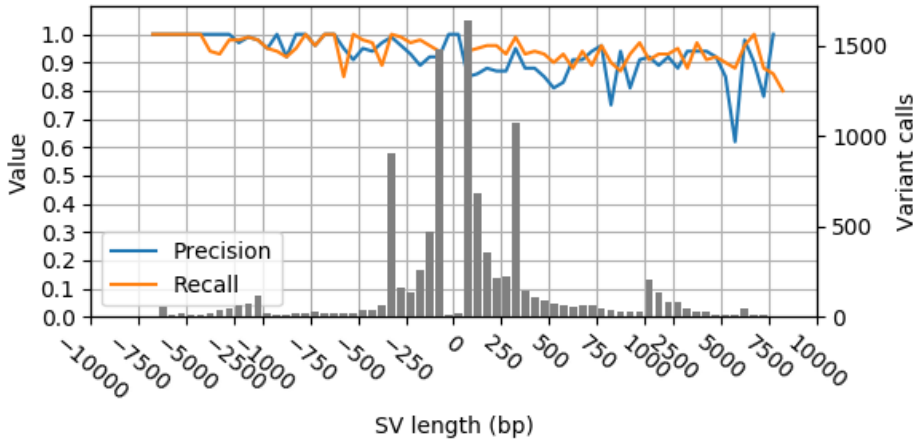


Figure B.7: **Number of SV calls from the 50.7x Oxford Nanopore dataset stratified into five size classes.** Shown is a stacked bar plot of SV classes represented by different colors. Each panel represents one size class and visualizes the number of calls in that size range called by *Sniffles* and *SVIM*. All callsets were generated with a confidence threshold of 5. The bottom right panel shows the counts for all SV calls regardless of size.

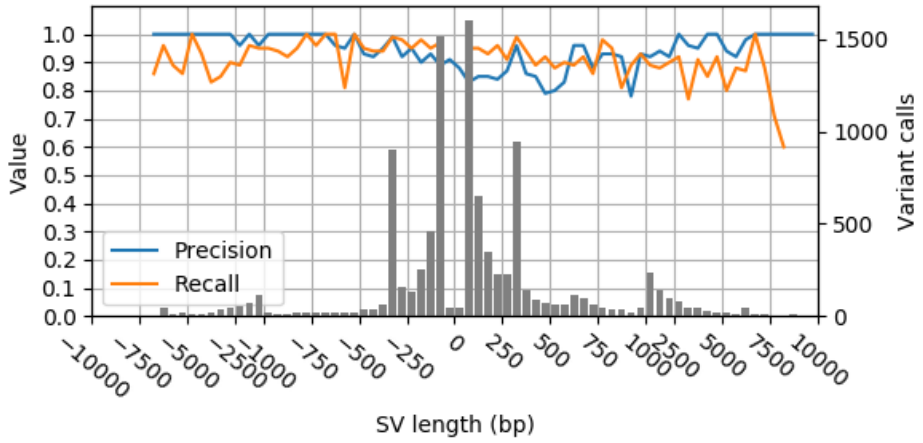


(a) **DipCall**

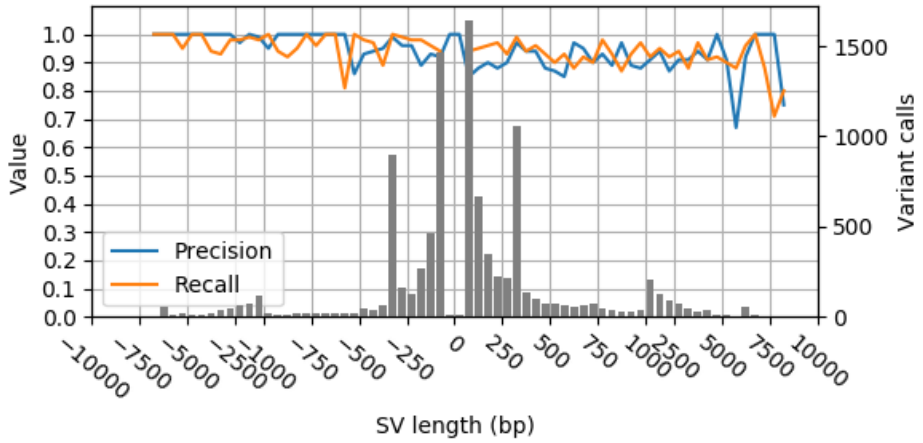


(b) **SVIM-asm**

Figure B.8: **SV detection performance across variant sizes for Assembly A.** Precision, recall and number of variant calls (y-axis) are shown for different SV length bins (x-axis) for the SV callers *DipCall* (a and b) and *SVIM-asm* (c and d). The bin size is 50 bp for variants shorter than 1 kbp and 500 bp for variants >1 kbp. Positive lengths indicate insertions and negative lengths indicate deletions.

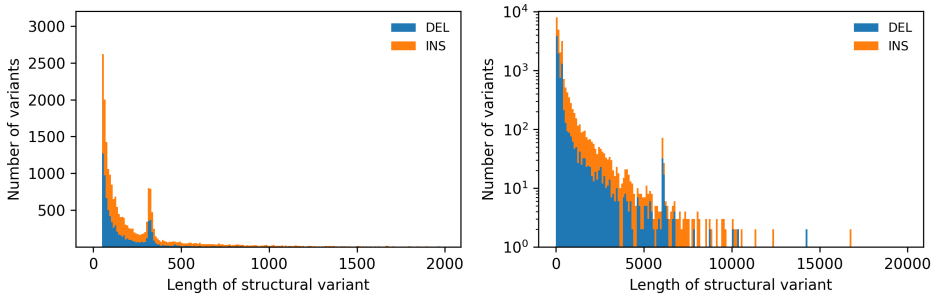


(a) **DipCall**

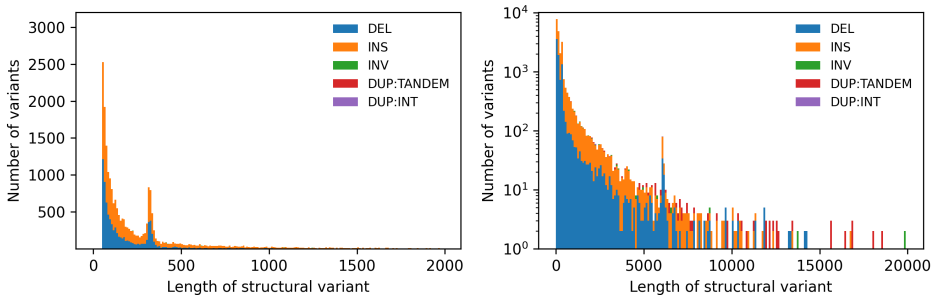


(b) **SVIM-asm**

Figure B.9: **SV detection performance across variant sizes for Assembly B.** Precision, recall and number of variant calls (y-axis) are shown for different SV length bins (x-axis) for the SV callers *DipCall* (a and b) and *SVIM-asm* (c and d). The bin size is 50 bp for variants shorter than 1 kbp and 500 bp for variants >1 kbp. Positive lengths indicate insertions and negative lengths indicate deletions.



(a) **DipCall**



(b) **SVIM-asm**

Figure B.10: **Size distribution of SVs identified in Assembly B.** Shown is a stacked histogram of SV classes represented by different colors. In the left panels, SV sizes up to 2 kbp are plotted with a bin size of 10 bp. In the right panel, SV sizes up to 20 kbp are plotted on a logarithmic y-axis with a bin size of 100 bp. The top and bottom panels, visualize callsets by *DipCall* and *SVIM-asm*, respectively.

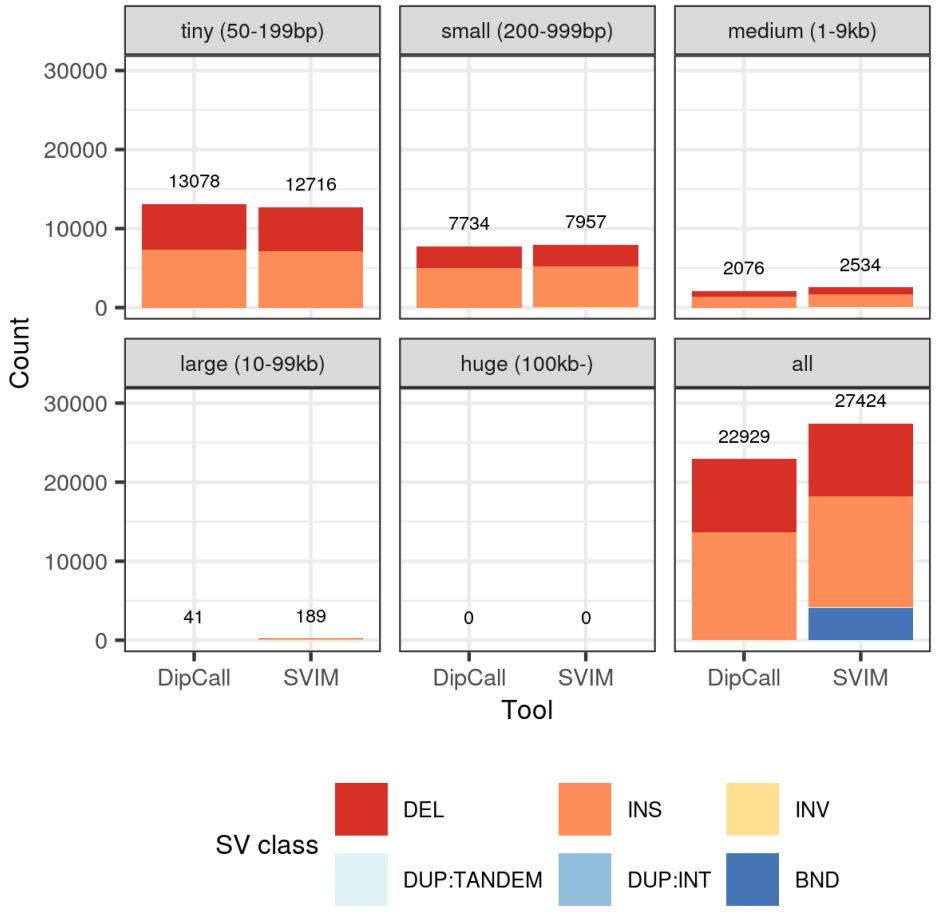


Figure B.11: **Number of SV calls from Assembly B grouped into five size classes.** Shown is a stacked bar plot of SV classes represented by different colors. Each panel represents one size class and visualizes the number of calls in that size range called by *DipCall* and *SVIM-asm*. The bottom right panel shows the counts for all SV calls regardless of size. As translocation breakpoints (BND) do not have a size, they are included only in this bottom right panel.

C

SUPPLEMENTARY TEXT

C.1 PARAMETERS AND THRESHOLDS OF SVIM

Like most software tools, our method *SVIM* needs to make certain decision during its execution that are dependent on parameters or thresholds. Some of these parameters, such as the minimum and maximum SV size, define the range of variants that *SVIM* is supposed to detect. Others, such as the partition and clustering thresholds, affect internal processing steps of the pipeline and have an impact on the quality of the results. All parameters can be modified by the user via command-line parameters. Table C.1 lists most of the parameters and thresholds used by *SVIM* with their respective default values. For a complete list, please refer to the command-line documentation of *SVIM* (`svim --help`).

Parameter	Default value
Minimum SV size	40 bp
Maximum SV size	100 kbp
Span-position normalization constant	900
Partition distance threshold	1 kbp
Clustering distance threshold	0.3
Homozygosity threshold	80%
Heterozygosity threshold	20%

Table C.1: **Parameters and thresholds of *SVIM*.** Shown are the most important parameters and thresholds of *SVIM* with their default values.

C.2 COMPLETE EVALUATION COMMANDS

In the following, we show the complete commands used to perform read alignment and SV calling for the evaluations in Chapter 5.

C.2.1 *Simulated long-read datasets*

We aligned the simulated reads with three different read aligners. For *SVIM* and *Sniffles*, reads were aligned using *minimap2* and *ngmlr*. For *pbsv*, reads were aligned using *pbmm2*.

```
#minimap2
minimap2 -ax map-pb -t 10 -z 200,100 --MD -Y <genome> <reads> >
    <alignments>

#pbmm2
pbmm2 index --num-threads 2 --preset SUBREAD <genome> <index>
pbmm2 align --preset SUBREAD -j 10 --sort <index> <reads>
    <alignments>

#ngmlr
ngmlr --presets pacbio -t 10 -r <genome> -q <reads> -o
    <alignments>
```

To perform SV calling on the read alignments, the following parameters were used.

```
#SVIM
svim alignment --interspersed_duplications_as_insertions
    <working_dir> <alignments> <genome>

#Sniffles
sniffles --mapped_reads <alignments> --min_support
    <confidence_threshold> --threads 3 --vcf <vcf>

#pbsv
pbsv discover <alignments> <sv_signatures>
pbsv call -j 2 --min-sv-length 40 --max-ins-length 100K
    --call-min-reads-one-sample <confidence_threshold>
    --call-min-reads-all-samples <confidence_threshold>
    --call-min-reads-per-strand-all-samples 0
    --call-min-bnd-reads-all-samples 0
    --call-min-read-perc-one-sample 0 <genome> <sv_signatures>
    <vcf>
```

C.2.2 *Real long-read datasets*

We aligned the real reads with *minimap2* (for *SVIM* and *Sniffles*) and *pbmm2* (for *pbsv*). For each input dataset, we used slightly different presets and parameters.

```
#minimap2 (CLR)
minimap2 -ax map-pb -t 10 --MD -Y <genome> <reads> >
    <alignments>

#minimap2 (CCS)
minimap2 -ax asm20 -k19 -w10 -O5,56 -E4,1 -A2 -B5 -z400,50
    -r2000 --lj-min-ratio 0.5 -g5000 -t 10 --MD -Y <genome>
    <reads> > <alignments>

#minimap2 (ONT)
minimap2 -ax map-ont -t 10 --MD -Y <genome> <reads> >
    <alignments>

#pbmm2 (CLR)
pbmm2 index --num-threads 2 --preset SUBREAD <genome> <index>
pbmm2 align --preset SUBREAD -j 10 --sort <index> <reads>
    <alignments>

#pbmm2 (CCS)
pbmm2 index --num-threads 2 --preset CCS <genome> <index>
pbmm2 align --preset CCS -j 10 --sort <index> <reads>
    <alignments>
```

To perform SV calling on the read alignments, the following parameters were used.

```
#SVIM
svim alignment --segment_gap_tolerance 20
    --segment_overlap_tolerance 20
    --interspersed_duplications_as_insertions
    --tandem_duplications_as_insertions --read_names
    --max.startv_size 1000000 <working_dir> <alignments>
    <genome>

#Sniffles
```

```
sniffles --mapped_reads <alignments> --min_length $40
    --min_support <confidence_threshold> --threads 1 --genotype
    --vcf <vcf>

#pbsv
pbsv discover <alignments> <sv_signatures>
pbsv call -t INS,DEL -j 1 --min-sv-length 40 --max-ins-length
    100K --call-min-reads-one-sample <confidence_threshold>
    --call-min-reads-all-samples <confidence_threshold>
    --call-min-reads-per-strand-all-samples 0
    --call-min-bnd-reads-all-samples 0
    --call-min-read-perc-one-sample 0 <genome> <sv_signatures>
    <vcf>
```

C.2.3 *Diploid genome assembly datasets*

We aligned the diploid assembly contigs with *minimap2* and called SVs using *SVIM-asm*. *DipCall* uses *minimap2*, too, but internally.

```
#minimap2
minimap2 -ax asm5 -r2k -t 8 <genome> <contigs_hap1> >
    <alignments_hap1>
minimap2 -ax asm5 -r2k -t 8 <genome> <contigs_hap2> >
    <alignments_hap2>

#SVIM-asm
svim-asm diploid <working_dir> <alignments_hap1>
    <alignments_hap2> <genome> --min_sv_size 20
    --tandem_duplications_as_insertions
    --interspersed_duplications_as_insertions
    --reference_gap_tolerance 1000
    --reference_overlap_tolerance 1000 --query_gap_tolerance
    2000 --query_overlap_tolerance 2000 --max_edit_distance 200
    --sample HG002 --query_names

#Dip-call
run-dipcall -t 10 -x hs37d5.PAR.bed HG002 <genome>
    <contigs_hap1> <contigs_hap2 > HG002.mak
make -j 40 -f HG002.mak
```

C.3 ASSEMBLY DATASETS

To evaluate the assembly-based SV callers, we analyzed two publicly available diploid genome assemblies of the HG002 individual from Wenger et al. (Assembly A) and Garg et al. (Assembly B) [30, 97]. The following Table C.2 contains information on both assemblies.

	Assembly A	Assembly B
Authors	Wenger et al.	Garg et al.
Assembler	Canu v1.7.1	DipAsm
Input data for contig assembly	29.7x PacBio CCS data (trio-binned)	29.7x PacBio CCS data
Input data for scaffolding	No scaffolding	28.5x Hi-C data
Polishing	Arrow v2.2.2	No polishing
Download	Maternal: https://downloads.pacbcloud.com/public/publications/2019-HG002-CCS/asm/HG002_canu_maternal.fasta Paternal: https://downloads.pacbcloud.com/public/publications/2019-HG002-CCS/asm/HG002_canu_paternal.fasta	Haplotype 1: ftp://ftp.dfci.harvard.edu/pub/hli/whdenovo/asm/NA24385-denovo-H1.fa.gz Haplotype 2: ftp://ftp.dfci.harvard.edu/pub/hli/whdenovo/asm/NA24385-denovo-H2.fa.gz
Citation	[97]	[30]

Table C.2: **Two recently generated diploid genome assemblies for the HG002 individual.**

D

ABSTRACT

Structural variants, commonly defined as genomic differences larger than 50 bp, are an important research target due to their large size and great impact on human phenotype and disease. Their unique properties and the weaknesses of traditional short-read sequencing technologies, however, complicate their detection and comprehensive characterization. Third-generation sequencing technologies, such as PacBio SMRT sequencing and ONT Nanopore sequencing, have the potential to resolve some of these problems through the generation of considerably longer reads. Despite their higher error rate and sequencing cost, they offer many advantages for the detection of structural variants and the complete reconstruction of personal genome sequences. Yet, available software tools for the detection of SVs from long reads and genome assemblies still do not fully exploit the possibilities.

Here we present two new computational methods, *SVIM* and *SVIM-asm*, for the detection and genotype estimation of structural variants using third-generation sequencing data. The methods can be applied to long, error-prone reads or genome assemblies and distinguish six canonical classes of structural variation. We apply both tools on simulated and real sequencing datasets and demonstrate that they outperform existing methods on the detection of genotyped SVs. In the context of a larger research project, we apply *SVIM* for the detection of both canonical SVs and long-range novel adjacencies in a set of highly rearranged genomes. After a stringent filtering process, the final callset of long-range novel adjacencies is validated with orthogonal Hi-C data. We show the completeness and precision of the callset demonstrating its suitability for downstream analyses, such as chromosome reconstruction.

E

ZUSAMMENFASSUNG

Vergleicht man die Genome verschiedener Lebewesen treten zahlreiche kleine und große Unterschiede zutage. Unterschiede mit einer Größe von mehr als 50 Basenpaaren werden auch Strukturvarianten genannt. Sie haben einen erheblichen Einfluss auf den Phänotyp des Menschen und seine Erkrankungen. Die Erkennung von Strukturvarianten wurde lange durch ihre besonderen Eigenschaften, aber auch Schwächen der gängigen Sequenziertechnologien erschwert. Neue Sequenziertechnologien der dritten Generation, z.B. PacBio SMRT Sequenzierung und ONT Nanopore Sequenzierung, sind nun in der Lage, einige dieser Probleme zu lösen. Sie produzieren Sequenzfragmente (Reads), die um ein vielfaches länger sind als Reads traditioneller Sequenziertechnologien, aber auch eine höhere Fehlerrate besitzen. Für die Erkennung von Strukturvarianten sowie die komplette Rekonstruktion von Genomsequenzen besitzen diese Technologien dennoch viele Vorteile. Bisher werden diese durch die bestehenden Software-Tools jedoch nicht ausgeschöpft.

Wir stellen zwei neue Softwaremethoden namens *SVIM* und *SVIM-asm* für die Erkennung und Genotypisierung von Strukturvarianten mittels Sequenzierdaten der dritten Generation vor. Die Anwendungen können sowohl für die Analyse langer Reads als auch kompletter Genomsequenzen eingesetzt werden und unterscheiden sechs klassische Typen von Strukturvarianten. Wir wenden beide Methoden auf simulierten und echten Sequenzierdaten an und zeigen, dass sie Strukturvarianten besser erkennen und genotypisieren können als bestehende Tools. Im Rahmen eines größeren Forschungsprojektes verwenden wir *SVIM*, um in einer Reihe von stark umstrukturierten Genomen sowohl klassische Strukturvarianten, als auch neue Verbindungen zwischen weit entfernten Genompositionen zu detektieren. Die Neuverbindungen werden nach verschiedenen Qualitätsmerkmalen gefiltert und anschließend mit unabhängigen Hi-C Daten validiert. Unser Ansatz bildet damit die Voraussetzung für nachfolgende Analysen, z.B. der Genregulation in umstrukturierten Genomen.

BIBLIOGRAPHY

[1] 1000 Genomes Project Consortium. "A global reference for human genetic variation." In: *Nature* 526.7571 (2015), pp. 68–74.

[2] Can Alkan, Bradley P Coe, and Evan E Eichler. "Genome structural variation discovery and genotyping." In: *Nat. Rev. Genet.* 12.5 (2011), pp. 363–376.

[3] Peter A Audano, Arvis Sulovari, Tina A Graves-Lindsay, Stuart Cantsilieris, Melanie Sorensen, AnneMarie E Welch, Max L Dougherty, Bradley J Nelson, Ankeeta Shah, Susan K Dutcher, et al. "Characterizing the major structural variant alleles of the human genome." In: *Cell* 176.3 (2019), pp. 663–675.

[4] Christoph Bartenhagen and Martin Dugas. "RSVSim: an R/Bioconductor package for the simulation of structural variations." In: *Bioinformatics* 29.13 (2013), pp. 1679–1681.

[5] Michael F Berger, Michael S Lawrence, Francesca Demichelis, Yotam Drier, Kristian Cibulskis, Andrey Y Sivachenko, Andrea Sboner, Raquel Esgueva, Dorothee Pflueger, Carrie Sougnez, et al. "The genomic complexity of primary human prostate cancer." In: *Nature* 470.7333 (2011), pp. 214–220.

[6] Pacific Biosciences. *PacBio CLR reads of HG002, Ashkenazim Son*. https://go.aws/2xBsACi.

[7] Pacific Biosciences. *PacBio HiFi reads of HG002, Ashkenazim Son*. https://go.aws/2X7sSvF.

[8] Pacific Biosciences. *pbsv*. 2020. URL: https://github.com/PacificBiosciences/pbsv.

[9] Hongzhi Cao, Alex R Hastie, Dandan Cao, Ernest T Lam, Yuhui Sun, Haodong Huang, Xiao Liu, Liya Lin, Warren Andrews, Saki Chan, et al. "Rapid detection of structural variation in a human genome using nanochannel-based genome mapping technology." In: *GigaScience* 3.1 (2014), pp. 2047–217X.

[10] Claudia MB Carvalho and James R Lupski. "Mechanisms underlying structural variant formation in genomic disorders." In: *Nat. Rev. Genet.* 17.4 (2016), pp. 224–238.

[11] Mark J Chaisson and Glenn Tesler. "Mapping single molecule sequencing reads using basic local alignment with successive refinement (BLASR): application and theory." In: *BMC Bioinformatics* 13.1 (2012), p. 238.

[12] Mark JP Chaisson, John Huddleston, Megan Y Dennis, Peter H Sudmant, Maika Malig, Fereydoun Hormozdiari, Francesca Antonacci, Urvashi Surti, Richard Sandstrom, Matthew Boitano, et al. "Resolving the complexity of the human genome using single-molecule sequencing." In: *Nature* 517.7536 (2015), pp. 608–611.

[13] Mark JP Chaisson, Ashley D Sanders, Xuefang Zhao, Ankit Malhotra, David Porubsky, Tobias Rausch, Eugene J Gardner, Oscar L Rodriguez, Li Guo, Ryan L Collins, et al. "Multi-platform discovery of haplotype-resolved structural variation in human genomes." In: *Nat. Commun.* 10.1 (2019), pp. 1–16.

[14] Chen-Shan Chin, Paul Peluso, Fritz J Sedlazeck, Maria Nattestad, Gregory T Concepcion, Alicia Clum, Christopher Dunn, Ronan O'Malley, Rosa Figueroa-Balderas, Abraham Morales-Cruz, et al. "Phased diploid genome assembly with single-molecule real-time sequencing." In: *Nat. Methods* 13.12 (2016), pp. 1050–1054.

[15] James Clarke, Hai-Chen Wu, Lakmal Jayasinghe, Alpesh Patel, Stuart Reid, and Hagan Bayley. "Continuous base identification for single-molecule nanopore DNA sequencing." In: *Nat. Nanotechnol.* 4.4 (2009), p. 265.

[16] Ryan L Collins, Harrison Brand, Claire E Redin, Carrie Hanscom, Caroline Antolik, Matthew R Stone, Joseph T Glessner, Tamara Mason, Giulia Pregno, Naghmeh Dorrani, et al. "Defining the diverse spectrum of inversions, complex structural variation, and chromothripsis in the morbid human genome." In: *Genome Biol.* 18.1 (2017), pp. 1–21.

[17] 1000 Genomes Project Consortium et al. "A map of human genome variation from population-scale sequencing." In: *Nature* 467.7319 (2010), p. 1061.

[18] Petr Danecek, Adam Auton, Goncalo Abecasis, Cornelis A Albers, Eric Banks, Mark A DePristo, Robert E Handsaker, Gerton Lunter, Gabor T Marth, Stephen T Sherry, et al. "The variant call format and VCFtools." In: *Bioinformatics* 27.15 (2011), pp. 2156–2158.

[19] Wouter De Coster and Christine Van Broeckhoven. "Newest methods for detecting structural variations." In: *Trends Biotechnol.* 37.9 (2019), pp. 973–982.

[20] Wouter De Coster, Peter De Rijk, Arne De Roeck, Tim De Pooter, Svenn D'Hert, Mojca Strazisar, Kristel Sleegers, and Christine Van Broeckhoven. "Structural variants identified by Oxford Nanopore PromethION sequencing of the human genome." In: *Genome Res.* 29.7 (2019), pp. 1178–1187.

[21] Mark A DePristo, Eric Banks, Ryan Poplin, Kiran V Garimella, Jared R Maguire, Christopher Hartl, Anthony A Philippakis, Guillermo Del Angel, Manuel A Rivas, Matt Hanna, et al. "A framework for variation discovery and genotyping using next-generation DNA sequencing data." In: *Nature Genet.* 43.5 (2011), p. 491.

[22] Jacob F Degner, John C Marioni, Athma A Pai, Joseph K Pickrell, Everlyne Nkadori, Yoav Gilad, and Jonathan K Pritchard. "Effect of read-mapping biases on detecting allele-specific expression from RNA-sequencing data." In: *Bioinformatics* 25.24 (2009), pp. 3207–3212.

[23] Jesse R Dixon, Siddarth Selvaraj, Feng Yue, Audrey Kim, Yan Li, Yin Shen, Ming Hu, Jun S Liu, and Bing Ren. "Topological domains in mammalian genomes identified by analysis of chromatin interactions." In: *Nature* 485.7398 (2012), pp. 376–380.

[24] Evan E Eichler, Royden A Clark, and Xinwei She. "An assessment of the sequence gaps: unfinished business in a finished human genome." In: *Nat. Rev. Genet.* 5.5 (2004), pp. 345–354.

[25] John Eid, Adrian Fehr, Jeremy Gray, Khai Luong, John Lyle, Geoff Otto, Paul Peluso, David Rank, Primo Baybayan, Brad Bettman, et al. "Real-time DNA sequencing from single polymerase molecules." In: *Science* 323.5910 (2009), pp. 133–138.

[26] Adam C English, William J Salerno, and Jeffrey G Reid. "PBHoney: identifying genomic variants via long-read discordance and interrupted mapping." In: *BMC Bioinformatics* 15.1 (2014), p. 180.

[27] Lars Feuk, Andrew R Carson, and Stephen W Scherer. "Structural variation in the human genome." In: *Nat. Rev. Genet.* 7.2 (2006), pp. 85–97.

[28] Josep V Forment, Abderrahmane Kaidi, and Stephen P Jackson. "Chromothripsis and cancer: causes and consequences of chromosome shattering." In: *Nat. Rev. Cancer* 12.10 (2012), pp. 663–670.

[29] Kelly A Frazer, Sarah S Murray, Nicholas J Schork, and Eric J Topol. "Human genetic variation and its contribution to complex traits." In: *Nat. Rev. Genet.* 10.4 (2009), pp. 241–251.

[30] Shilpa Garg, Arkarachai Fungtammasan, Andrew Carroll, Mike Chou, Anthony Schmitt, Xiang Zhou, Stephen Mac, Paul Peluso, Emily Hatas, Jay Ghurye, et al. "Efficient chromosome-scale haplotype-resolved assembly of human genomes." In: *Nat. Biotechnol.* (in press).

[31] Erik Garrison, Jouni Sirén, Adam M Novak, Glenn Hickey, Jordan M Eizenga, Eric T Dawson, William Jones, Shilpa Garg, Charles Markello, Michael F Lin, et al. "Variation graph toolkit improves read mapping by representing genetic variation in the reference." In: *Nat. Biotechnol.* 36.9 (2018), pp. 875–879.

[32] Spiral Genetics. *truvari*. 2019. URL: https://github.com/spiralgenetics/truvari.

[33] Travis C Glenn. "Field guide to next-generation DNA sequencers." In: *Mol. Ecol. Resour.* 11.5 (2011), pp. 759–769.

[34] Manish Goel, Hequan Sun, Wen-Biao Jiao, and Korbinian Schneeberger. "SyRI: finding genomic rearrangements and local sequence differences from whole-genome assemblies." In: *Genome Biol.* 20.1 (2019), pp. 1–13.

[35] Manuel L Gonzalez-Garay. "The road from next-generation sequencing to personalized medicine." In: *Pers. Med.* 11.5 (2014), pp. 523–544.

[36] Sara Goodwin, John D McPherson, and W Richard McCombie. "Coming of age: ten years of next-generation sequencing technologies." In: *Nat. Rev. Genet.* 17.6 (2016), p. 333.

[37] K Chidananda Gowda and Edwin Diday. "Symbolic clustering using a new dissimilarity measure." In: *Pattern Recogn.* 24.6 (1991), pp. 567–578.

[38] Björn Grüning, Ryan Dale, Andreas Sjödin, Brad A Chapman, Jillian Rowe, Christopher H Tomkins-Tinch, Renan Valieris, and Johannes Köster. "Bioconda: sustainable and comprehensive software distribution for the life sciences." In: *Nat. Methods* 15.7 (2018), pp. 475–476.

[39] Olivier Harismendy, Pauline C Ng, Robert L Strausberg, Xiaoyun Wang, Timothy B Stockwell, Karen Y Beeson, Nicholas J Schork, Sarah S Murray, Eric J Topol, Samuel Levy, et al. "Evaluation of next generation sequencing platforms for population targeted sequencing studies." In: *Genome Biol.* 10.3 (2009), R32.

[40] Philip J Hastings, James R Lupski, Susan M Rosenberg, and Grzegorz Ira. "Mechanisms of change in gene copy number." In: *Nat. Rev. Genet.* 10.8 (2009), pp. 551–564.

[41] Cheng R L Huang, Kathleen H Burns, and Jef D Boeke. "Active transposition in genomes." In: *Annu. Rev. Genet.* 46 (2012), pp. 651–675.

[42] John Huddleston and Evan E Eichler. "An incomplete understanding of human genetic variation." In: *Genetics* 202.4 (2016), pp. 1251–1254.

[43] John Huddleston, Mark JP Chaisson, Karyn Meltz Steinberg, Wes Warren, Kendra Hoekzema, David Gordon, Tina A Graves-Lindsay, Katherine M Munson, Zev N Kronenberg, Laura Vives, et al. "Discovery and genotyping of structural variation from long-read haploid genome sequence data." In: *Genome Res.* 27.5 (2017), pp. 677–685.

[44] Miten Jain, Sergey Koren, Karen H Miga, Josh Quick, Arthur C Rand, Thomas A Sasani, John R Tyson, Andrew D Beggs, Alexander T Dilthey, Ian T Fiddes, et al. "Nanopore sequencing and assembly of a human genome with ultra-long reads." In: *Nat. Biotechnol.* 36.4 (2018), p. 338.

[45] Tao Jiang, Bo Liu, Junyi Li, and Yadong Wang. "rMETL: sensitive mobile element insertion detection with long read realignment." In: *Bioinformatics* 35.18 (2019), pp. 3484–3486.

[46] Yinping Jiao, Paul Peluso, Jinghua Shi, Tiffany Liang, Michelle C Stitzer, Bo Wang, Michael S Campbell, Joshua C Stein, Xuehong Wei, Chen-Shan Chin, et al. "Improved maize reference genome with single-molecule technologies." In: *Nature* 546.7659 (2017), pp. 524–527.

[47] Wigard P Kloosterman, Victor Guryev, Mark van Roosmalen, Karen J Duran, Ewart de Bruijn, Saskia CM Bakker, Tom Letteboer, Bernadette van Nesselrooij, Ron Hochstenbach, Martin Poot, et al. "Chromothripsis as a mechanism driving complex de novo structural rearrangements in the germline." In: *Hum. Mol. Genet.* 20.10 (2011), pp. 1916–1924.

[48] Sergey Koren, Arang Rhie, Brian P Walenz, Alexander T Dilthey, Derek M Bickhart, Sarah B Kingan, Stefan Hiendleder, John L Williams, Timothy PL Smith, and Adam M Phillippy. "De novo assembly of haplotype-resolved genomes with trio binning." In: *Nat. Biotechnol.* 36.12 (2018), pp. 1174–1182.

[49] Shunichi Kosugi, Yukihide Momozawa, Xiaoxi Liu, Chikashi Terao, Michiaki Kubo, and Yoichiro Kamatani. "Comprehensive evaluation of structural variation detection algorithms for whole genome sequencing." In: *Genome Biol.* 20.1 (2019), p. 117.

[50] Eliyahu Kraus, Wai-Ying Leung, and James E Haber. "Break-induced replication: a review and an example in budding yeast." In: *P. Natl. Acad. Sci. USA* 98.15 (2001), pp. 8255–8262.

[51] Zev N Kronenberg, Ian T Fiddes, David Gordon, Shwetha Murali, Stuart Cantsilieris, Olivia S Meyerson, Jason G Underwood, Bradley J Nelson, Mark JP Chaisson, Max L Dougherty, et al. "High-resolution comparative analysis of great ape genomes." In: *Science* 360.6393 (2018).

[52] Zev N Kronenberg, Arang Rhie, Sergey Koren, Gregory Concepcion, Paul Peluso, Katherine Munson, Stefan Hiendleder, Olivier Fedrigo, Erich Jarvis, Adam Phillippy, et al. "Extended haplotype phasing of de novo genome assemblies with FALCON-Phase." In: *bioRxiv* (2019), p. 327064.

[53] Ernest T Lam, Alex Hastie, Chin Lin, Dean Ehrlich, Somes K Das, Michael D Austin, Paru Deshpande, Han Cao, Niranjan Nagarajan, Ming Xiao, et al. "Genome mapping on nanochannel arrays for structural variation analysis and sequence assembly." In: *Nat. Biotechnol.* 30.8 (2012), pp. 771–776.

[54] ES Lander, LM Linton, B Birren, C Nusbaum, MC Zody, J Baldwin, K Devon, K Dewar, M Doyle, W FitzHugh, et al. "Initial sequencing and analysis of the human genome." In: *Nature* 409.6822 (2001), pp. 860–921.

[55] Heng Li. "Toward better understanding of artifacts in variant calling from high-coverage samples." In: *Bioinformatics* 30.20 (2014), pp. 2843–2851.

[56] Heng Li. "Minimap2: pairwise alignment for nucleotide sequences." In: *Bioinformatics* 1 (2018), p. 7.

[57] Heng Li, Bob Handsaker, Alec Wysoker, Tim Fennell, Jue Ruan, Nils Homer, Gabor Marth, Goncalo Abecasis, and Richard Durbin. "The sequence alignment/map format and SAMtools." In: *Bioinformatics* 25.16 (2009), pp. 2078–2079.

[58] Heng Li, Jonathan M Bloom, Yossi Farjoun, Mark Fleharty, Laura Gauthier, Benjamin Neale, and Daniel MacArthur. "A synthetic-diploid benchmark for accurate variant-calling evaluation." In: *Nat. Methods* 15.8 (2018), pp. 595–597.

[59] Yilong Li, Nicola D Roberts, Jeremiah A Wala, Ofer Shapira, Steven E Schumacher, Kiran Kumar, Ekta Khurana, Sebastian Waszak, Jan O Korbel, James E Haber, et al. "Patterns of somatic structural variation in human cancer genomes." In: *Nature* 578.7793 (2020), pp. 112–121.

[60] Erez Lieberman-Aiden, Nynke L Van Berkum, Louise Williams, Maxim Imakaev, Tobias Ragoczy, Agnes Telling, Ido Amit, Bryan R Lajoie, Peter J Sabo, Michael O Dorschner, et al. "Comprehensive mapping of long-range interactions reveals folding principles of the human genome." In: *Science* 326.5950 (2009), pp. 289–293.

[61] Pengfei Liu, Ayelet Erez, Sandesh C Sreenath Nagamani, Shweta U Dhar, Katarzyna E Kołodziejska, Avinash V Dharmadhikari, M Lance Cooper, Joanna Wiszniewska, Feng Zhang, Marjorie A Withers, et al. "Chromosome catastrophes involve replication mechanisms generating complex genomic rearrangements." In: *Cell* 146.6 (2011), pp. 889–903.

[62] Siyang Liu, Shujia Huang, Junhua Rao, Weijian Ye, Genome Denmark Consortium, Anders Krogh, and Jun Wang. "Discovery, genotyping and characterization of structural variation and novel sequence at single nucleotide resolution from de novo genome assemblies on a population scale." In: *GigaScience* 4.1 (2015), s13742–015.

[63] Yueyuan Liu, Mingyue Zhang, Jieying Sun, Wenjing Chang, Manyi Sun, Shaoling Zhang, and Jun Wu. "Comparison of multiple algorithms to reliably detect structural variants in pears." In: *BMC Genomics* 21.1 (2020), pp. 1–15.

[64] Kenneth J Locey and Jay T Lennon. "Scaling laws predict global microbial diversity." In: *P. Natl. Acad. Sci. USA* 113.21 (2016), pp. 5970–5975.

[65] Glennis A Logsdon, Mitchell R Vollger, and Evan E Eichler. "Long-read human genome sequencing and its applications." In: *Nat. Rev. Genet.* (2020), pp. 1–18.

[66] Erick W Loomis, John S Eid, Paul Peluso, Jun Yin, Luke Hickey, David Rank, Sarah McCalmon, Randi J Hagerman, Flora Tassone, and Paul J Hagerman. "Sequencing the un-

sequenceable: expanded CGG-repeat alleles of the fragile X gene." In: *Genome Res.* 23.1 (2013), pp. 121–128.

[67] Christopher A Maher and Richard K Wilson. "Chromothripsis and human disease: piecing together the shattering process." In: *Cell* 148.1-2 (2012), pp. 29–32.

[68] Jason D Merker, Aaron M Wenger, Tam Sneddon, Megan Grove, Zachary Zappala, Laure Fresard, Daryl Waggott, Sowmi Utiramerur, Yanli Hou, Kevin S Smith, et al. "Long-read genome sequencing identifies causal structural variation in a Mendelian disease." In: *Genet. Med.* 20.1 (2018), pp. 159–163.

[69] André E Minoche, Juliane C Dohm, and Heinz Himmelbauer. "Evaluation of genomic high-throughput sequencing data generated on Illumina HiSeq and genome analyzer systems." In: *Genome Biol.* 12.11 (2011), R112.

[70] Camilo Mora, Derek P Tittensor, Sina Adl, Alastair GB Simpson, and Boris Worm. "How many species are there on Earth and in the ocean?" In: *PLoS Biol.* 9.8 (2011), e1001127.

[71] Maria Nattestad and Michael C Schatz. "Assemblytics: a web analytics tool for the detection of variants from an assembly." In: *Bioinformatics* 32.19 (2016), pp. 3021–3023.

[72] Rasmus Nielsen, Joshua S Paul, Anders Albrechtsen, and Yun S Song. "Genotype and SNP calling from next-generation sequencing data." In: *Nat. Rev. Genet.* 12.6 (2011), pp. 443–451.

[73] Stephan Pabinger, Andreas Dander, Maria Fischer, Rene Snajder, Michael Sperk, Mirjana Efremova, Birgit Krabichler, Michael R Speicher, Johannes Zschocke, and Zlatko Trajanoski. "A survey of tools for variant analysis of next-generation genome sequencing data." In: *Brief. Bioinform.* 15.2 (2014), pp. 256–278.

[74] Aurèle Piazza and Wolf-Dietrich Heyer. "Homologous recombination and the formation of complex genomic rearrangements." In: *Trends Cell Biol.* 29.2 (2019), pp. 135–149.

[75] David Porubsky, Peter Ebert, Peter A Audano, Mitchell R Vollger, William T Harvey, Katherine M Munson, Melanie Sorensen, Arvis Sulovari, Marina Haukness, Maryam Ghareghani, et al. "A fully phased accurate assembly of an individual human genome." In: *bioRxiv* (2019), p. 855049.

[76] Marta Puig, Sonia Casillas, Sergi Villatoro, and Mario Cáceres. "Human inversions and their functional consequences." In: *Brief. Funct. Genomics* 14.5 (2015), pp. 369–379.

[77] Richard Redon, Shumpei Ishikawa, Karen R Fitch, Lars Feuk, George H Perry, T Daniel Andrews, Heike Fiegler, Michael H Shapero, Andrew R Carson, Wenwei Chen, et al. "Global variation in copy number in the human genome." In: *Nature* 444.7118 (2006), pp. 444–454.

[78] Martin G Reese, Barry Moore, Colin Batchelor, Fidel Salas, Fiona Cunningham, Gabor T Marth, Lincoln Stein, Paul Flicek, Mark Yandell, and Karen Eilbeck. "A standard variation file format for human genome sequences." In: *Genome Biol.* 11.8 (2010), R88.

[79] University of California in Santa Cruz. *Oxford Nanopore reads of HG002, Ashkenazim Son*. https://go.aws/2xBsACi.

[80] David C Schwartz, Xiaojun Li, Luis I Hernandez, Satyadarshan P Ramnarain, Edward J Huff, and Yu-Ker Wang. "Ordered restriction maps of Saccharomyces cerevisiae chromosomes constructed by optical mapping." In: *Science* 262.5130 (1993), pp. 110–114.

[81] Jonathan Sebat, B Lakshmi, Dheeraj Malhotra, Jennifer Troge, Christa Lese-Martin, Tom Walsh, Boris Yamrom, Seungtai Yoon, Alex Krasnitz, Jude Kendall, et al. "Strong association of de novo copy number mutations with autism." In: *Science* 316.5823 (2007), pp. 445–449.

[82] Fritz J Sedlazeck, Philipp Rescheneder, Moritz Smolka, Han Fang, Maria Nattestad, Arndt Von Haeseler, and Michael C Schatz. "Accurate detection of complex structural variations using single-molecule sequencing." In: *Nat. Methods* 15.6 (2018), pp. 461–468.

[83] Kishwar Shafin, Trevor Pesout, Ryan Lorig-Roach, Marina Haukness, Hugh E Olsen, Colleen Bosworth, Joel Armstrong, Kristof Tigyi, Nicholas Maurer, Sergey Koren, et al. "Nanopore sequencing and the Shasta toolkit enable efficient de novo assembly of eleven human genomes." In: *Nat. Biotechnol.* (2020), pp. 1–10.

[84] Haojing Shao, Devika Ganesamoorthy, Tania Duarte, Minh Duc Cao, Clive J Hoggart, and Lachlan JM Coin. "npInv: accurate detection and genotyping of inversions using long read sub-alignment." In: *BMC Bioinformatics* 19.1 (2018), p. 261.

[85] Andrew J Sharp, Devin P Locke, Sean D McGrath, Ze Cheng, Jeffrey A Bailey, Rhea U Vallente, Lisa M Pertz, Royden A Clark, Stuart Schwartz, Rick Segraves, et al. "Segmental duplications and copy-number variation in the human genome." In: *Am. J. Hum. Genet.* 77.1 (2005), pp. 78–88.

[86] Martin Šošić and Mile Šikić. "Edlib: a C/C++ library for fast, exact sequence alignment using edit distance." In: *Bioinformatics* 33.9 (2017), pp. 1394–1395.

[87] Malte Spielmann, Darío G Lupiáñez, and Stefan Mundlos. "Structural variation in the 3D genome." In: *Nat. Rev. Genet.* 19.7 (2018), pp. 453–467.

[88] Mircea Cretu Stancu, Markus J Van Roosmalen, Ivo Renkens, Marleen M Nieboer, Sjors Middelkamp, Joep De Ligt, Giulia Pregno, Daniela Giachino, Giorgia Mandrile, Jose Espejo Valle-Inclan, et al. "Mapping and phasing of structural variation in patient genomes using nanopore sequencing." In: *Nat. Commun.* 8.1 (2017), pp. 1–13.

[89] Philip J Stephens, Chris D Greenman, Beiyuan Fu, Fengtang Yang, Graham R Bignell, Laura J Mudie, Erin D Pleasance, King Wai Lau, David Beare, Lucy A Stebbings, et al. "Massive genomic rearrangement acquired in a single catastrophic event during cancer development." In: *Cell* 144.1 (2011), pp. 27–40.

[90] Chip Stewart, Deniz Kural, Michael P Strömberg, Jerilyn A Walker, Miriam K Konkel, Adrian M Stütz, Alexander E Urban, Fabian Grubert, Hugo YK Lam, Wan-Ping Lee, et al. "A

comprehensive map of mobile element insertion polymorphisms in humans." In: *PLoS Genet.* 7.8 (2011), e1002236.

[91] Bianca K Stöcker, Johannes Köster, and Sven Rahmann. "SimLoRD: Simulation of Long Read Data." In: *Bioinformatics* 32.17 (2016), pp. 2704–2706.

[92] Peter H Sudmant, Tobias Rausch, Eugene J Gardner, Robert E Handsaker, Alexej Abyzov, John Huddleston, Yan Zhang, Kai Ye, Goo Jun, Markus Hsi-Yang Fritz, et al. "An integrated map of structural variation in 2,504 human genomes." In: *Nature* 526.7571 (2015), pp. 75–81.

[93] Cheng Yong Tham, Roberto Tirado-Magallanes, Yufen Goh, Melissa J Fullwood, Bryan TH Koh, Wilson Wang, Chin Hin Ng, Wee Joo Chng, Alexandre Thiery, Daniel G Tenen, et al. "NanoVar: accurate characterization of patients' genomic structural variants using low-depth nanopore sequencing." In: *Genome Biol.* 21.1 (2020), pp. 1–15.

[94] Ayelet Voskoboynik, Norma F Neff, Debashis Sahoo, Aaron M Newman, Dmitry Pushkarev, Winston Koh, Benedetto Passarelli, H Christina Fan, Gary L Mantalas, Karla J Palmeri, et al. "The genome sequence of the colonial chordate, Botryllus schlosseri." In: *eLife* 2 (2013), e00569.

[95] Jason L Weirather, Mariateresa de Cesare, Yunhao Wang, Paolo Piazza, Vittorio Sebastiano, Xiu-Jie Wang, David Buck, and Kin Fai Au. "Comprehensive comparison of Pacific Biosciences and Oxford Nanopore Technologies and their applications to transcriptome analysis." In: *F1000Research* 6 (2017).

[96] Joachim Weischenfeldt, Orsolya Symmons, Francois Spitz, and Jan O Korbel. "Phenotypic impact of genomic structural variation: insights from and for human disease." In: *Nat. Rev. Genet.* 14.2 (2013), pp. 125–138.

[97] Aaron M Wenger, Paul Peluso, William J Rowell, Pi-Chuan Chang, Richard J Hall, Gregory T Concepcion, Jana Ebler, Arkarachai Fungtammasan, Alexey Kolesnikov, Nathan D Olson, et al. "Accurate circular consensus long-read sequencing improves variant detection and assembly of a human genome." In: *Nat. Biotechnol.* 37.10 (2019), pp. 1155–1162.

[98] Thomas Willems, Melissa Gymrek, Gareth Highnam, David Mittelman, Yaniv Erlich, 1000 Genomes Project Consortium, et al. "The landscape of human STR variation." In: *Genome Res.* 24.11 (2014), pp. 1894–1904.

[99] Elzo de Wit and Wouter De Laat. "A decade of 3C technologies: insights into nuclear organization." In: *Gene Dev.* 26.1 (2012), pp. 11–24.

[100] Cinthya J Zepeda-Mendoza and Cynthia C Morton. "The iceberg under water: unexplored complexity of Chromoanagenesis in congenital disorders." In: *Am. J. Hum. Genet.* 104.4 (2019), pp. 565–577.

[101] Justin M Zook, David Catoe, Jennifer McDaniel, Lindsay Vang, Noah Spies, Arend Sidow, Ziming Weng, Yuling Liu, Christopher E Mason, Noah Alexander, et al. "Extensive sequencing of seven human genomes to characterize benchmark reference materials." In: *Scientific data* 3.1 (2016), pp. 1–26.

[102] Justin M Zook, Nancy F Hansen, Nathan D Olson, Lesley Chapman, James C Mullikin, Chunlin Xiao, Stephen Sherry, Sergey Koren, Adam M Phillippy, Paul C Boutros, et al. "A robust benchmark for detection of germline large deletions and insertions." In: *Nat. Biotechnol.* (2020), pp. 1–9.

SELBSTSTÄNDIGKEITSERKLÄRUNG

Name: David Heller

Ich erkläre gegenüber der Freien Universität Berlin, dass ich die vorliegende Dissertation selbstständig und ohne Benutzung anderer als der angegebenen Quellen und Hilfsmittel angefertigt habe. Die vorliegende Arbeit ist frei von Plagiaten. Alle Ausführungen, die wörtlich oder inhaltlich aus anderen Schriften entnommen sind, habe ich als solche kenntlich gemacht. Diese Dissertation wurde in gleicher oder ähnlicher Form noch in keinem früheren Promotionsverfahren eingereicht.

Mit einer Prüfung meiner Arbeit durch ein Plagiatsprüfungsprogramm erkläre ich mich einverstanden.

Berlin, September 2020

David Heller