

Universiteit
Leiden
The Netherlands

Algorithms for structural variant detection

Lin, J.

Citation
Lin, J. (2022, June 24). *Algorithms for structural variant detection*. Retrieved from https://hdl.handle.net/1887/3391016

Version:	Publisher's Version
License:	Licence agreement concerning inclusion of doctoral thesis in the Institutional Repository of the University of Leiden
Downloaded from:	https://hdl.handle.net/1887/3391016

Note: To cite this publication please use the final published version (if applicable).

Algorithms for Structural Variant Detection

Proefschrift

ter verkrijging van
de graad van doctor aan de Universiteit Leiden
op gezag van de rector magnificus prof.dr.ir. H. Bijl,
volgens besluit van het college van promoties
te verdedigen op vrijdag 24 juni 2022
klokke 10.00 uur

door

Jiadong Lin

geboren te Xi'an, China

in 1991

Promotiecommissie

Promotores:	Dr. Walter A. Kosters	
	Prof.dr. Kai Ye	
Overige leden:	Prof.dr. Thomas H.W. Bäck	
	Dr. Lu Cao	
	Dr. Hailiang Mei	Leiden University Medical Center
	Prof.dr. Yianyong Sun	Xi'an Jiaotong University
	Prof.dr.ir. Fons J. Verbeek	
	Prof.dr. Gerard P. van Westen	

Copyright © 2022 Jiadong Lin All rights reserved

ISBN 978-94-6421-778-0

Het onderzoek beschreven in dit proefschrift is uitgevoerd aan het Leiden Institute of Advanced Computer Science (LIACS, Universiteit Leiden) en aan Xi'an Jiaotong University.
The research is financially supported by the Chinese Scholarship Council (CSC No. 201906280462).

Contents

1 Introduction and background **1**
- 1.1 Computational genomics . . . 1
- 1.2 Emerging DNA sequencing technologies . . . 2
- 1.3 Genome structural variations are important . . . 3
- 1.4 Detecting structural variation . . . 5
- 1.5 Pairwise sequence alignment for nucleotide sequences . . . 7
- 1.6 Usage of graphs for structural variants detection and analysis 10
- 1.7 Frequent subgraph mining . . . 13
- 1.8 Objective and outline of the thesis . . . 13
- 1.9 List of abbreviations . . . 16

2 Mako **19**
- 2.1 Introduction . . . 20
- 2.2 Materials and methods . . . 22
  - 2.2.1 Overview of Mako . . . 22
  - 2.2.2 Building signal graph . . . 24
  - 2.2.3 Detecting CSVs with pattern growth . . . 24
  - 2.2.4 Performance evaluation . . . 26
  - 2.2.5 Preparing CSV benchmarks for performance evaluation 27
  - 2.2.6 Orthogonal validation of Mako detected CSVs . . . 28
  - 2.2.7 Data availability . . . 28
- 2.3 Results . . . 29
  - 2.3.1 Mako effectively characterizes multiple breakpoints of CSV . . . 29
  - 2.3.2 Mako precisely discovers CSV unique-interval . . . 31
  - 2.3.3 Performance on real data . . . 31
  - 2.3.4 CSV subgraph illustrates breakpoints connections . . . 32
  - 2.3.5 Contribution of homology sequence in CSV formation 35
- 2.4 Conclusion . . . 37

3 SVision **39**
3.1 Introduction . . . 40
3.2 Material and methods . . . 40
3.2.1 Overview of SVision . . . 40
3.2.2 Three-channel coding of sequence . . . 41
3.2.3 Detecting CSVs from denoised images via tMOR . . . 43
3.2.4 Creating CSV graphs from denoised images . . . 44
3.2.5 Quality score of discoveries . . . 46
3.2.6 Training data and CNN model training . . . 47
3.2.7 Evaluating simple structural variants detection with real data . . . 48
3.2.8 Evaluating complex structural variant detection . . . 49
3.2.9 Analysis and validation of high-quality CSVs detected from HG00733 . . . 51
3.2.10 Data availability . . . 53
3.3 Results . . . 54
3.3.1 Evaluating simple SV detection with real data . . . 54
3.3.2 Performance of detecting complex structural variants . 56
3.3.3 CSV mediated gene structure change and genome evolution . . . 59
3.4 Conclusion . . . 62

4 SpotSV **63**
4.1 Introduction . . . 64
4.2 Material and methods . . . 66
4.2.1 Overview of SpotSV . . . 66
4.2.2 Modify reference sequence with predicted structural variants . . . 68
4.2.3 Generating denoised segments based on k-mers . . . 68
4.2.4 Calculating structural variant validation score . . . 69
4.2.5 Data availability . . . 70
4.3 Results . . . 71
4.3.1 Evaluating SpotSV with simulated data . . . 71
4.3.2 Validating structural variants in a well-characterized genome . . . 76
4.3.3 Structural variant breakpoint validation and accuracy 78
4.4 Conclusion . . . 81

5 Assessing reproducibility **83**
5.1 Introduction . . . 84
5.2 Materials and methods . . . 86
5.2.1 Read mapping and SV detection . . . 86
5.2.2 Evaluating recall and precision of each algorithm . . . 86
5.2.3 Identification and classification of PAV calls missed by each algorithm . . . 87
5.2.4 Evaluating breakpoint accuracy . . . 88
5.2.5 Examine call set overlaps between platforms and aligners 89
5.2.6 Data availability . . . 89
5.3 Results . . . 90
5.3.1 Evaluating the impact of aligners and platforms on detection algorithms . . . 90
5.3.2 Evaluation recall and precision of detection algorithms using different benchmarks . . . 92
5.3.3 Features of PAV calls missed by detection algorithms . 95
5.3.4 Examining the effects of platforms and aligners on breakpoint accuracy . . . 96
5.3.5 Effects of aligners on tumor SV detection . . . 98
5.4 Conclusion . . . 101

6 Conclusions and perspectives **105**
6.1 Conclusions . . . 105
6.2 Perspectives . . . 107

Bibliography **109**

English summary **121**

Nederlandse samenvatting **125**

Acknowledgements **129**

Curriculum vitae **131**

Chapter 1

Introduction and background

This thesis is about developing algorithms for structural variant detection, validation and analysis. We focus on long-read sequencing technologies. In this chapter, we explain the biological background, the sequencing technologies and computational approaches for the analysis of human genomes. We also mention our contributions and research questions.

1.1 Computational genomics

Computational genomics is an interdisciplinary field, combining biology, computer science, information engineering, mathematics and statistics, that develops and applies computational methods to analyze deoxyribonucleic acid (DNA) sequences for predictions or novel discoveries.

DNA is a molecule composed of two polynucleotide chains that form a double helix structure carrying genetic instructions for development, functioning, growth, reproduction, etc. The two DNA strands consist of monomeric units called nucleotides, where each nucleotide is composed of one of four nitrogen-containing nucleobases, cytosine (C), guanine (G), adenine (A) or thymine (T). From the computational perspective, a genomic sequence is a special type of string, consisting of four characters (i.e., A, T, C and G), and contains many repeated substrings.

One of the common applications of computational genomics is to assess the similarity between strings or in a set of strings, such that the candidate genes, genome evolution, genetic variants, etc. can be inferred or identified. Given that genetic variants are the major sources to form population differences and to drive diseases (i.e., cancer, autism disorder, Alzheimer, etc.), the detection of genetic variants has become a major focus in the

field of computational genomics since the development of high-throughput-sequencing (HTS) technologies [1]. Briefly, genetic variants are identified by comparing an individual genome (alternative sequence, ALT) with a reference genome (reference sequence, REF). To detect genetic variants in the sequencing era, computer science and statistical approaches have been applied, the Burrows-Wheeler Transform [2] and FM-index [3] were used to perform efficient sequence alignment, the convolutional neural network [4] was used to identify single-nucleotide-polymorphism (SNP), etc. Genome rearrangement or structural variants (SV) is another form of genetic variants, and usually affects a substring containing more than 50 characters, whereas a SNP only replaces one single character. In the past decade, great efforts have been made to generate longer DNA sequences and to optimize algorithms for the discovery and genotyping of genome rearrangements.

1.2 Emerging DNA sequencing technologies

The hybridization-based microarray approaches (i.e., for comparative genomic hybridization (CGH) and SNP microarrays) are first used to infer copy number gains or losses compared to a reference sample or population, whereas these approaches cannot identify balanced SVs (i.e., inversion), as well as their structures [1]. Another approach is the single-molecule analysis, such as fluorescent in situ hybridization (FISH) and spectral karyotyping, providing the first glimpses of common and rare SVs, such as the translocation mediated BCR-ABL fusion in Leukemia [1]. However, their low throughput and low resolution limit their application to a few individuals and to particularly large SVs (≈500kb to 5Mb).

The advent of next-generation-sequencing (NGS) technology or the so-called short-read sequencing promises to revolutionize the SV studies, and replaces the microarrays for high-throughput personal genomes variant detection. Most importantly, the NGS technology opens the field of detecting and genotyping SVs with HTS technologies, and DNA sequences produced by HTS technologies are termed as read [1]. So far, the most widely-used NGS technology is the read-pair technology, which has been applied to several population-scale genome studies, such as the 1000 Genomes Project [5], International Cancer Genome Consortium (ICGC) [6], Genome Aggregation Database (gnomAD) [7], etc. Starting from 2015, a considerable increase of novel HTS technologies that leverage single-molecule-sequencing (SMS) strategies, has led to platforms that produce reads several orders of magnitude longer than short-read data, enabling the direct detection of many previously

undetected SVs. The most representative SMS platforms are single molecule real-time sequencing (SMRT) invented by Pacific Bioscience (PacBio) and single stranded DNA nanopore sequencing invented by Oxford Nanopore Technology (ONT). The average DNA sequence length, i.e., read length, generated by PacBio and ONT is around 15kbp. To get the entire human genome of 3Gbp, an individual genome is usually sequenced multiple times, called sequencing coverage. For example, if a genome is sequenced at 30X coverage, the fragmented DNA sequences could span the entire genome 30 times. In this thesis, NGS or short-read data is referred to as paired-end sequencing, and long-read data or long-read sequencing is referred to as DNA sequences produced by PacBio and ONT sequencers.

1.3 Genome structural variations are important

In the past decade, widespread application of whole-genome HTS technology for the genetic variant detection has shown that difference between individuals is presented as single-nucleotide-variants (SNVs), small insertions and deletions (indels, <50bp) and SVs. Compared with SNVs and indels, SVs are extremely diverse in size and type, ranging from 50bp to megabases of the genome. SVs (Figure 1.1A) consist of copy number variations (CNVs), which include deletions (DEL), insertions (INS) and duplications (DUP), as well as balanced rearrangements, such as inversions (INV) and inter- or intra-chromosomal translocations (TRA, Figure 1.1B) [8]. These four types were discovered and defined in the early stages of the Human Genome Project (HGP) based on short-read sequencing, and we define them as simple SVs or canonical SVs.

Recently, based on the most advanced single-molecule-sequencing (SMS) technology, producing long-read data, a series of studies conducted by the Human Genome Structural Variation Consortium (HGSVC) has estimated that each human genome contains approximately 20,000–25,000 SVs, which doubles the number of SVs estimated by next-generation-sequencing technology (NGS) [9]. Remarkably, SMS facilitates the high-quality haplotype-aware human genome assembly and phased SV detection. The Phased Assembly Variant (PAV) allows researchers to establish their population frequency, identify ancestral haplotypes and discover new associations with respect to gene expression, splicing, and candidate disease loci [10].

Additionally, another special type of SVs, consisting of multiple combinations of the simple SV types, is called complex SV [11] (CSV, Figure 1.1C). In 2015, the 1KGP first profiled the CSVs of a healthy genome, of which the

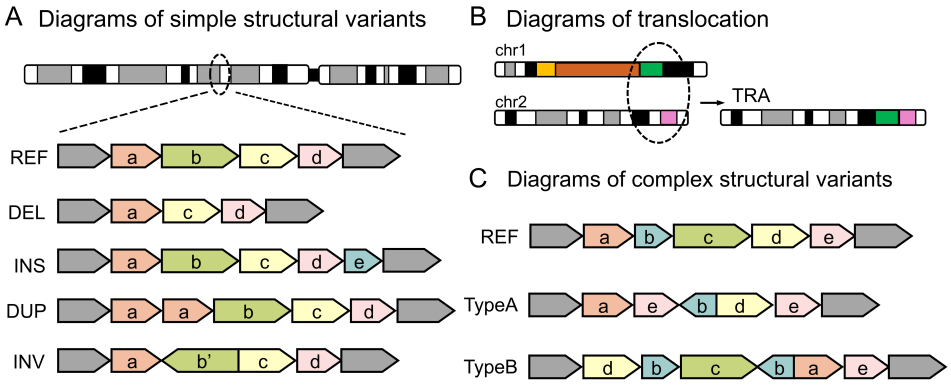


Figure 1.1: Diagrams of simple and complex structural variants. (A) Diagrams of four simple structural variants, including deletion (DEL), insertion (INS), duplication (DUP) and inversion (INV). (B) The diagram of a translocation (TRA), combining sequences from two different chromosomes. (C) The diagrams of two complex structural variant types (i.e., TypeA and TypeB).

CSVs were detected with intensive breakpoint analysis and manual curations based on SMS. This study first applied long-read data to resolve the structure of CSVs and suggested that 8% and 68% of the simple deletions and inversions are complex events [5]. In 2017, a group of researchers systematically analyzed the CSVs in a cohort of 689 patients with autism spectrum disorder and other developmental abnormalities, which was the biggest CSV study based on linked-read sequencing [12]. They identified 11,735 distinct large SV sites, and estimated each genome harbors 14 large CSVs on average. Notably, this study also found a high percentage of inversion associated CSVs, which took 84.4% of the detected CSVs.

Cancer is another complex disease, where the genome of cancer patients was changed dramatically during tumorigenesis, resulting in a great number of simple and complex SVs. The study conducted by ICGC profiled the SVs in 2,685 samples of 38 tumor types based on NGS, and first identified a group of unclassified or complex SV types in tumor genomes [6].

In Chapter 2 and Chapter 3, novel algorithms are developed to detect both simple and complex SVs from short- and long-read data, respectively. In addition, though SVs could be identified, an orthogonal approach to validate the correctness (i.e., breakpoint accuracy and type) is also important for future downstream analysis and clinical applications. Therefore, we developed

a novel algorithm to assess the quality of SVs detected by different algorithms in Chapter 4. Moreover, accumulating studies have revealed the unique strength of using long-read data to detect SVs from disease genomes, such as cancer and Mendelian disease. For example, a study of undiagnosed rare disease patients successfully identified three pathogenic CSVs that cannot be resolved by short-read data, suggesting the strength of using long-reads to characterize the exact breakpoints and structure of CSVs. In Chapter 5, we systematically evaluate the performance of the state-of-the-art long-read algorithms for both germline and somatic SV detection.

Besides the influence in downstream molecular and cellular processes, such as transcription and regulation [13], SVs are also important sources to understand the DNA damage repair mechanisms in the pathophysiological process of complex diseases such as cancer [14]. For example, the homologous recombination deficiency (HRD) has been used as an important biomarker to select drugs for a certain group of cancer patients [15].

In general, SVs are usually classified as recurrent and non-recurrent rearrangement to investigate their formation separately, where the recurrent SVs share the same size and genomic content in unrelated individuals, while the nonrecurrent ones have unique size and genomic content at a given locus in unrelated individuals [14]. CSVs often have more than one breakpoint junction and genomic interval of copy number change that can be observed at loci with susceptibility to nonrecurrent rearrangements, and replication-based mechanisms have been proposed to underlie the formation of CSVs as a result of interactive DNA template switches during replicative repair of single-ended, double-stranded DNA breaks [14]. In Chapter 2 and Chapter 3, the microhomology was identified to be the major mechanism for CSV formation, and we identified that different microhomology configurations at the breakpoint junction led to different forms of CSV. It should be noted that correct characterization of CSV formation requires accurate configuration of the breakpoint and structure, which is usually difficult to achieve using short-read data.

1.4 Detecting structural variation

Indeed, SVs of an individual genome manipulate the sequence of the reference genome, resulting in the so-called alternative sequence, and different types of SVs alter the reference sequence in different ways. In principle, all reads would be properly aligned if the sample's genome is identical to the reference, whereas the abnormally aligned reads footprint the signatures of SVs. For

instance, a deletion event indicates the sample genome missed one fragment of DNA sequence that was found in the reference genome (Figure 1.2A). The start and end position on the reference genome of the altered sequence are called *breakpoints* or *breakpoint junctions*, which are the junctions between alternative and reference sequence of the sample.

It should be noted that the SV breakpoint is defined according to the reference coordinate system, thus insertion only has one breakpoint junction on the reference compared with deletion, inversion and duplication (Figure 1.2B). The number of breakpoint junctions is often used to distinguish the simple and complex SVs, where CSVs usually have more than two breakpoint junctions. Afterwards, according to the altered sequence originating from different SV types, detection algorithms first build the SV signature model from the abnormally aligned reads for each type, where the model essentially depicts the pattern indicating how reads are aligned across the breakpoint junctions (Figure 1.3A). Therefore, to detect SVs, it is important to know how a specific SV type alters the reference sequence and its corresponding pattern inferred from the alignment. Once the SV signature models are built for each type, the detection algorithm would fit the observed read alignments with the expected model to make the detection. This approach is considered as a model-based approach, containing two major steps: i) SV signature modeling and ii) model fitting.

The model-based approach is initially designed for short-read data due to lack of SV spanning sequences, while assembly of short-reads provides longer DNA sequences and improves the detection performance. Briefly, the assembly based approach first collects all abnormally aligned reads to produce longer sequences based on the De Bruijn graph [21, 22] or string graph [23, 24]. Then, the assemblies are realigned to the focal regions, which is used to fit the prebuilt SV signature models for discoveries. Moreover, because the assemblies might span multiple breakpoint junctions, the assembly approach is widely used to detect CSVs from short-read data. Though the assembly approach is able to detect multiple breakpoints of CSVs, it often requires further efforts to filter redundant breakpoints and identify breakpoints belonging to the same events [5, 11].

The long-read technology produces even longer sequences than the assemblies from short-read data. It greatly simplifies simple SV detection from both signature modeling and model fitting because of variant spanning reads (Figure 1.3B). For CSV detection, though the long-read data avoid the assembly issues, it also follows the model-based approach, such as Sniffles [18], which is the only algorithm that detects two specific types of complex events with extra models (Figure 1.3C). However, CSVs are largely unexplored and

contain complex breakpoint configurations [25], making them challenging to model in a brute-force way. Moreover, current studies interpret and define CSVs in various ways, hindering the generalization of CSV study between researchers. Thus, one of the major objectives of this thesis is to develop novel algorithms for CSV detection without models.

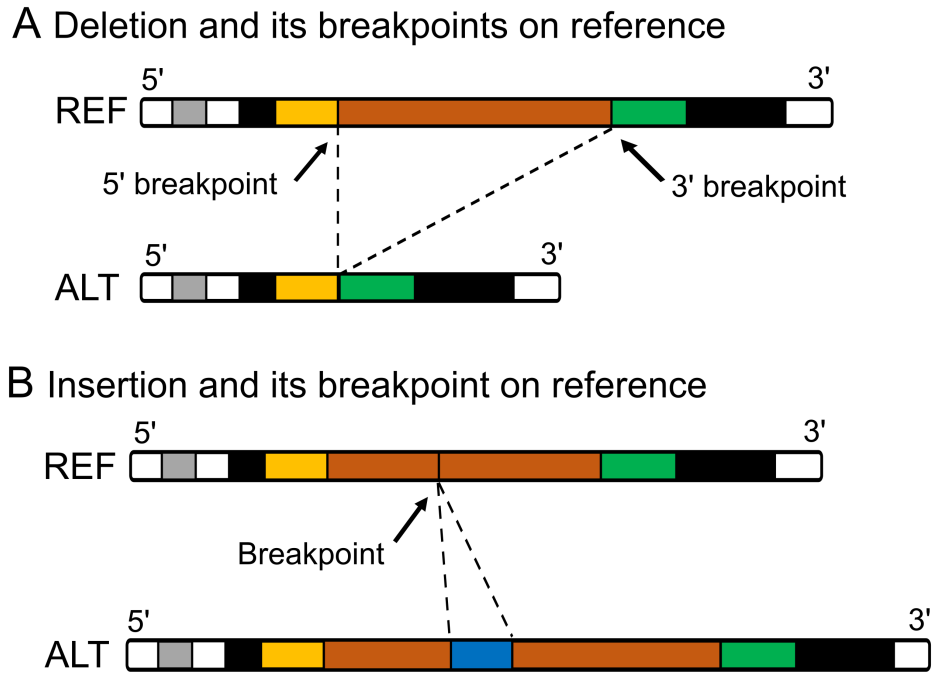


Figure 1.2: Breakpoints of two simple structural variants. (A) The breakpoints defined for a deletion, including the 5' breakpoint and 3' breakpoint on the reference. (B) The breakpoint defined for an insertion, which only has one breakpoint.

1.5 Pairwise sequence alignment for nucleotide sequences

Pairwise nucleotide sequence alignment has been used to investigate the differences between multiple genomes, to create the evolutionary tree for species, etc., which is one of the classical computational biology problems.

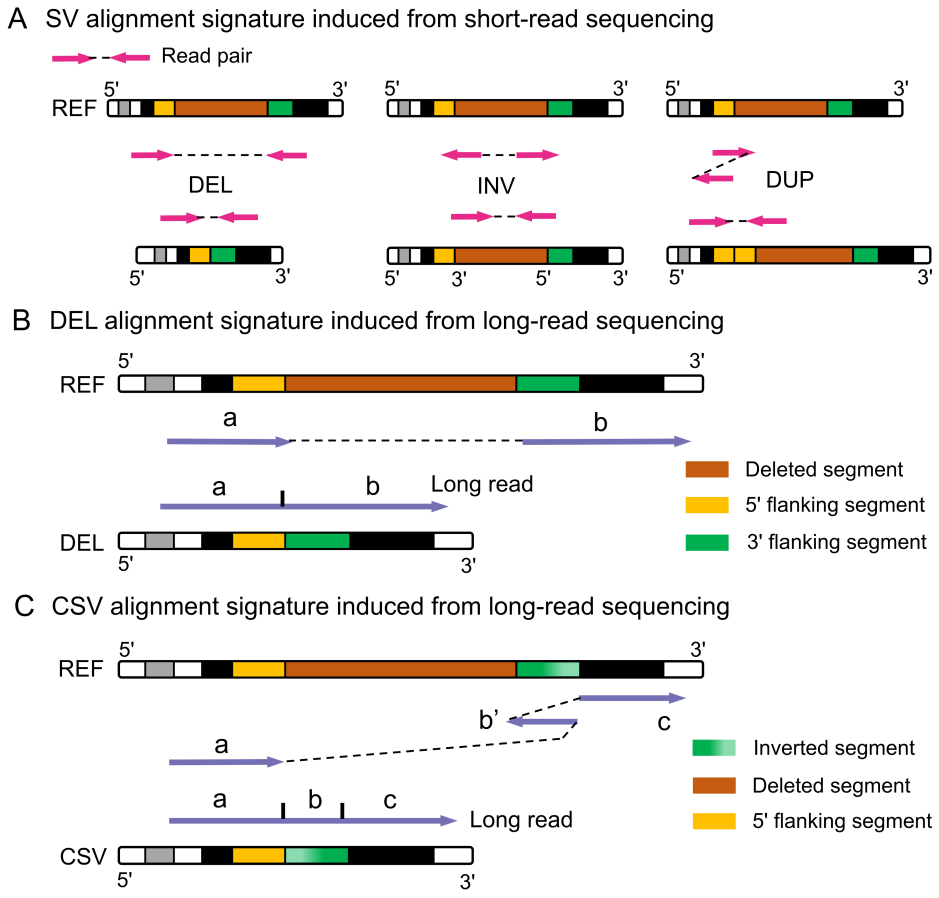


Figure 1.3: Structural variant alignment signature derived from short-read and long-read sequencing. (A) The short-read alignment signatures of three simple structural variants (SV), i.e., deletion (DEL), inversion (INV) and duplication (DUP). (B) The DEL alignment signature derived from long-read alignment. (C) The alignment signature of a complex structural variant (CSV) that is derived from long-read data.

MUMMer [16], using suffix-trees, is the most widely used algorithm for large-scale genome alignment, and it has been used to investigate the genome rearrangements between genomes.

Detecting SVs is similar to genome rearrangement detection between genomes, whereas the most advanced SMS technologies only produce fragmented DNA sequences, making it difficult for a MUMMer like approach to detect SVs genome-wide. Therefore, in order to detect SVs from an individual genome, the fragmented DNA sequences are first aligned to the human reference genome. The most common whole genome alignment algorithms, i.e., minimap2 [17] and ngmlr [18], adopt a typical seed-chain-align procedure to map the sequenced reads to the reference genome.

Briefly, for each query sequence (DNA sequence), minimap2 takes query minimizer as seeds, i.e., a longest exact match between query sequence and reference, and identifies sets of colinear matches as chains. Afterwards, dynamic programming is used to extend from the ends of the chains and to close regions between adjacent matches in chains. Fortunately, the long-read data spanning the SV site enables pairwise read and reference sequence comparison, promoting correct characterization of CSV structure. In Chapter 3, a light-weighted focal sequence realignment is proposed to refine the potential breakpoints of CSVs. This realignment approach is also based on seed-and-extension, whereas the gaps between nonlinear matches are not extended and considered to contain breakpoints.

The reference genome was first published in 2001 by HGP and has been significantly improved due to long-read technologies [19]. Although studies based on long-read data suggest that the reference could not easily serve as a standard genome, the routine genomic analysis, such as SV detection, still uses the reference genome as a universal genome. Thus, it should be noted that SVs of an individual genome are the different sequences compared with the reference genome, and the same reference is used to explore SVs in populations. Currently, the human genome is at version 38 (GRCh38), which now has fewer than 1,000 reported gaps, driven by the efforts of the Genome Research Consortium (GRC) [19].

The standard format of the alignment output is the Sequence Alignment Map (SAM) [20], and the Binary Alignment Map (BAM) is the binary version of a SAM file. The BAM file is usually used as input for SV detection, while recently another form of compressed BAM file (CRAM) is introduced for processing the large data volumes for population scale genome studies.

1.6 Usage of graphs for structural variants detection and analysis

A graph, consisting of nodes and edges, is an important data structure to model many types of relations and processes in physical, biological, social and information systems, and has a wide range of useful applications. There are three major graph types, i.e., undirected graphs, directed graphs and weighted graphs, and they have been broadly used for computational genomics. One of the most important applications of graphs is genome assembly, especially since the development of HTS technologies. The ultimate purpose of genome assembly is to build each chromosome from the fragmented DNA sequences. The method can be classified into reference guided assembly and de novo assembly, where de novo assembly achieves a real personal genome [26].

The development of long-read sequencing greatly promotes the de novo genome assembly, where two major graph data structures (i.e., De Bruijn graph and string graph) are used to produce long contiguous pieces of sequence (contigs). For the De Bruijn graph, each k-mer (a length k substring of a DNA sequence) is an edge directed from node A to node B if the $(k-1)$-mer in node A is a prefix, and that in B is a suffix of the k-mer [21]. Different from the De Bruijn graph, the nodes in string graphs are reads and edges connect two overlapping reads [24]. In the past decade, several optimized graph data structures based on either De Bruijn graph [22] or string graph [23] have been proposed to achieve the longest continuous sequence. Currently, with the PacBio hifi-fidelity (HiFi) reads, assemblers such as hifiasm [23] perform graph trio binning on the string graph to generate the final haplotype-resolved assembly of human genomes.

As we mentioned in the above section, realignment of short-read de novo assembly is a popular approach for both simple and complex SV detection. Another approach uses graphs but avoids assembly, aiming to identify fragmented DNA sequences that originated from the same longer piece of sequence, from which SVs could be accurately detected with short-read. For example, CLEVER [27] organized all abnormally aligned short-reads into a read alignment graph, where max-cliques were detected and statistically evaluated for their potential to reflect insertion or deletion based on the pre-built signature models. Inspired by CLEVER and the nature of SV, either SVs or CSVs would alter one or more genomic segments at a focal region and lead to disordered segment connections compared to the reference. Specifically, SVs or CSVs change the connection relation of DNA segments at the breakpoint junctions, and the reads across the junction will

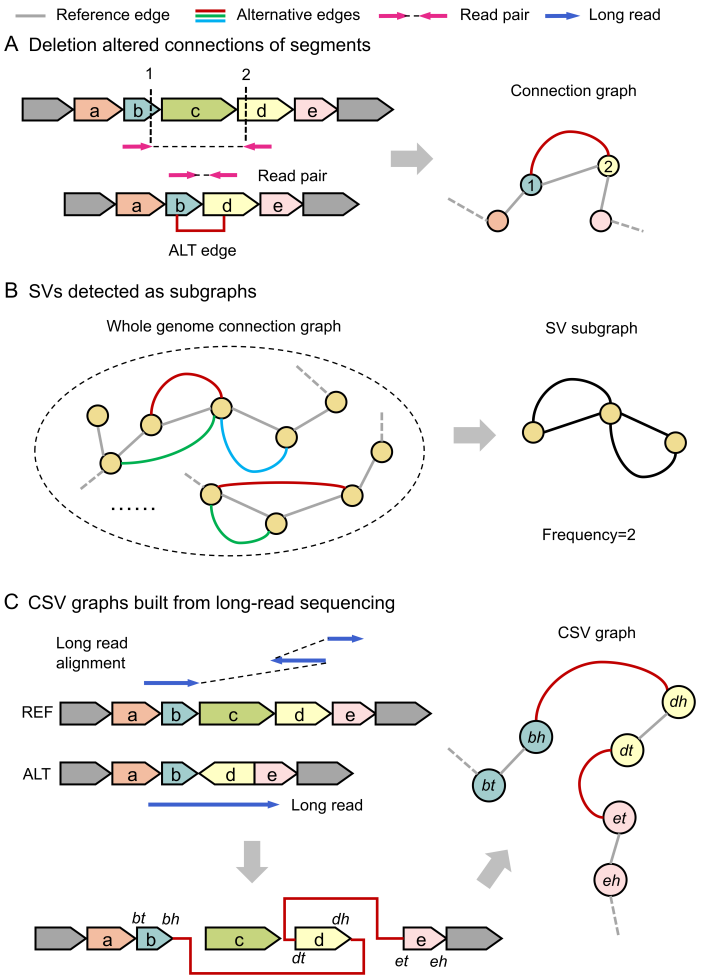


Figure 1.4: Overview and examples of detecting structural variants with a graph. (A) The connection graph created from short-read alignment on a deletion event, where node 1 and 2 indicate the mapping position of the read-pair on the reference genome (i.e., anchor position of read-pair). The connection provided by read-pair alignment is considered as alternative (ALT) edge in the graph. (B) Given the whole genome graph built based on (A), a structural variant is detected as a subgraph. (C) The graph is used to represent a complex structural variant based on long-read alignment, each node in the graph indicating a segment with tail and head. E.g., segment `b` produces two nodes in the graph: *bt* and *bh*. Similar to an edge in a short-read connection graph, the ALT edge is obtained from long-read mapping.

connect segments that are distant or discontinuous on the reference genome. For example, a deletion indeed connects two distant segments that are not adjacent on the reference genome (Figure 1.4A). Thus, we are able to create a segment connection graph (Figure 1.4A) from short-read (i.e., paired-end reads) alignments according to the signature model (Figure 1.3A), where the anchored positions of a mapped read are used as nodes and two nodes could be obtained from one paired-end read mapping (i.e., one end corresponds to one node). In terms of the edge set, one part of the edges are derived from the reference connection, indicating the identical connections between two adjacent segments on the reference, while the alternative edges are given by the abnormal aligned paired-end reads. Then, a SV or CSV is modeled as a subgraph that is involved in the genome wide segment connection graph (Figure 1.4B).

In Chapter 2, this segment connection graph is called signal graph, and we add extra attributes to the nodes and edges for CSV detection. However, using the current linear reference genome for variant detection, some major allele events have been observed and shown to be population specific variants, which might mislead further variant analysis. Therefore, another important application for graphs is to encode the population genetic diversities, producing the so-called graph genome [28, 29] or variation graph [30, 31, 32], which is then used as reference for variant detection. Given a graph reference, algorithms for efficient building, augmenting, storing, querying and variant calling are under active development [33, 34]. In 2018, the first genome-wide full pipeline of using graphs is developed, named vg [32], which improves read mapping sensitivity and increases the variant calling recall, and effectively removes reference bias.

Inspired by the above applications, we consider the graph to be a powerful data structure to represent complex events, such as the junction-balanced genome graph [35] which has been proposed to infer and classify complex rearrangements observed in tumor genomes. In Chapter 2, the CSV detected as a subgraph from the signal graph could be interpreted from the graph connections, which also enables the comparison of different types of CSVs. Inspired by the SV subgraph introduced in Chapter 2, the graph is used to represent and interpret complex events detected from long-read data in Chapter 3. This CSV mini-graph induced from long-read data contains nodes from matched segments between reference and alternative sequence, where the edges originate from the alternative sequence (Figure 1.4C). Note that in the long-read induced graphs, each node not only indicates the position on reference but also the matched segment sequences, which is different from the one derived from short-read data. Moreover, we introduce head and tail

annotation of each segment to indicate potential inversions. For example (Figure 1.4C), a segment is inverted, and the head is connected with an alternative edge observed in long read. Given the CSV mini-graph, we are able to compare different CSV events and identify those of the same type based on isomorphic graphs. Moreover, this mini-graph provides a so-called SV graph reference, such that the same event at identical loci could be identified or genotyped among populations via graph-based sequence alignment.

1.7 Frequent subgraph mining

Frequent subgraph mining raised great interest in the data mining community since 2000, and had a broad application in many fields, such as social media, chemical compound analysis, etc. The idea behind frequent subgraph mining is to "grow" candidate subgraphs, in either a breadth first or depth first manner, and then determine if the identified candidate subgraph occurs frequently enough in the graph data set for them to be considered interesting. Effective candidate subgraph generation is required to avoid the generation of duplicate or superfluous candidates, and the occurrences counting needs examination of graph isomorphism.

According to different applications, the optimization of frequent subgraph mining algorithms usually focuses on i) candidate generation strategy; ii) reduction of search space and iii) graph structure comparison. In Chapter 2, the abnormal short-read alignments, footprinting potential SVs, were coded in a signal graph (Figure 1.4B). Since SVs alter the focal genome by adding alternative edges or removing reference edges in the signal graph, the SV or CSV could be detected as a local maximal subgraph. In addition, a real SV or CSV graph structure usually occurs more frequently than the subgraph induced from alignment artifacts, thus the algorithm of detecting frequent local maximal subgraphs was developed to identify both simple and complex events (Figure 1.5). According to the specific application, this algorithm optimized the approach of saving the signal graph and comparing graph structure.

1.8 Objective and outline of the thesis

The main research questions for this thesis include i) detecting complex structural variants without prior knowledge and ii) reproducing structural variants detection among different datasets. The three main perspectives of this thesis are i) to develop novel algorithms for the detection of genomic

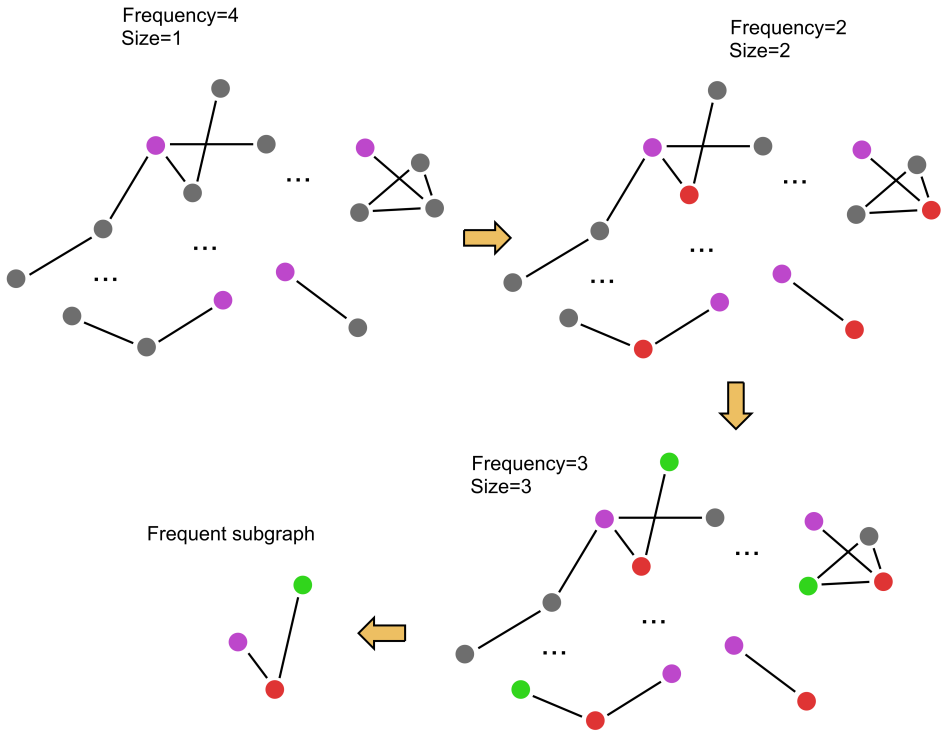


Figure 1.5: A toy example for detecting frequent maximal subgraphs from the signal graph. The subgraphs starts growth with purple node, from which subgraphs of size 2 and frequency 2 are obtained by adding red node. Furthermore, green node that satisfies the growth constraint are added to existing graph of size 2, resulting in subgraphs of size 3 and frequency 3. Finally, a frequent subgraph consisting of purple, red and green node is detected.

structural variation, especially for complex structural variations, from the short-read data (i.e., for the Illumina sequencer) and long-read data (i.e., for the PacBio and ONT platforms), ii) to develop structural variation validation algorithms and iii) to evaluate a pipeline or pipelines for detecting germline and clinically relevant structural variations.

First, for the paired-end sequencing, we use a graph data structure to encode the footprints of SVs from a given alignment and detect both simple and complex SVs by mining the frequent maximal subgraphs (Chapter 2). In the graph, a node consists of abnormal alignments, such as split-reads and discordant alignments. For the edges, two adjacent nodes are connected via a reference edge, and paired-end reads could connect two distinct nodes, referring to alternative edges. Afterwards, a frequent maximal subgraph is considered as a SV, and we set a frequency threshold for users to filter potential subgraphs made by background noise. However, since the paired-end reads are only approximately one hundred base pairs in length, it becomes challenging or even impossible to detect and interpret the whole structure of SVs, especially for complex ones. Given that the SMS provides long DNA sequencing that could cover the entire SV structure, in Chapter 3, we automate the detection and interpretation of both simple and complex SVs by recognizing the sequence differences coded in the image. Briefly, we visualize each alignment as a sequence similarity image, for which the multi-object recognition framework is applied to detect SVs without any predefined signature models. Then, we introduce a graph to represent and classify different classes of complex SVs.

Given that SMS based population scale SV study becomes common, the orthogonal approach to validate detected SVs is in great demand for the community, especially for the potential clinical application. In Chapter 4, inspired by the sequence similarity image, we provide a novel approach called SpotSV, to validate and characterize simple and complex SVs occurring in genomic regions of different complexity. In general, SpotSV validates a given SV in two major steps: i) simulating an alternative sequence with SV profile and ii) pairwise comparison of the reads and the simulated alternative sequence.

As the sequencing price drops, long-read technologies have been applied to study population genetic diversities, evolution, etc. Notably, in the past two years, several studies have shown the power of using long-read data to investigate disease genomes (i.e., for tumor and Mendelian diseases). Therefore, in Chapter 5, we aim to evaluate the existing long-read detection algorithms for both germline and somatic SV discovery. Specifically, we use five samples sequenced by PacBio HiFi and ONT, two alignment algorithms

and six widely-used detection algorithms, to examine and compare the performance of each detection algorithm.

In Chapter 6, conclusions are drawn and further perspectives are discussed.

In this thesis, Chapter 2 and Chapter 3 are based on the following publications:

- Jiadong Lin, Xiaofei Yang, Walter Kosters, Tun Xu, Yanyan Jia, Songbo Wang, Qihui Zhu, et al. "Mako: a graph-based pattern growth approach to detect complex structural variants." Genomics, proteomics & bioinformatics, 2021.

- Jiadong Lin, Songbo Wang, Peter Audano, Deyu Meng, Jacob Flores, Walter Kosters, Xiaofei Yang, Peng Jia, Tobias Marschall, Christine Beck and Kai Ye. "SVision: A deep learning approach to resolve complex structural variants." Nature Methods (Under revision, submission ID: NMETH-BC48137), 2022.

Moreover, we contributed to the following publications:

- Peter Ebert, Peter A. Audano, Qihui Zhu, Bernardo Rodriguez-Martin, David Porubsky, Marc Jan Bonder, Arvis Sulovari, Jiadong Lin, et al. "Haplotype-resolved diverse human genomes and integrated analysis of structural variation." Science, 2021.

- Peng Jia, Xiaofei Yang, Li Guo, Bowen Liu, Jiadong Lin, Hao Liang, Jianyong Sun, Chengsheng Zhang, and Kai Ye. "MSIsensor-pro: fast, accurate, and matched-normal-sample-free detection of microsatellite instability." Genomics, proteomics & bioinformatics, 2020.

1.9 List of abbreviations

Aligner Algorithm that maps long-read data to human reference genome

BAM Binary alignment map

Caller A certain algorithm for structural variant detection

Callset A set of structural variants discovered by detection algorithms

CCS Circular consensus sequencing

CGH Comparative genomic hybridization

CLR Continuous long read

CNV Copy number variation

CRAM Compressed BAM file

CSV Complex structural variant

GIAB Genome in a bottle

HGSVC Human genome structural variation consortium

HiFi Long-read data generated by PacBio CCS sequencing

HGP Human genome project

HRD Homologous recombination deficiency

HTS High throughput sequencing

Long-read Long-read data generated by SMS technology, such as a PacBio sequencer

NGS Next-generation sequencing

ONT Oxford nanopore technology

PacBio Pacific Bioscience

PAV Phased assembly variant

NGS Next generation sequencing

SAM Sequence alignment map

Short-read Paired-end data generated by NGS technology, such as an Illumina sequencer

SMS Single molecule sequencing

SNP Single nucleotide polymorphism

SNV Single nucleotide variant

SV Structural variant

1KGP 1000 Genomes project

Chapter 2

Mako: A graph-based pattern growth approach to detect complex structural variants

Abstract Complex structural variants (CSVs) are genomic alterations that have more than two breakpoints and are considered as the simultaneous occurrence of simple structural variants. However, detecting the compounded mutational signals of CSVs is challenging through a commonly used model-match strategy.
We systematically analyzed the multi-breakpoint connection feature of CSVs, and proposed Mako, utilizing a bottom-up guided model-free strategy, to detect CSVs from paired-end short-read sequencing. Specifically, we implemented a graph-based pattern growth approach, where the graph depicts potential breakpoint connections, and pattern growth enables CSV detection without pre-defined models. Comprehensive evaluations on both simulated and real datasets revealed that Mako outperformed other algorithms. Notably, validation rates of CSV on real data based on experimental and computational validations as well as manual inspections are around 70%, where the medians of experimental and computational breakpoint shift are 13bp and 26bp, respectively. Moreover, the Mako CSV subgraph effectively characterized the breakpoint connections of a CSV event and uncovered a total of 15 CSV types, including two novel types of adjacent segments swap and tandem dispersed duplication. Further analysis of these CSVs also revealed the impact of sequence homology in the formation of CSVs.
Mako is publicly available at https://github.com/xjtu-omics/Mako.

2.1 Introduction

Computational methods based on next-generation sequencing (NGS) have provided an increasingly comprehensive discovery and catalog of simple structure variants (SVs) that usually have two breakpoints, such as deletions and inversions [36, 37, 38, 39, 40, 41, 42]. In general, these approaches follow a model-match strategy, where a specific SV model and its corresponding mutational signal model are proposed. Afterward, the mutational signal model is used to match observed signals for the detection (Figure 2.1A). This model-match strategy has proved effective for detecting simple SVs, providing us with prominent opportunities to study and understand genome evaluation and disease progression [5, 9, 43, 44]. However, recent research has revealed that some rearrangements have multiple, compounded mutational signals and usually cannot fit into the simple SV models [5, 11, 45, 46, 47, 48] (Figure 2.1B). For example, in 2015, Sudmant et al. systematically categorized 5 types of complex structural variants (CSVs) and found that a remarkable 80% of 229 inversion sites were complex events [5]. Collins et al. used long-insert size whole genome sequencing (liWGS) on autism spectrum disease (ASD) and successfully resolved 16 classes of 9666 CSVs from 686 patients [12]. In 2019, Lee et al. revealed that 74% of known fusion oncogenes of lung adenocarcinomas were caused by complex genomic rearrangements, including EML4-ALK and CD74-ROS1 [48]. Though less frequently reported compared with simple SVs, these multiple breakpoint rearrangements were considered as punctuated events, leading to severe genome alterations at once [14, 43, 49, 50, 51]. This dramatic change of genome provided distinctive evidence to study formation mechanisms of rearrangement and to understand cancer genome evolution [12, 45, 46, 49, 51, 52, 53, 54, 55].

However, due to the lack of effective CSV detection algorithms, most CSV-related studies screen these events from the "sea" of simple SVs through computational expensive contig assembly and realignment, incomplete breakpoints clustering, or even targeted manual inspection [5, 11, 48]. In fact, many CSVs have already been neglected or misclassified in this "sea" because of the incompatibility between complicated mutational signals and existing SV models. Although the importance and challenge for CSV detection have been recognized, only a few dedicated algorithms were proposed for CSVs discovery, and they followed two major approaches guided by the model-match strategy. TARDIS and SVelter utilize the top-down approach, where they attempt to model all the mutational signals of a CSV event instead of modeling specific parts of signals. In particular, TARDIS [56] proposed sophisticated abnormal alignment models to depict the mutational signals reflected by

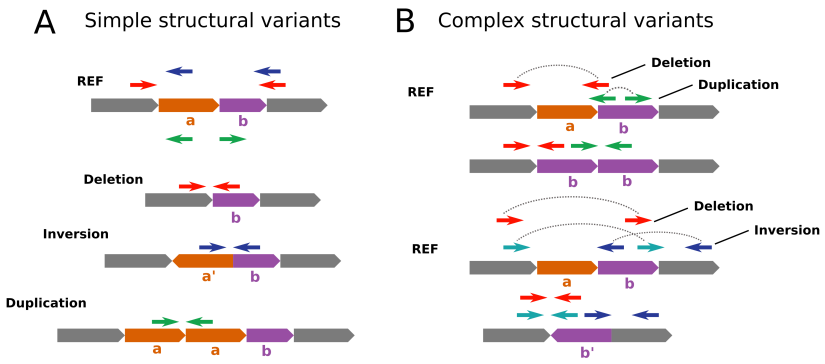


Figure 2.1: Explanation of simple and complex structure variants alignment models derived from abnormal read-pairs. (A) Three common simple SVs and their corresponding abnormal read-pair alignment on the reference genome, representing by red, blue, and green arrows. (B) The alignment signature of two CSVs, each of them, involves two types of signatures that can be matched by a simple SV alignment model.

dispersed duplication and inverted duplication. The pre-defined models were then used to fit observed signals from alignments for the detection of the two specific CSV types. Indeed, this was complicated and greatly limited by the diverse types of CSV. To solve this, SVelter [57] replaced the modeling process for specific CSVs with a randomly created virtual rearrangement. And CSVs were detected by minimizing the difference between the virtual rearrangement and the observed signals. On the other hand, GRIDSS [58] represents the assembly-based approach, which detects CSVs through extra breakpoints discovered from contig-assembly and realignment. Though the assembly-based approach is sensitive for breakpoint detection, it lacks certain regulations to constrain or classify these breakpoints and leave them as independent events. As a result, these model-match-guided approaches would substantially break up or misinterpret the CSVs because of partially matched signals (Figure 2.1B). Moreover, the graph is another approach that has been widely used for simple [27, 37] and complex [49, 59] SV detection. Notably, ARC-SV [59] uses clustered discordant read-pairs to construct an adjacency graph and adopts a maximum likelihood model to detect complex SVs, showing the great potential of using the graph to detect complex SVs. Accordingly, there is an urgent demand for a new strategy, enabling CSV detection without pre-defined models as well as maintaining the completeness

of a CSV event.

In this chapter, we propose a bottom-up guided model-free strategy, implemented as Mako, to effectively discover CSVs all at once based on short-read sequencing. Specifically, Mako uses a graph to build connections of mutational signals derived from abnormal alignment, providing the potential breakpoint connections of CSVs. Meanwhile, Mako replaces model fitting with the detection of maximal subgraphs through a pattern growth approach. Pattern growth is a bottom-up approach, which captures the natural features of data without sophisticated model generation, allowing CSV detection without pre-defined models. We benchmarked Mako against five widely used tools on a series of simulated and real data. The results show that Mako is an effective and efficient algorithm for CSV discovery, which will provide more opportunities to study genome evolution and disease progression from large cohorts. Remarkably, the analysis of subgraphs detected by Mako highlights the unique strength of Mako, where Mako was able to effectively characterize the CSV breakpoint connections, confirming the completeness of a CSV event. Moreover, we systematically analyzed the CSVs detected by Mako on three healthy samples, revealing a novel role of sequence homology in CSV formation.

In Section 2.2, materials used in this chapter and related methods are described in details. Then, results are discussed in Section 2.3 and conclusions are drawn in Section 2.4.

2.2 Materials and methods

In this section, we introduce the workflow of Mako and its major components for CSV detection. Moreover, related methods used for performance evaluation and orthogonal validation are described in details.

2.2.1 Overview of Mako

Given that a CSV is a single event with multiple breakpoint connections, breakpoints in the current CSV are not connected with false-positive breakpoints or those from unrelated events. Thus, we formulate the discovery of CSVs as maximal subgraph pattern detection in a signal graph. Accordingly, Mako detects CSVs with NGS data in two major steps, e.g., signal graph creation and subgraph detection (Figure 2.2). Firstly, Mako collects and clusters abnormally aligned reads as signal nodes and defines two types of edges to build the signal graph $G = (V, E)$, with $V = \{v_1, v_2, \ldots, v_n\}$ and $E = E_{pe} \cup E_{ae}$. Each signal node $v \in V$ is represented as $v = (type, pos, weight)$,

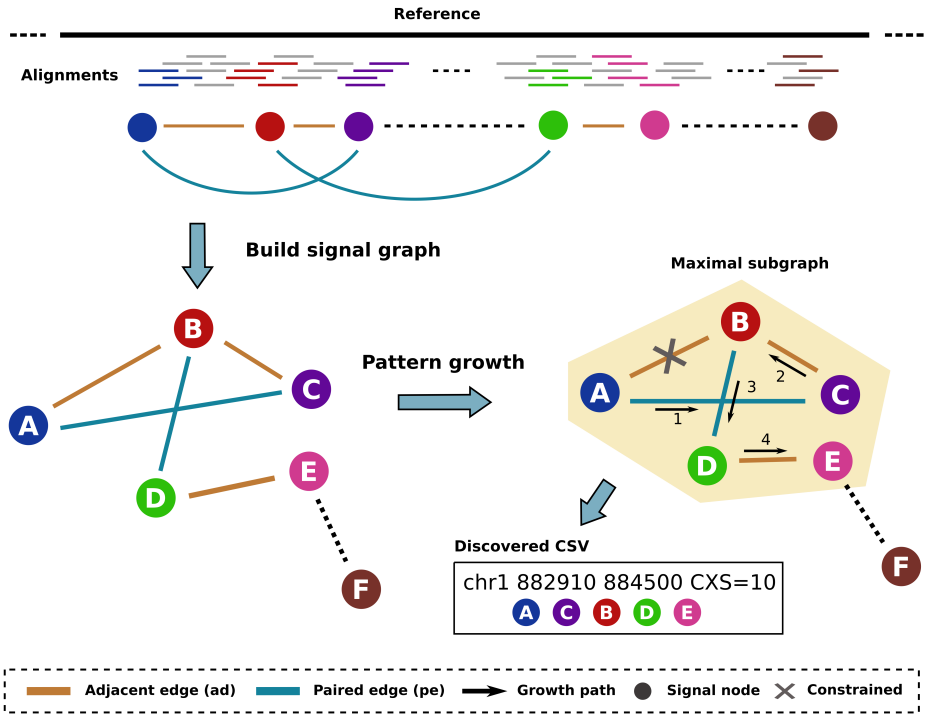


Figure 2.2: Overview of Mako. Mako first builds a signal graph by collecting abnormally aligned reads as nodes, and their edge connections are provided by paired-end alignment and split alignment. Afterward, Mako utilizes the pattern growth approach to find a maximal subgraph as a potential CSV site. In the example output, the maximal subgraph G contains nodes A, B, C, and D, whereas F is not able to be appended because of no existing edge (dashed line). The CSV is derived from this subgraph with estimate breakpoints and complexity score, where the discovered CSV subgraph contains four different nodes, one A_{ae} edge and two E_{pe} edges of type Del and Inv.

where *type*, *pos*, and *weight* denote the abnormal alignment type, node position, and the number of supporting abnormal reads, respectively. For the edge set, each edge in E_{pe} and E_{ae} is represented as $e_{pe} = (v_i, v_j, rp \cup sr)$ and $e_{ae} = (v_i, v_j, dist)$, respectively, where $v_i, v_j \in V$. Specifically, E_{pe} represents paired edges from a certain number of supporting read-pairs (rp) or split-reads (sr). E_{ae} indicates the adjacent edges induced from the reference genome, connecting two adjacent signal nodes at some distance ($dist$). Secondly, Mako applies a pattern growth approach to detect the maximal subgraphs as potential CSVs at the whole genome-scale. Meanwhile, the attributes of the subgraph are used to measure the complexity, and CSV types are determined by the edge connection types of the corresponding subgraphs (Figure 2.2).

2.2.2 Building signal graph

To create the signal graph, Mako collects abnormally aligned reads that satisfy one of the following criteria from the alignment file: 1) clipped portion with minimum 10% size fraction of the overall read length; 2) split reads with high mapping quality; 3) discordant read-pairs. As a result, one group of signal nodes is created by clustering clipped-reads or split-reads at the same position on the genome, which is filtered by *weight* and the ratio between *weight* and the coverage at *pos*. Another group of signal nodes is derived from clusters of discordant read-pairs, where the clustering distance is the estimated average insert size minus two times read length. It should be noted that a discordant alignment produces two nodes, and Mako separately clusters discordant alignments with multiple abnormally aligned types, such as abnormal insert size and incorrect mapping orientation. We adopt the procedure introduced by Chen [39] to avoid using randomly occurring discordant alignment. Additionally, edges are created alone with the signal nodes, where multiple types of edges might co-exist between two nodes.

2.2.3 Detecting CSVs with pattern growth

Pattern growth has been widely used in many areas [60, 61, 62, 63, 64, 65], such as Indel detection in DNA sequences [36, 54]. For CSV detection, the subgraph pattern starts at a single node and grows by adding one node each time until it cannot find a proper one (Algorithm I in Figure 2.3). During graph mining, the subgraph is allowed to grow according to the increasing order of pos value for each node, and backtracking is only allowed for nodes involved in the current subgraph. In Algorithm I, we build the

index-projection while graph mining, where the current graph G is used where prefix α and their corresponding suffix graphs are used to build the index-projection $G|_\alpha$. This index-projection contains nodes of coordinates bigger than its suffix coordinates on the reference genome. Note that pattern growth via adjacent edges is conditional on the distance constraint (*minDist*) because these edges are derived from the reference genome instead of alternatives. For example, Mako detects the maximal subgraph ACBD by visiting nodes A, C, B, and D, while the edge between D and E is constrained because of the larger distance (Figure 2.2).

Input: Signal graph $G = (V, E)$, **parameters** *minFreq*, *minDist*
Output: A set of CSV subgraphs $O = \{g_1, \ldots, g_n\}$ with $freq(g_j) \geq minFreq$

procedure *findMaximalSubgraph*(G, *minFreq*, *minDist*)
    Initialize *freq_types* to *type* frequency of nodes in V; $i \leftarrow 0$
    Build index-projection $G|_\emptyset$ of G
    for α **in** *freq_types* **do**
        Build index-projection $G|_\alpha$
        if $freq(\alpha) \geq minFreq$ **then**
            $i \leftarrow i + 1$; $g_i \leftarrow \alpha$
            multiLocPatternGrowth($O, g_i, G|_\alpha$, *minFreq*, *minDist*)
        end if
    end for
end procedure

Figure 2.3: Algorithm I: Detect maximal subgraphs.

Given that the signal graph contains millions of nodes at the whole genome scale, we adopt the "seed-and-extension" [66, 67] strategy to accelerate subgraph detection. Moreover, the discovered subgraphs not only differ in edge connections but also in node *type* of the subgraph. Therefore, we propose an algorithm that starts at multiple signal nodes of the same *type* at the whole genome scale, while extends locally for subgraph detection (Algorithm II in Figure 2.4). The parameter *minFreq* is used to measure the frequency of detected subgraphs, and Mako uses $minFreq = 1$ to avoid missing subgraphs of rare CSVs or incomplete ones. The detected CSV subgraph provides the connections between multiple breakpoints of a CSV, and the attributes of the subgraph are used to measure the complexity of CSVs. Accordingly, Mako defines the boundary of CSVs using the leftmost and rightmost *pos* value of the nodes and utilizes the number of identical node types multiplied

by the number of E_{pe} edges as a complexity measurement score, CXS. For example, the discovered CSV subgraph ACBD has a CXS score of 8 due to four different node types, e.g., A, C, B, and D, and two paired edges (Figure 2.2, a toy example of executing the algorithm is shown in Figure 1.5).

procedure *multiLocPatternGrowth*$(O, g, G|_g, minFreq, minDist)$
    Initialize *adj_list* with adjacent node direct after g through E
    for *node* **in** *adj_list* **do**
        if *nodeInRange*$(g, node)$ **then**
            $g' \leftarrow g + node$
            $O.append(g')$
            multiLocPatternGrowth$(O, g', G|_{g'}, minFreq, minDist)$
        end if
    end for
end procedure

procedure *nodeInRange*(g, v)
    Put the nodes in g in increasing order of *pos* value: $v_0, \ldots, v_m$
    $v' \leftarrow v_m$
    if $freq(v) > minFreq$ **then**
        if $dist(v', v) < minDist$ **then**
            return True
        else
            for $i \leftarrow m$ **downto** 0 **do**
                if $\exists\ e_{pe}$ between v and v_i **then**
                    return True
                end if
            end for
        end if
    end if
    return False
end procedure

Figure 2.4: Algorithm II: Multi-location subgraph growth.

2.2.4 Performance evaluation

Since CSVs contain multiple breakpoints, we propose two tiers of stringency for their evaluation, e.g., unique-interval match and all-breakpoint match.

For a unique-interval match, the correct predicted breakpoints shall be within 500bp distance to the leftmost and rightmost breakpoints of a benchmark CSV. For the all-breakpoint match initially proposed by Sniffles, the benchmark CSV is divided into separate subcomponents, and each of them should be correctly detected. For a CSV with inversion flanked by two deletions containing three components, the correct prediction of all breakpoints for the three components is considered as an all-breakpoint match. Meanwhile, if only one prediction is close to the leftmost and rightmost breakpoints of the CSV, this prediction is considered as a unique-interval match. For simulated CSVs, true positive (TP) is defined as predictions satisfying either match criterion, while predictions not in the benchmark are false positives (FP). False negatives (FN) are events in the benchmark set that are not matched by predictions. Whereas it is usually challenging to measure the false positives for real data due to the lack of a curated CSV set, we only consider the number of correct discoveries.

2.2.5 Preparing CSV benchmarks for performance evaluation

In this chapter, we use both simulated and real CSVs to benchmark the performance of different callers. We follow the workflow introduced by Sniffles [18] to create simulated CSVs. Firstly, VISOR [68] is used to create deletion (Del), inversion (Inv), inverted tandem duplication (Invdup), tandem duplication (Tandup), and dispersed duplication (Disdup). These events, termed as basic operations, are implanted and marked on the reference genome GRCh38 to generate an alternative genome. Secondly, CSVs are created by randomly adding basic operations to those marked operations, leading to a new genome harboring CSVs (CSV genome). Meanwhile, the purity parameter of VISOR is used to produce homozygous and heterozygous CSVs. Afterward, VISOR generates simulated paired-end reads based on the CSV genome with wgsim (`https://github.com/lh3/wgsim`) and aligns them to the reference genome with BWA-MEM [67]. According to the above-generalized simulation procedures, we create reported CSV types published by previous studies [5, 12] and randomized CSV types.

In terms of the real data, we are not aware of any public CSV benchmarks due to the breakpoint complexity and underdeveloped methods [5, 11, 57, 69, 70]. Fortunately, PacBio reads could span multiple breakpoints of CSVs, providing direct evidence to validate CSVs through sequence Dotplot [71]. Thus, we curate the CSV benchmark from a simple SV callset by breakpoint clustering and manual inspection. For SV clustering, each of them is considered as an interval, and hierarchical clustering with the average method is

used to find interval clusters. We then use the threshold that could produce the most clusters for merging clusters, which could potentially reduce the number of missed CSVs. Given these simple SV clusters, we apply Gepard to create Dotplots based on PacBio HiFi reads and manually investigate each Dotplot. Since CSVs are rare and might appear at the minor allele, we create Dotplot for each long read that spans the corresponding region.

2.2.6 Orthogonal validation of Mako detected CSVs

To fully characterize Mako's performance on real data, we use experimental and computational validation as well as manual inspections of CSVs from HG00733. The raw CSV calls from HG00733 are obtained by selecting events with more than one link type observed in the subgraph. For the experimental validation, Primer3 (https://github.com/primer3-org/primer3) is used to design PCR primers, where primers are selected within the extended distance but 200bp outside of the boundaries of the breakpoints defined by Mako. BLAT (https://users.soe.ucsc.edu/~kent/) search is performed at the same time to ensure all primer candidates have only one hit in the human genome. Afterward, we select amplification products with the expected product size and bright electrophoretic bands for Sanger sequencing. The obtained Sanger sequences are aligned against the reference allele of the CSV site and visualized with Gepard for breakpoint inspection.

As for the computational validation, two orthogonal data obtained from the Human Genome Structural Variant Consortium (HGSVC) are used, e.g., Oxford Nanopore sequencing (ONT) and HiFi contigs. We first apply VaPoR [72] on the ONT reads to validate CSVs, referred to as ONT validation. Additionally, we apply a k-mer based breakpoint examination based on haplotype-aware HiFi contigs, from which we calculate the difference between the k-mer breakpoints and predicted breakpoints.

Furthermore, we manually curate detected CSVs via Dotplots created by Gepard, which is similar to the procedure of creating the benchmark CSV for real data. For CSVs at highly repetitive regions, we further validate them according to specific patterns.

2.2.7 Data availability

The high coverage Illumina data (i.e., short-read data) for NA19240, HG00733 and HG00514 can be obtained from http://ftp.1000genomes.ebi.ac.uk/vol1/ftp/datacollections/hgsvsv_discovery/data/, and the SVelter callset for NA19240 is available at http://ftp.1000genomes.ebi.ac.u

k/vol1/ftp/datacollections/hgsvsvdiscovery/working/20160728SVelter_UMich/. The PacBio HiFi reads for NA19240, HG00733 and HG00514 were obtained from http://ftp.1000genomes.ebi.ac.uk/vol1/ftp/data_collections/HGSVC2/working/, the HiFi assembly for HG00733 is from http://ftp.1000genomes.ebi.ac.uk/vol1/ftp/datacollections/HGSVC2/working/20200628HHUassembly-resultsCCS_v12/assemblies/phased/, and the ONT reads for HG00733 are available at http://ftp.1000genomes.ebi.ac.uk/vol1/ftp/datacollections/hgsvsvdiscovery/working/20181210ONT_rebasecalled/. Moreover, the short-read data, long-read data and SV callset for SK-BR-3 can be obtained from http://labshare.cshl.edu/shares/schatzlab/www-data/skbr3/.

2.3 Results

In this section, we evaluate the performance of detecting CSVs using both simulated and real data. Moreover, we apply Mako to three samples (i.e., HG00514, HG00733 and NA19240), aiming to detect novel CSVs and understand CSV formation mechanisms. The original publication can be found at https://www.sciencedirect.com/science/article/pii/S1672022921001431, where related supplementary materials can be downloaded.

2.3.1 Mako effectively characterizes multiple breakpoints of CSV

The most important feature for a CSV is the presence of multiple breakpoints in a single event. Thus, we first examined the performance of multiple breakpoints detection for Mako, Lumpy, Manta, SVelter, TARDIS, and GRIDSS. The results were evaluated according to the all-breakpoint match criteria on both reported and randomized CSV-type simulations. Overall, for the heterozygous (HET) (Figure 2.5A) and homozygous (HOM) (Figure 2.5B) simulation, Mako was comparable to GRIDSS, and those two methods outperformed other algorithms. For example, GRIDSS, Mako and Lumpy detected 50%, 51% and 46% for reported HET CSV breakpoints, while they reported 53%, 54% and 44% for randomized ones. Because the graph encoded both multiple breakpoints and their substantial connections for each CSV, Mako achieved better performance on randomized events, which included more subcomponents than the reported ones. Indeed, by comparing reported and randomized simulation, the breakpoint detection sensitivity (Figure 2.5A, Figure 2.5B) of Mako increased, while that of other algorithms dropped except for GRIDSS. Although the assembly-based method, GRIDSS,

is as effective as Mako for breakpoint detection, it lacks a proper procedure to resolve the connections among breakpoints.

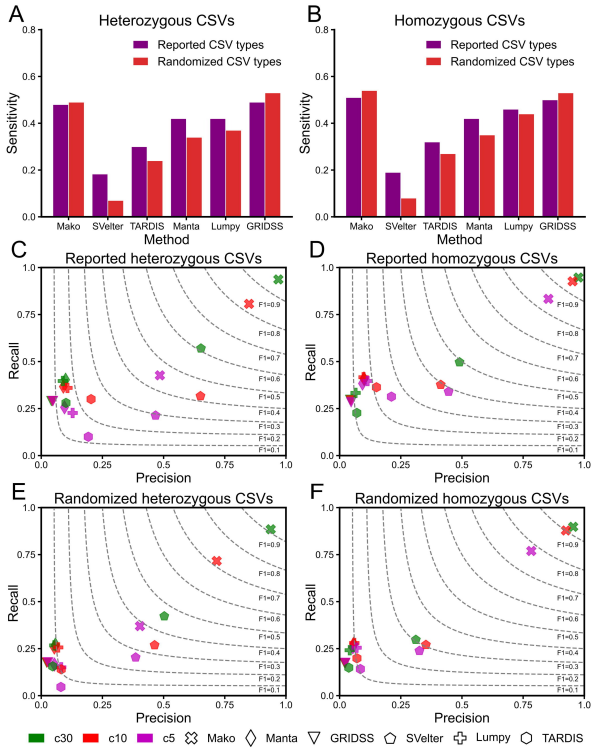


Figure 2.5: Performance comparison on simulated CSVs with different match criteria. All-breakpoint match (A and B) and unique-interval match (C–F) evaluation of selected tools for detecting simulated CSVs. (A) The sensitivity of detecting heterozygous CSVs breakpoints. (B) The sensitivity of detecting homozygous CSVs breakpoints. The red and purple bar indicates randomized and reported CSV types, respectively. (C) Evaluation of reported heterozygous CSV simulation. (D) Evaluation of reported homozygous CSV simulation. (E) Evaluation of randomized heterozygous CSV simulation. (F) Evaluation of randomized homozygous CSV simulation. From (C) to (F), the performance is evaluated by recall (vertical axis), precision (horizontal axis) and F-score (dotted lines). The right top corner of the plot indicates better performance. The c5–c30 indicates coverage, e.g., c5 indicates 5X coverage.

2.3.2 Mako precisely discovers CSV unique-interval

CSV is considered as a single event consisted of connected breakpoints, and we have demonstrated that Mako was able to detect CSV breakpoints effectively. However, the breakpoint detection evaluation only assesses the discovery of basic components for a CSV and lacks examination for CSV completeness. We then investigated whether Mako could precisely capture the entire CSV interval even with missing breakpoints. According to the unique-interval match criteria, Mako consistently outperformed other algorithms for both reported and randomly created CSVs, while SVelter and GRIDSS ranked second and third, respectively.

For the reported CSVs at 30× coverage (Figure 2.5C, Figure 2.5D), the recall of Mako was 94% and 92%, which was significantly higher than SVelter (49% and 57%) for both reported HET and HOM CSVs, respectively. Due to the randomized top-down approach, SVelter was able to discover some complete CSV events, but it may not explore all possibilities. Remarkably, we noted that Mako's sensitivity was even better for randomized simulation (Figure 2.5E, Figure 2.5F), which was consistent with our previous observation (Figure 2.5A, Figure 2.5B). In particular, at 30X coverage, Mako detected 203% more HET CSVs than SVelter (Figure 2.5E), probably due to the complementary graph edges for accurate CSV site discovery.

2.3.3 Performance on real data

We further compared Mako with SVelter, GRIDSS, and TARDIS on whole-genome sequencing data of NA19240 and SKBR3. Firstly, we compared the callsets of different callers, and we found that Mako shared most calls with GRIDSS (Figure 2.6A, Figure 2.6B), which was consistent with our observation in simulated data (Figure 2.5). Furthermore, we examined the discovery completeness of 59 (NA19240) and 21 (SKBR3) benchmark CSVs (Table 2.1). Because Manta and Lumpy contributed to the CSV benchmark sets, they were excluded from the comparison. The results showed that Mako performed the best for the two benchmarks with different CXS thresholds, while TARDIS ranked second (Figure 2.6C). Given that inverted duplication and dispersed duplication dominated the benchmark set and that TARDIS has designed specific models for these two types, TARDIS detected more events of these two duplication types than SVelter and GRIDSS. SVelter only detected three benchmark CSVs for SKBR3 because the randomized approach may not explore all combinations of CSVs. Based on the above observation, we concluded that the graph-based model-free strategy of Mako

performed better than that of either randomized model (SVelter) or specific model (TARDIS) with few computational resources.

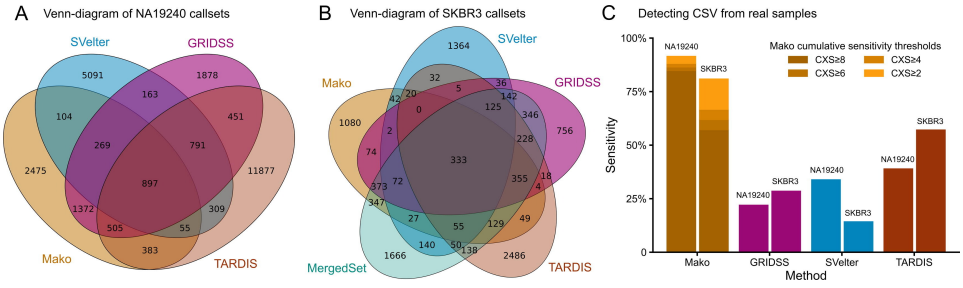


Figure 2.6: Overview of performance on NA19240 and SKBR3 for Mako, GRIDSS, SVelter and TARDIS. (A) Venn diagram of NA19240 callsets. (B) Venn diagram of SKBR3 callsets. The Venn diagrams are created by 50% reciprocal overlap via a publicly available tool Intervene with '`--bedtools-options`' enabled. The MergedSet is obtained from the original publication. (C) The percentage of completely and uniquely discovered CSVs from the NA19240 and SKBR3, respectively. The results of Mako are shown according to different CXS thresholds.

2.3.4 CSV subgraph illustrates breakpoints connections

Having demonstrated the performance of Mako on simulated and real data, we surveyed the landscape of CSVs from three individual genomes. Specifically, CSVs from autosomes were selected from Mako's callset with more than one edge connection type observed in the subgraph, leading to 403, 609, and 556 events for HG00514, HG00733, and NA19240, respectively (Figure 2.7A).

We systematically evaluated all CSV events in HG00733 via experimental and computational validation as well as manual inspection. For experimental validation, we successfully designed primers for 107 CSVs, where 15 out of 21 (71%, Table 2.2) were successfully amplified and validated by Sanger sequencing. The computational validation showed up to 87% accuracy, indicating a combination of methods and external data is necessary for comprehensive CSV validation. Further analysis showed that the medians of breakpoint shift were 13bp and 26bp compare to breakpoints given by experimental and computational evaluation. We observed that approximately 54% of CSVs were found in either STR or VNTR regions, contributing to 75% of all events inside the repetitive regions (Figure 2.7A). For the connection types, more

Type	NA19240	SKBR3	Description
disDup	15	12	Dispersed duplication
invDup	18	-	Inverted duplication
delINV	7	5	Deletion associated with inversion
delDisDup	5	1	Deletion associated with dispersed duplication
delInvDup	1	-	Deletion associated with inverted duplication
disDupInvDup	2	2	Dispersed duplication with inverted duplication
insINV	1	-	Insertion associated with inversion
tanTrans	1	-	Adjacent segments swap
delSapDel	8	1	Two deletions with inverted or non-inverted spacer
tanDisDup	1	-	Tandem dispersed duplications

Table 2.1: Summary of benchmark CSVs. The CSV type abbreviations and their corresponding descriptions are also listed.

than half of the events contain Dup and Ins edges in the graph, indicating duplication involved sequence insertion. Moreover, around 40% of the events contain Del edges (Figure 2.7B), showing two distant segment connections derived from either duplication or inversion events.

We further examined whether the CSV subgraph depicts the connections for each CSV via discordant read-pairs. Interestingly, we observed two representative events with four breakpoints at chr6:128,961,308–128,962,212 (Figure 2.7C) and chr5:151,511,018–151,516,780 (Figure 2.7D) from NA19240 and SKBR3, respectively. Both events were correctly detected by Mako, but missed by SVelter and reported more than once by GRIDSS and TARDIS. In particular, the CSV at chr6:128,961,308-128,962,212 that consists of two deletions and an inverted spacer was reported twice and five times by GRIDSS and TARDIS. The event at chromosome 5 that consists of deletion and dispersed duplication was reported four and three times by GRDISS and TARDIS. These redundant predictions complicate and mislead downstream functional annotations. On the contrary, Mako was able to completely detect the above two CSV events and also capable of revealing the breakpoint connections of CSVs encoded in the subgraphs. The above observations suggested that Mako's subgraph representation is interpretable, so that we

can characterize the breakpoint connections for a given CSV event.

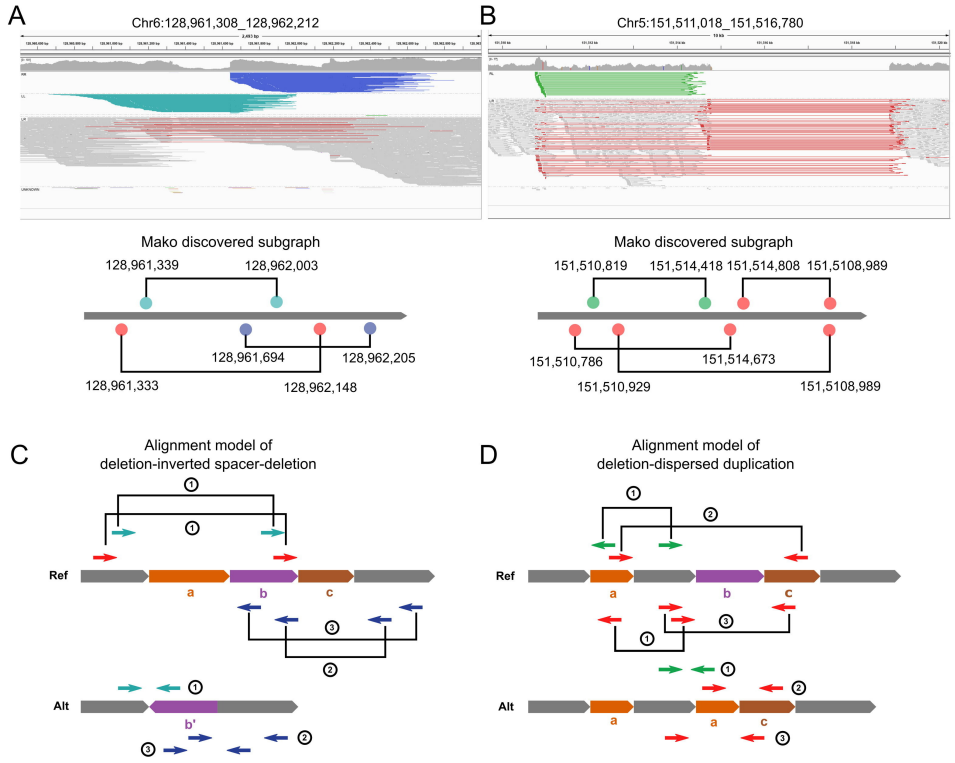


Figure 2.7: Two representative CSV subgraphs identified by Mako. The top panel of (A) and (B) are IGV views of the two events, and the alignments are grouped by read-pair orientation. The dark blue shows reverse-reverse alignments, light blue represents forward-forward alignments, green represents reverse-forward alignments, and red indicates the alignment of large insert size. The bottom panels of (A) and (B) are subgraph structures discovered by Mako. The colored circles and solid lines are nodes and edges in the subgraph. (C) The alignment model of deletions with inverted spacer. (D) The alignment model of deletion associated with dispersed duplication. In (C) and (D), short arrows are paired-end reads that span breakpoint junctions, and their alignments are shown on the reference genome with the corresponding ID in the circle. Note that a single ID may have more than one corresponding abnormal alignment type on the reference.

Validation Strategy	Total	Valid	Invalid	Inconclusive
Experimental (PCR succeeded)	21	15 (71%)	6 (29%)	-
ONT reads	609	256 (42%)	-	353 (58%)
HiFi contig		414 (68%)	191 (32%)	-
ONT reads or HiFi contig		544 (87%)	76 (13%)	-
Manual HiFi reads	609	440 (72%)	169 (28%)	-

Table 2.2: Summary of experimental and computational validation as well as manual inspection for CSVs.

2.3.5 Contribution of homology sequence in CSV formation

Given 1,568 detected CSVs from three genomes, we further investigated the formation mechanisms of these CSVs. Ongoing studies have revealed that inaccurate DNA repair and the 2–33bp long microhomology sequence at breakpoint junctions play an important role in CSV formation [14, 73, 74, 75, 76].

To further characterize CSVs' internal structure and examine the impact of homology sequence on CSV formation, we manually reconstructed 1,052 high-confident CSV calls given by Mako (252/403 from HG00514, 440/609 from HG00733, and 360/556 from NA19240) via Dotplots created by PacBio HiFi reads (Figure 2.8A). The percentage of successfully reconstructed events was similar to the orthogonal validation rate, showing CSVs detected by Mako were accurate, and the validation method was effective. The high-confident CSV callset contains 816 InsDup events with both insertion and duplication edge connections. Further investigation revealed that these events contain irregular repeat sequence expansion, making them different from simple insertion or duplications. Besides, we found two novel types, which were named adjacent segments swap and tandem dispersed duplication (Figure 2.8B). We inferred that homology sequence mediated inaccuracy replication was the major cause for these two types.

Furthermore, we observed that 134 CSVs contain either inverted or dispersed duplications. These CSVs containing duplications were mainly caused by microhomology mediated break-induced replication (MMBIR) according to previous studies [14, 74, 77]. It was known that different homology patterns cause distinct CSV types (Figure 2.8C, Figure 2.8D). Surprisingly, one particular pattern of homology sequence yielded multiple CSV types (Figure 2.8E).

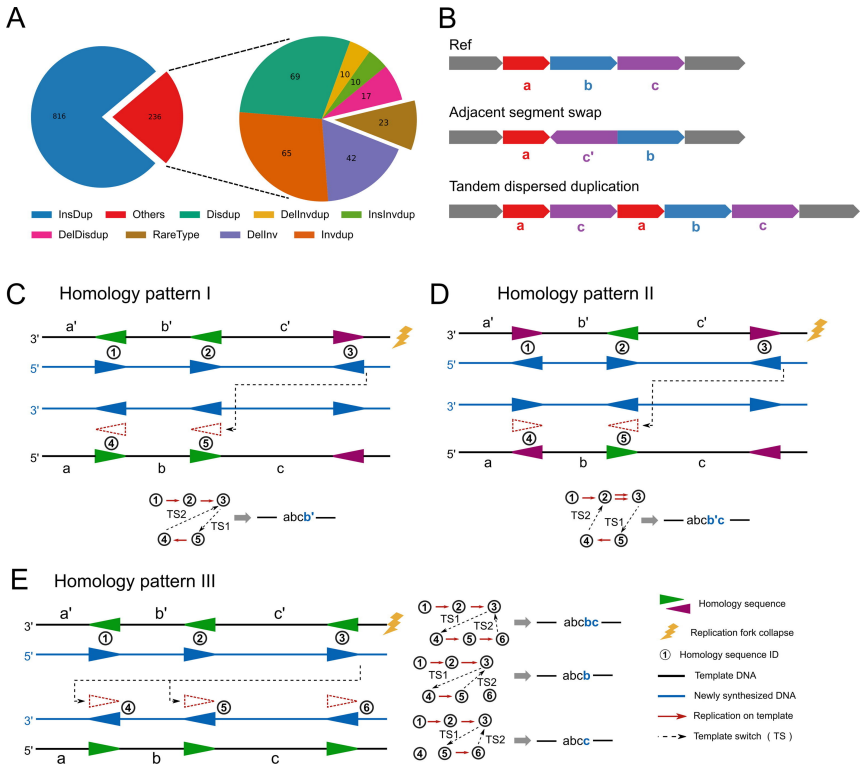


Figure 2.8: Overview of Mako's CSV discoveries from three healthy samples and proposed CSV formation mechanisms. (A) Summary of discovered CSV types, these types are reconstructed by HiFi PacBio reads, where a type with fewer than 10 events was summarized as RareType. (B) Diagrams of two novel and rare CSV types discovered by Mako. In particular, Mako finds three events of adjacent segments swap and only one tandem dispersed duplication. (C-E) Different replication diagrams explain the impact of homology pattern for MMBIR produced CSVs. In these diagrams, sequence abc has been replicated before the replication fork collapse (flash symbol). The single-strand DNA at the DNA double-strand break (DSB) starts searching for homology sequence (purple and green triangle) to repair. The above procedure is explicitly explained as a replication graph, from which nodes are homology sequences, and edges keep track of the template switch (dotted arrow lines) as well as the normal replication at different strands (red lines). If there are two red lines between two nodes, the sequence between these two nodes will be replicated twice, as shown in (D).

In particular situations of the three different homology patterns, DNA double strand break (DSB) occurred after replication of the c fragment. According to the MMBIR mechanism and template switch [53, 74, 75, 76], the pattern I (Figure 2.8C) and pattern II (Figure 2.8D) yield one output, but pattern III (Figure 2.8E) produces three different outcomes. The results provided additional evidence for understanding the impact of sequence contents on DNA DSB repair, leading to a better understanding of diversity variants produced by CRISPR [78, 79].

2.4 Conclusion

Currently, short-read sequencing is significantly reduced in cost and has been applied to clinical diagnostics and large cohort studies [48, 80, 81]. However, CSVs from short-read data are not fully explored due to the methodology limitations. Though long-read sequencing technologies bring us promising opportunities to characterize CSVs [18, 45, 46], their application is currently limited to small-scale projects, and the methods for CSV discovery are also underdeveloped. As far as we know, ngmlr combined with Sniffles is the only pipeline that utilizes the model-match strategy to discover two specific forms of CSVs, namely deletion-inversion and inverted duplication. Therefore, there is a strong demand in the genomic community to develop effective and efficient algorithms to detect CSV using short-read data. It should be noted that CSV breakpoints might come from either single haplotype or different haplotypes, where two simple SVs from different haplotypes lead to false positives. This may increase the false discovery rate due to a lack of haplotype information. Therefore, the combination of short-read and long-read sequencing might improve CSV discovery and characterization.

To sum up, we developed Mako, utilizing the graph-based pattern growth approach, for CSV discovery with 70% accuracy and 20bp median breakpoint shift. To the best of our knowledge, Mako is the first algorithm that utilizes the bottom-up guided model-free strategy for SV discovery, avoiding the complicated model and match procedures. Given the fact that CSVs are largely unexplored, Mako presents opportunities to broaden our knowledge of genome evolution and disease progression.

Chapter 3

SVision: A deep learning approach to resolve complex structural variants

Abstract Complex structural variants (CSVs) encompass multiple breakpoints and are often missed or misinterpreted by state-of-the-art long-read variant detection algorithms. As an increasing number of CSVs have been revealed through intensive breakpoint analysis and visual confirmation, there is an urgent demand of novel algorithms for detecting and characterizing CSVs at scale for future clinical applications. In this chapter, we develop SVision, a deep-learning based multi-object recognition framework, to automatically detect and characterize both simple and complex SVs from sequence image. SVision consists of three major modules: 1) an encoder that codes the differences and similarities between variant feature sequence and reference sequence as a denoised image; 2) a targeted multi-object recognition framework that detects and characterizes CSVs via a convolutional neural network in the denoised image; and 3) an illustrator that creates and unifies the detected CSV as a graph representation. Comprehensive evaluations on both simulated and real datasets reveal that SVision outperformed other algorithm and could accurately detect and characterize CSVs. Moreover, SVision resolved 80 CSVs with 25 distinct structures from an individual genome, from which we found CSVs disrupting important neural development genes and CSVs revealing the ancestral state of the human genome. The SVision program (v1.3.6) and trained model are available at GitHub (https://github.com/xjtu-omics/SVision).

3.1 Introduction

Complex structural variants (CSVs) contain multiple breakpoints and may delete, duplicate, and/or invert multiple segments of DNA, creating events that are both larger and more likely to be deleterious than simple structural variants [12, 82]. For instance, in 2015, by integrating short- and long-read sequencing, the 1000 Genomes Project (1KGP) revealed that 6% of deletions and 80% of inversions in NA12878 were complex events [5]. In 2020, the Pan-Cancer Analysis of Whole Genomes Consortium uncovered 22 out of 31 histology groups containing 10 to 1,000 complex breakpoints per sample through short-read sequencing of 2,658 cancer samples [6].

Previous short-read-based approaches to CSV detection require intensive breakpoint analysis and subsequent manual inspections with complementary data [11]. Even though long-reads have greatly facilitated phased structural variation (SV) detection [10], three major issues have impeded their usage in CSV detection. Firstly, the model-based inference approach, initially designed for simple SV discovery from short-read [1], requires the construction of each SV model for fitting aberrant alignment patterns and prohibits effective discovery of largely unexplored CSV structures [8, 18]. Secondly, ambiguous alignments at repetitive regions complicate SV discovery, leading to false calls or missing events. Lastly, the current subjective definition of CSV types based on predefined models lacks a unified and computer-interpretable framework [12], hindering cross-study comparison of CSVs.

In Section 3.2, materials and related methods are described in details. Moreover, results are discussed in Section 3.3 and conclusions are drawn in Section 3.4.

3.2 Material and methods

This section introduces the workflow of SVision and provide detailed description of SVision's three major components. Moreover, related methods, such as performance evaluation, CSV analysis, etc., are described in details.

3.2.1 Overview of SVision

SVision begins by encoding pairs of sequences, a given read and its counterpart in reference genome, as an image showing sequence similarity and difference adapting variant detection to a multi-object recognition problem amenable to an existing deep learning framework. SVision is composed of three core components: an encoder that represents the differences and similarities

between a variant supporting read and its corresponding segment in the reference genome as a denoised image, a targeted multi-object recognition (tMOR) framework that detects and characterizes CSVs via a convolutional neural network (CNN) in the denoised image, and an illustrator that creates and unifies each detected CSV as a graph representation from the denoised image (Figure 3.1A).

To generate a denoised image, the encoder first collects aberrant long-read alignments, the so-called variant feature sequence (VAR), and its aligned segment on the reference genome, referred to as reference sequence (REF). For a VAR, the encoder identifies matched and unmatched bases, from which the matched and the locally realigned unmatched sequences are combined to create VAR-to-REF and REF-to-REF images (Figure 3.1B). Since the repetitive sequences are present in both variant feature and reference sequences, the variant signature can be isolated and accentuated when the reference background is removed. Thus, a denoised image is created for each feature sequence by subtracting the REF-to-REF image from its corresponding VAR-to-REF image, which reduces false calls introduced by repeats.

In the tMOR step, since a denoised image might contain more than one SV, SVision uses a two-step image segmentation process to first obtain a one-variant image, containing the full structure of a SV. Then, SVision defines each location surrounding a breakpoint in the one-variant image as a Segment of Interest (SOI), and SOIs that are collected from a one-variant image are recognized as a single CSV through a pre-trained CNN.

The third component of SVision, illustrator, adopts a graph-based approach to depict different CSV structures. A given CSV graph structure and its topologically equivalent events are combined through detection of isomorphic graphs. Additionally, SVision reports the CSV graph in the Reference Graphical Fragment Assembly (rGFA) format introduced by MiniGraph [29]. Finally, SVision clusters similar one-variant images that supports an event and integrates CNN prediction probability of each one-variant image and similarity across one-variant images in a cluster to measure confidence of an event.

3.2.2 Three-channel coding of sequence

SVision takes the sequence alignment file in BAM format and reference file as input. The encoder consists of two major steps, i.e., variant feature sequence selection and sequence coding. Variant feature sequences are directly identified from long-read aberrant alignments containing SV signatures, such as inter-read and intra-read alignments. Intra-read alignments are derived from

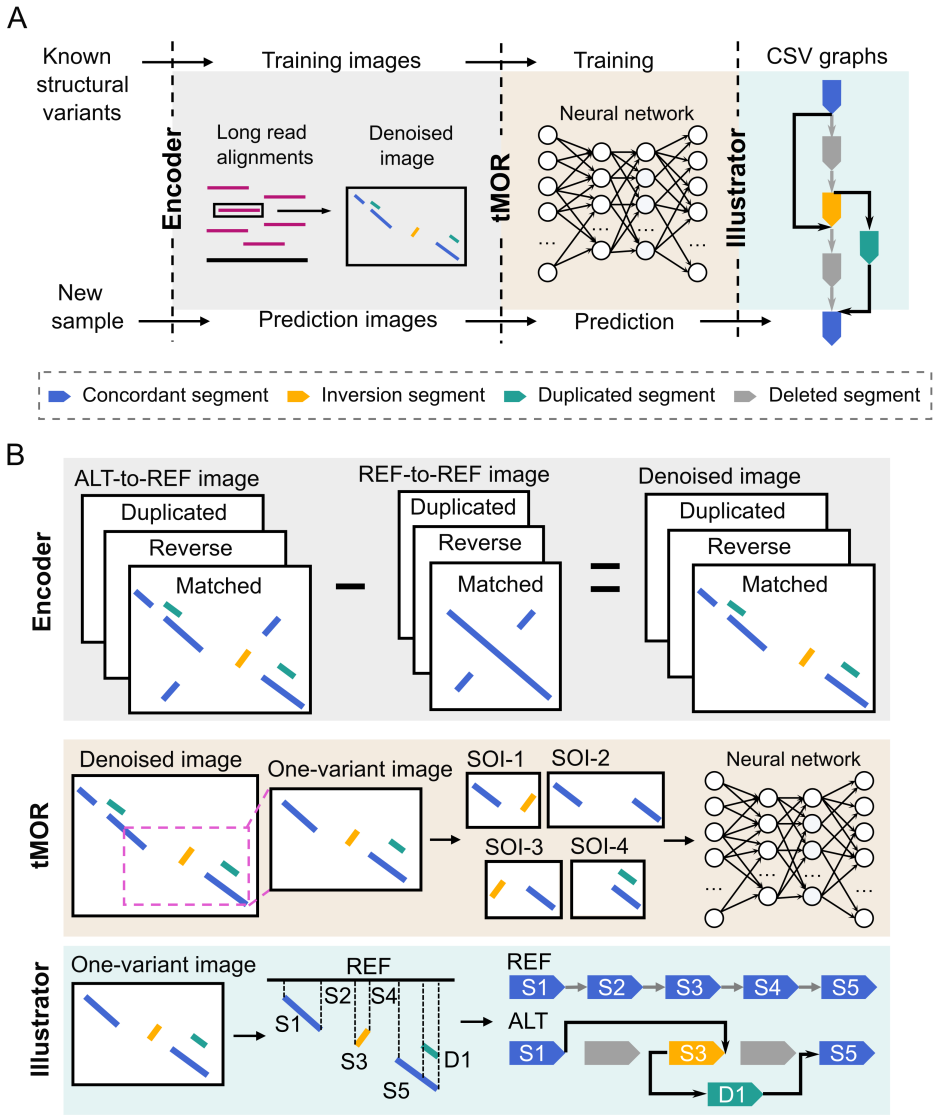


Figure 3.1: Overview of SVision. (A) Overview of the SVision workflow. (B) Details of three major modules implemented in SVision.

reads spanning the entire SV locus, while inter-read alignments are obtained from reads that are aligned to larger SV event, resulting in supplementary alignments. SVision identifies additional SV signatures by applying a k-mer based realignment approach for unmapped segment in feature sequence, such as 'I's from CIGAR string and gap sequence obtained from inter-read alignments. Then, sequence differences and similarities derived from matched and unmatched segments between variant feature sequence (VAR) and its corresponding segment on the reference genome (referred to as REF) is coded as an image.

The image contains three channels, including (0, 0, 255), (0, 255, 0), and (255, 0, 0), to code the matched, the duplicated and the inverted segments, respectively. Given the three-channel image, SVision first creates the REF-to-REF image through k-mer realignment. As for VAR-to-REF image, matched segments obtained from CIGAR string and supplementary alignments, originating from the aligner's outputs, are directly used for image coding to reduce computational cost, and realignment results are further added to complete image coding. The denoised image is obtained by subtracting the REF-to-REF image from the VAR-to-REF image. Because the background originates from reference sequence context, the encoder subtracts the segments of two images based on the REF sequence coordinates. Specifically, if segments from two images overlap on the reference dimension and their difference is larger than 50bp (minimum SV report size), the encoder keeps the non-overlapping part of the segment in the similarity image, where its coordinates are determined by the VAR-to-REF image. Finally, the denoised image of each variant feature sequence is created and saved as matrix along with segment information tables for further processing.

3.2.3 Detecting CSVs from denoised images via tMOR

In principle, for each denoised image, the regions where VAR and REF are identical must be a straight line while SVs introduce discontinuous segments. These discontinuous segments indicating putative variants and their breakpoints in the denoised image are surrounded by segment signatures, which are considered as breakpoint object and further defined as Segment of Interest (SOI). Since long reads are likely to span more than one variant in the denoised image, the tMOR contains a two-step image segmentation process for further SOI recognition. Specifically, the tMOR first obtains a so-called one-variant image, from the denoised image based on the following steps.:

1. Sorting and tagging. We sort all segments in the denoised image by

their positions on read in ascending order. Then, the major segment is defined according to the matched segments derived from CIGAR operations, while the minor segment should meet one of the following conditions:

- Condition 1: the segment is derived from the hash-table based realignment.
- Condition 2: the segment is inverted compared to the reference genome.
- Condition 3: the segment is totally covered by another one.

2. Creating one-variant image. SVision partitions the denoised image into several one-variant images via sequential combination of the major segments. Specifically, each major segment and its neighboring major segment along with the minor segments (if they exist) between them are used to create a one-variant image.

Afterwards, SVision clusters similar one-variant images by measuring the distance of segment signatures between one-variant images. Thus, one-variant images in a cluster supports the same variant, and the size of a cluster is termed as the number of variant supporting image. Secondly, SVision collects SOIs from each one-variant image. Unlike traditional multi-object recognition that uses complex algorithms to select regions of interest, the segment signatures in the one-variant image enable efficient SOI identification by sequentially combining both major and minor segments. Then, SOIs are used as input for CNN prediction, and the interpreted SV types are given by the labels involved in the training set, including deletion (DEL), inversion (INV), insertion (INS), duplication (DUP) and tandem duplication (tDUP). The CNN assigns the probability score to assess the existence of variant subcomponents in the one-variant image.

3.2.4 Creating CSV graphs from denoised images

SVision uses a graph to unify the definition of different CSV types and provides a computational method to compare different CSV graph structures. To create a CSV graph $G = (V, E)$, SVision first collects the node set $V = V_S \cup V_I \cup V_D$ of G. Specifically, $V_S = \{S_1, S_2, \ldots, S_n\}$, $V_I = \{I_1, I_2, \ldots, I_m\}$ and $V_D = \{D_1, D_2, \ldots, D_k\}$, where n, m and k are the number of skeleton nodes, insertion nodes and duplication nodes in the graph, respectively. Skeleton nodes are derived from major segments in a one-variant image and sequence between discontinuous major segments on REF (i.e., concordant

segments between VAR and REF). Insertion nodes consist of minor segments in the one-variant image, while insertion nodes with known origins are defined as duplication nodes, representing duplicated segments in the one-variant image. Moreover, each node $v_i \in V$ is represented as a tuple $v_i = (Seq, MathitPos, Strand)$, which represents a segment in the one-variant image. Here *Seq* indicates the segment sequence, *Pos* is the position of the segment on VAR and *Strand* represents the forward or reverse strand of the segment. The edges in G are collected by $E = E_{ad} \cup E_{dp}$. Here E_{ad} represents a set of adjacency edges $e^j_{ad} = (v_j, v_{j+1})$, connecting two adjacent nodes v_j and v_{j+1}, and E_{dp} represents a set of duplication edges e_{dp}, connecting the duplicated node with its known origin.

Given a graph G, a CSV could be interpreted by visiting each node through the E_{ad} edges. Assume the CSV path is given as "S1+S3-S3-S4+", where '+' or '-' indicates the direction of visiting a specific node, i.e., node *Strand*. Specifically, node S1 and S4 are visited in forward direction (+), while S3 is visited in reverse direction (-), so that the path should be "S1+S1+S3-S3-S4+S4+". But for simplicity, only the intermediate nodes, such as S3, are kept twice, whereas the start node (S1) and the end node (S4) are used once in the path.

Determining the isomorphism of two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ is a NP-hard problem, but the ordered nodes based on the reference simplifies this problem. Therefore, SVision first compares the numbers of edges and nodes between two graphs G_1 and G_2, which are considered as different if either number is different. On the other hand, if graph G_1 and G_2 have topologically identical path in addition to the same numbers of nodes and edges, they are isomorphic CSV graphs, i.e., $G_1 = G_2$. If graph G_1 and G_2 have the same number of nodes and edges but differ in paths, we further examine whether G_1 and G_2 share symmetric topology, since a variant might be identified on either forward or minus strand, i.e., from 5' to 3' or from 3' to 5'. In particular, we create a mirror graph G'_1 of the original graph G_1, and obtain a new path from G'_1. Similarly we also create G'_2 from G_2. Then, we cross compare whether the paths between G'_1 and G_2 as well as between G'_2 and G_1 are topologically identical. We consider G_1 and G_2 to be isomorphic if both comparisons are equal.

SVision keeps isomorphic graphs and symmetric graphs in two separate files, enabling search of CSV events of the same structure. For each variant call, SVision keeps all its breakpoints in the "BKPS" column in the INFO field and a type ("SVTYPE" column). Especially for CSVs, their breakpoints are kept with both coordinates and associated graph structure in the "BKPS"

and "GraphID" column, respectively. Note that the "GraphID" is used to search events of a specific graph structure in isomorphic and symmetric graph output files. Moreover, SVision involves the graph breakpoints induced from the CSV Reference Graphical Fragment Assembly (rGFA) file in the "GraphBRPKS" column. Note that the "GraphID" and "GraphBRPKS" columns are only reported when the parameter '`--graph`' and '`--qname`' are activated.

3.2.5 Quality score of discoveries

SVision uses a score function to measure the quality of each discovery based on consistency and prediction reliability derived from one-variant image clusters:

- One-variant image consistency. Intuitively, the non-linear segments in a given one-variant image indicate potential differences between REF and VAR. We thus first compute the non-linear score for all images that support each event, i.e., one-variant images originating from a variant feature sequence cluster. The non-linear score of a one-variant image is calculated by its segments coordinates and lengths. Specifically, for a one-variant image with segments:

$$nonlinear_score_i = \frac{\sum_k |k.\mathrm{ref}_{\mathrm{mid}} - k.\mathrm{read}_{\mathrm{mid}}| \times k.\mathrm{length}}{RefSpan}$$

  where the summation is over all segments k in image i, $k.\mathrm{ref}_{\mathrm{mid}}$ and $k.\mathrm{read}_{\mathrm{mid}}$ are the center of segment k on reference and read, respectively, and k.length is the length of segment k. Then we normalize the summation by dividing by *RefSpan*, which denotes the distance between the leftmost and rightmost coordinates of the one-variant image. Finally, for a SV of M supporting images, we calculate the consistency score with the following equation:

$$Consistency = \frac{Std(\{nonlinear_score_1, \ldots, nonlinear_score_M\})}{M}$$

  Here Std denotes standard deviation. Accordingly, we expect a smaller consistency value for high-quality SV predictions.

- Prediction reliability. This part evaluates the deep learning prediction quality. The last layer in the CNN architecture is a SoftMax layer,

which outputs the probability of the prediction results. Therefore, we use the average probability of all SOIs as the CNN reliability:

$$Reliability = \frac{\sum_s s.\text{softmax} \times 100}{\#\text{SOIs}}$$

where the summation is over all SOIs in a one-variant image. The reliability will range from 0 to 100 because the SoftMax probabilities always range from 0 to 1. We expect higher reliability values for accurate SVs.

Finally, we sum up the two features and normalize it to range from 0 to 100:

$$qual = Consistency + (1 - Reliability)$$

and

$$Normalized_score = \left(1 - \frac{\text{sum(Scores)} - \text{min(Scores)}}{\text{max(Scores)} - \text{min(Scores)}}\right) \times 100$$

where Scores $= \{qual_1, \ldots, qual_M\}$, and M is again the total number of images supporting this variant.

3.2.6 Training data and CNN model training

The CNN model in SVision is trained with both real and simulated simple SVs of DEL, INV, INS, DUP and tDUP, to avoid usually unbalanced numbers of SV types in real data. We obtained real SVs from NA19240 (4,282) and HG00514 (3,682) by selecting calls supported by both PacBio CLR reads and Illumina reads [9]. In this integrated real SV set, we labeled SVs with the above-mentioned five rearrangement types. We further used VISOR to simulate SV events with the parameters '`-n 4000 -r 20:20:20:20:20 -l 1000 -s 500`', and simulated the PacBio CLR reads. For all training SVs, their one-variant images and SOIs are created as we described in the above sections, leading to 75,000 SOIs (15,000 per type) in total, where 50% SOIs are from real events. These SOIs are shuffled for further CNN model training.

SVision adopts AlexNet, a widely-used CNN model, to recognize sequence differences in similarity images. The AlexNet architecture consists of five convolutional layers and three fully-connected layers. Specifically, the first convolution layer loads images of size $224 \times 224 \times 3$, and it uses the $11 \times 11 \times 3$ convolution kernel with stride 4. The last three layers are fully connected and contains a five-class SoftMax layer with inputs from the five preceding convolution layers. In the end, the input SOIs are detected as either INS,

DEL, INV, DUP, tDUP or mixed types for CSVs. We apply the idea of transfer learning to train CNN with 75,000 SOIs. First, the parameters of all layers in the CNN are initialized to the best parameter set that was achieved on the ImageNet competition. Afterwards, we fine-tune the parameters of the last three fully-connected layers on our data using back propagation and gradient descent optimization with a learning rate of 0.001. The loss function is defined as the cross entropy between predicted probability and the true class labels. Moreover, SVision's CNN architecture is lightweight and has far fewer layers than complex CNN models such as ResNet and Inception V3, which results in a highly efficient fine-tuning process with large batch size (default: 128) even on a single CPU machine. To evaluate the trained CNN model, we apply ten-fold cross validation, and the trained model at each round is applied to an independent test set of 7,500 SOIs derived from simulated SVs. Finally, SVision selects the model with the best performance.

3.2.7 Evaluating simple structural variants detection with real data

To benchmark the performance on HG002, we follow the procedure introduced by Genome-In-A-Bottle (GIAB), which has also been used by CuteSV. Briefly, the high confidence insertion and deletion calls and high confidence regions published by the GIAB consortium are used as ground truth. The HiFi reads are aligned to reference hg19 by pbmm2 (`https://github.com/PacificBiosciences/pbmm2`, v1.4.0) with parameter '`--preset CCS`', while ONT reads are aligned with pbmm2 default settings. The 5X and 10X coverage of HiFi and ONT data were further obtained with SAMtools [20] '`-s`' option. Sniffles (v1.0.12), CuteSV (v1.0.10), pbsv (v2.2.2), SVision (v1.3.6) and SVIM (v1.4.0) were applied to the pbmm2 aligned file with default parameters. The minimum supporting read was 2 and 3 for 5X and 10X data, while 10 was used for the original coverage. Moreover, the HiFi data of NA12878 was aligned to reference GRCh38 with minimap2 default settings, of which all callers were applied to detect SVs. To examine recall and precision, raw SV calls supported by at least five reads were used to compare with PAV calls. A correct detection (TP) should pass the 50% reciprocal overlap, while others were considered as false detections (FN). Then, the recall, precision and F-score are calculated as follows:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{F-score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Note that TP + FP is the total number of SVs detected by each caller, and TP + FN is the total number of SVs in the benchmark set.

3.2.8 Evaluating complex structural variant detection

First of all, the 10 simulated complex structural variant (CSV) types were derived from types reported by the 1000 Genomes Project (1KGP) [5] and a cohort study of autism spectrum disorder (ASD) [12]. The 1KGP reported CSV types included 'Ins and Del', 'Ins with Dup and Del', 'Ins with MultiDup and Del', 'MultiDel with Inverted or non-inverted spacer', 'Inv and Del' and 'Inverted Dup', were classified and combined to three basic CSV types (BCT). Specifically, 'Inverted Dup', labeled as BCT-ID1, was used to produce CSV types ID1 and ID2. 'MultiDel with Inverted or non-inverted spacer' and 'Inv and Del' (BCT-ID2) are simulated as ID4. Moreover, 'Ins and Del', 'Ins with Dup and Del' and 'Ins with MultiDup and Del' were considered as one type (BCT-ID3) but of different insertion sequence, which were used to produce ID5, ID6, ID7 and ID8.

Secondly, we expanded the simulated CSV types by introducing the study of ASD. In this research, we noticed that reported CSV types 'delINV', 'INVdel' and 'delINVdel' could be classified to BCT-ID2, and 'dupINV', 'INVdup', 'dupINVdup' and 'IR' were considered as BCT-ID1. BCT-ID3 was found as 'INSdel', 'cpdINSdel', 'dupINVdel', 'delINVdup' and 'dDUPdel'. Specifically, 'delINVdup' was simulated as ID5 and ID8, while 'dDUPdel' was simulated as ID6 and ID7. We also simulate 'dDUP', the dispersed duplication, as ID3, which was not included in 1000GP. In addition, we produced two novel types ID9 and ID10 by combining BCT-ID2 and BCT-ID3, where direct and inverted repeats were added to the deletion associated with inversion events.

In terms of simulation, a CSV was essentially the combination of breakpoints from simple structural variants (SSV), which were also termed as nested events. The simulation process contained four major steps. VISOR [83] was first used to simulate five simple SV (SSV) types (deletion, inverted dispersed duplication, inverted tandem duplication, tandem duplication and dispersed duplication), which were randomly implanted on reference genome GRCh38. Secondly, we followed the procedure introduced by Sniffles to simulate CSVs, where SSVs of the above five types were randomly added to the flanking regions of the existing SSVs implanted by VISOR in the first step. Accordingly, 3,000 SSV of five types were created by VISOR with parameters

'`-n 3000 -r 20:20:20:20:20 -l 500 -s 150`'. Then, we added extra variants required in predefined CSV types to existing SSVs by following the type order deletion, inverted dispersed duplication, inverted tandem duplication, tandem duplication and dispsersed duplication. For instance, we first used deletions as seeds to create all deletion involved CSV instances, and turned to instances of the next type until deletions were all used. Finally, the variation genome with CSVs was used as input for the VISOR LASoR module to simulate 30X HiFi reads and further aligned with ngmlr [18] (v0.2.7) default settings. Note that VISOR is only used to simulate variants at one haplotype in this chapter.

To examine the correctness of detected CSVs, we used closeness and size similarity to assess whether two events are identical according to Truvari (https://github.com/spiralgenetics/truvari/) introduced by GIAB. The closeness *bpDist* and size similarity *sim* between prediction and benchmark were 500bp and 0.7, respectively. Moreover, we only considered predictions with at least 10 support reads for the CSV performance comparison. For example, assume a particular benchmark CSV $[b.\text{start}, b.\text{end}, b.\text{size}]$, and a prediction $[p.\text{start}, p.\text{end}, p.\text{size}]$; then a correct region-match should satisfy the following equations:

$$\max(|b.\text{start} - p.\text{start}|, |b.\text{end} - p.\text{end}| \leq bpDist$$

and

$$b.\text{size} \times sim \leq p.\text{size} \leq b.\text{size} \times (2 - sim)$$

Comparably, the exact-match not only required region-match but also required the correct detection of all subcomponents of the CSV, including the subcomponent breakpoint type. Therefore, for a deletion-inversion that contained two subcomponents, e.g., INV and DEL, the exact-match becomes a three-step evaluation:

1. Region-match between predicted CSV and benchmark deletion-inversion event.
2. For each subcomponent, we examine the breakpoint closeness and event size as well as the detected type.
3. The correct detection should pass condition 1) and 2). The subcomponent match is considered as either deletion or inversion correctly detected in 2).

In this study, we only considered INS, DEL, DUP and INV as subcomponent types in the evaluation. Any benchmark CSVs without a matched prediction were counted as false negatives.

In addition, we used CSVs from NA12878 to assess the performance of SVision. The CSV set of NA12878 was obtained from the 1000 Genomes Project (1KGP) publication [5], including events from the supplementary tables 12 and 15 in the original publication, containing 62 and 251 CSV sites in hg19 coordinates, respectively. Based on the latest HiFi sequencing of NA12878 released by Human Genome Structural Variants Consortium (HGSVC) [10], we aligned HiFi reads with ngmlr (v0.2.7) default settings and manually inspected the Dotplot of every read that overlaps with the CSV site. Briefly, SAMtools and Gepard [84] were used to extract HiFi reads and generate Dotplot, respectively. Afterwards, SVision was applied to the ngmlr (v0.2.7) alignment for CSV discovery with default settings.

3.2.9 Analysis and validation of high-quality CSVs detected from HG00733

SVision was run under the default setting except parameters '`-s 5 --graph --qname`'. The HiFi reads of HG00733 were aligned to reference GRCh38 by ngmlr (v0.2.7) with the default setting. Firstly, the events detected by SVsion at low mapping quality regions, centromeres, genome gap regions, etc., were excluded from analysis. These regions were obtained from https://github.com/mills-lab/svelter/tree/master/Support/GRCh38 and the UCSC genome centromere for reference GRCh38. Then, we applied the following steps to filter CSVs from the raw callset:

1. Filtering CSVs of length larger than 100kbp;
2. Filtering CSVs without complete graph representation, where the path ends with other node types instead of 'S' and
3. For multiple CSVs at one site, we only kept the one with the largest number of supporting reads.

SVision revealed two special complex structures, i.e., a structure consisting of nodes 'S:2,I:2,D:1' and path 'S1+I1+I1+I2+I2+S2+' as well as another structure consisting of nodes 'S:2,I:1,D:1' and path 'S1+I1+I1+S2+', which were visually confirmed as local targeted site duplication and tandem duplication. Events of these two structures were also filtered because they were considered as simple events from biological perspective. Afterwards, we used RepeatMasker and tandem repeat finder (TRF) annotated files from UCSC genome browser to annotate the CSVs passed the filters through BEDtools [85] intersect option. The repeat type was assigned if the CSV region overlaps with the repeat element, while the size or percentage of overlaps was

not required. For CSVs with multiple repeat types, the one with the largest overlapping region with the CSV was chosen. Meanwhile, CSV was annotated as STR if the repeat unit length <7bp; otherwise, it was annotated as VNTR. Finally, we termed all CSVs outside of VNTR/STR regions as high-quality CSVs, which were further validated and used for further analysis. The PAV and short-read data matched CSV loci were obtained through BEDtools without requiring overlap size. For the short-read data, a matched CSV locus was considered as completely reconstructed if both breakpoint positions and types matched what SVision reported, otherwise as partially reconstructed events if either breakpoints or types agreed with SVision's prediction.

The PAV merged call set from 35 haplotype-resolved samples was used to explore the frequency of CSV on CNTN5. In addition, the RNA-Seq data of precuneus and primary visual cortex from both control and disease samples were obtained from a recent study of Alzheimer's disease [86] to understand the potential functional impact of CSV on CNTN5. The paired-end RNA data was aligned with hisat2 default setting, from which the duplicated exon signature could be observed from discordant read-pairs alignment, i.e., read-pair aligned in reverse and forward direction. The insertion-inversion-insertion event at chr9:74,283,222-74,283,473 detected by SVision, it was reported as insertion of variant id chr9-74283228-INS-1797 by a recent study conducted by HGSVC[10]. The insertional sequence was extracted from HiFi assembly and Blast against several primate genomes. Moreover, the assemblies of chimpanzee and gorilla were mapped to GRCh38 with minimap2 and called variant with PAV, from which the same insertion event was identified.

We validated 80 CSVs detected by SVision in HG00733 via 1) graph-based alignment; 2) contig-based visual confirmation; and 3) PCR and Sanger sequencing:

Graph-based alignment. For each CSV graph in rGFA format, we extracted the CSV locus spanning reads with SAMtools and aligned these reads to each CSV graph via GraphAligner (v1.0.12) with the default setting. A CSV was successfully validated if a single ONT read could be aligned to the corresponding variant path specified in the rGFA file. We then counted the number of long reads covering the entire VAR path as the number of support for this CSV event.

Contig-based visual confirmation. To examine the internal structure of CSVs, the phased-assembly specified in the PAV (v1.1.2, TIG_REGION column) at the reported variant region was used for further analysis. We first extracted the contig sequence harboring variant based on the coordinates provided in the 'PAV_TIG_REGION'. For example, a sequence containing variant was extracted from the h1 assembled genome for '1|1' and '1|0' genotype,

while from h2 assembled genome for '0|1'. In order to validate a CSV structure containing a complex insertion, we extended 5kbp both upstream and downstream the CSV region to extract the reference genome via BEDtools getfasta option, from which the origin of the inserted sequence could be identified. Afterwards, Gepard was used to create the Dotplot of contig sequence (vertical axis in the Dotplot) and reference sequence (horizontal axis in the Dotplot) for each CSV locus. Based on each contig Dotplot, the manual validation contained two tiers of metrics: 1) whether the reported region contains a variant; and 2) whether the SVision reported structure is identical to what was revealed by Dotplot. A CSV was considered completely reconstructed if both 1) and 2) were satisfied, while others were considered as inconclusive events.

PCR and Sanger sequencing. We first determined that about half of the 80 CSVs (39/80) were intractable for PCR due to their location within segmental duplications, the size of the amplicon needed to validate the rearrangement, or the simple repeat nature of the rearrangement. We then randomly selected 20 of the remaining rearrangements, and performed BLAT on the local region from the HG0733 assembly data. We next attempted to PCR each of the 20 CSVs. Briefly, we designed primers flanking the CSV or flanking breakpoints within the CSV for each of the 20 events. Next, we attempted to amplify each region using Takara LA taq. We obtained the predicted band size for 12 of the 20 variant loci; the remaining 8 regions did not amplify in 3 separate attempts with alterations of the PCR conditions and template amounts. All PCR products were sent to Sanger sequencing and validated as on target, and contained the correct amplicon with the breakpoint from the assembly and SVision call.

3.2.10 Data availability

Both the HiFi and Oxford Nanopore sequencing data for HG002 are available at the Genome in a Bottle (GIAB) FTP site (ftp://ftp.ncbi.nlm.nih.gov/giab/ftp/data/AshkenazimTrio/HG002_NA24385_son/). The PacBio HiFi sequencing data. For NA12878 is available at http://ftp.1000genomes.ebi.ac.uk/vol1/ftp/data_collections/HGSVC2/release/v1.0/assemblies/20200628_HHU_assembly-results_CCS_v12/haploid_reads/. Primary raw PacBio HiFi sequencing data for HG00733 is from http://ftp.1000genomes.ebi.ac.uk/vol1/ftp/data_collections/HGSVC2/working/20190925_PUR_PacBio_HiFi/, and the high-quality phased assemblies is available at http://ftp.1000genomes.ebi.ac.uk/vol1/ftp/data_collections/HGSVC2/working/20200417_Marschall-Eichler_NBT_hap-assm/.

The Oxford Nanopore sequencing data used for graph-based validation is from http://ftp.1000genomes.ebi.ac.uk/vol1/ftp/data_collections/hgsv_sv_discovery/working/20181210_ONT_rebasecalled/. The latest HG00733 PAV (v1.1.2) call is from http://ftp.1000genomes.ebi.ac.uk/vol1/ftp/data_collections/HGSVC2/working/20210806_PAV_VCF/, and the latest release of PAV calls for 35 samples is from http://ftp.1000genomes.ebi.ac.uk/vol1/ftp/data_collections/HGSVC2/release/v2.0/integrated_callset/. The RNA-Seq data of precuneus and primary visual cortex could be accessed in SRA with PRJNA720779.

3.3 Results

In this section, we first evaluate the performance of detecting simple SVs using benchmark sets of HG002 and NA12878. Then, the performance of detecting CSVs is assessed on both simulated CSVs and real CSVs in NA12878. We further apply SVision to HG00733 to detect novel CSV loci and types.

3.3.1 Evaluating simple SV detection with real data

To start with, we explored how well the sequence-to-image coding schema and the CNN model perform across different long-read sequencing platforms for canonical SV detection, where SVision, CuteSV, pbsv, SVIM and Sniffles were applied to the HG002 genome (≈27X PacBio HiFi and ≈47X Oxford Nanopore, ONT). The results showed that SVision outperforms other callers at different coverages, where the F-score of SVision ranged from 0.83 to 0.90 for HiFi and from 0.76 to 0.92 for ONT (Figure 3.2A). In addition, we examined the performance with NA12878 PAV calls released by HGSVC [10], consisting of deletions, insertions and inversions. The result was consistent with the performance evaluated by HG002 benchmark, where SVision achieved the highest F-score (Figure 3.2B). Moreover, SVision was more sensitive than other callers across different SV size range with high precision, especially for SVs ranged from 50 to 300bp, where SVision detected 10% more PAV calls than others (Figure 3.2C, Figure 3.2D). Altogether, our results suggested that SVision was able to detect canonical SVs accurately compared with the model-based callers, and SVision was versatile across sequencing platforms and varying sequencing depth.

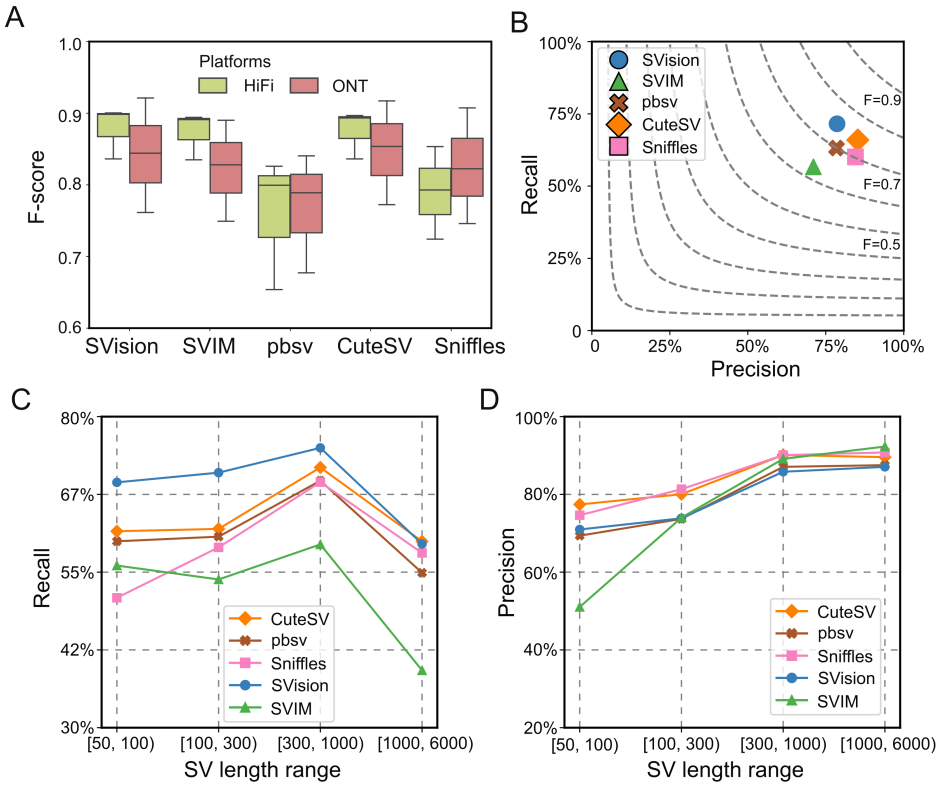


Figure 3.2: Performance of detecting simple structural variants from real data. (A) F-score of detecting variants in HG002 evaluated with Truvari. (B) Recall and precision of detecting NA12878 Phased Assembly Variant (PAV) calls. (C) Recall of detecting NA12878 PAV calls at different size range. (D) Precision of detecting NA12878 PAV calls at different size range.

3.3.2 Performance of detecting complex structural variants

Furthermore, the performance was assessed on simulated CSVs of 10 types extracted from the 1KGP [5] and a cohort study of autism disorders [12]. The simulated genome harboring 3,000 CSVs (300 per each of 10 types) was created on one haplotype and sequenced at 30X coverage in HiFi mode. Motivated by Sniffles [18], we introduced region-match and exact-match for performance evaluation. The region-match requires correct detection of the CSV site, while exact-match requires correct detection of both the CSV site and its subcomponents (i.e., the deletions and insertions that comprise a CSV). For the region-match, the recall and precision of SVision were 91% and 93%, while those of the second-best tool CuteSV were 62% and 36%, respectively (Figure 3.3A). A significant proportion of CSV sites were missed by CuteSV because the observed novel signatures were beyond the predefined SV models, while the low precision could be largely attributed to partial CSV detection (Figure 3.3B). By exact-match, SVision detected 89% of the CSVs, more than double of Sniffles, while other callers were not able to characterize any CSVs (Figure 3.3A).

To examine the performance of detecting CSV from real data, we first manually curated 62 complex deletion and 251 complex inversion sites in NA12878 reported by 1KGP [5]. As a result, 18 CSVs were verified (two from the 62 deletion sites, 16 from the 251 inversion sites), while the rest of the events were simple SVs (one duplication, two inversions and 57 deletions) (Figure 3.4A). This suggested the manual curation through visualization was one of the critical steps for CSV detection. Given the manually curated CSV benchmark, SVision automatically and correctly characterized the internal structure of all CSVs (Figure 3.4A), including two CSVs failed to interpret with short-read data, i.e., a deletion replaced by an inverted segment and a duplicated segment (Figure 3.4B) and a complex insertion consisting of inverted duplication and dispersed duplications (Figure 3.4C). Moreover, SVision was able to distinguish simple event from the complex ones at complex genomic regions. For example, a simple deletion (chr9:71,895,338-71,896,537) at a region flanked by duplicates (inverted and dispersed) was detected as CSV based on short-read (Figure 3.4D), while SVision correctly detected it as a simple deletion. Taken together, our results suggest that SVision can detect both simple and complex structural variants from long-read data with high sensitivity and accuracy.

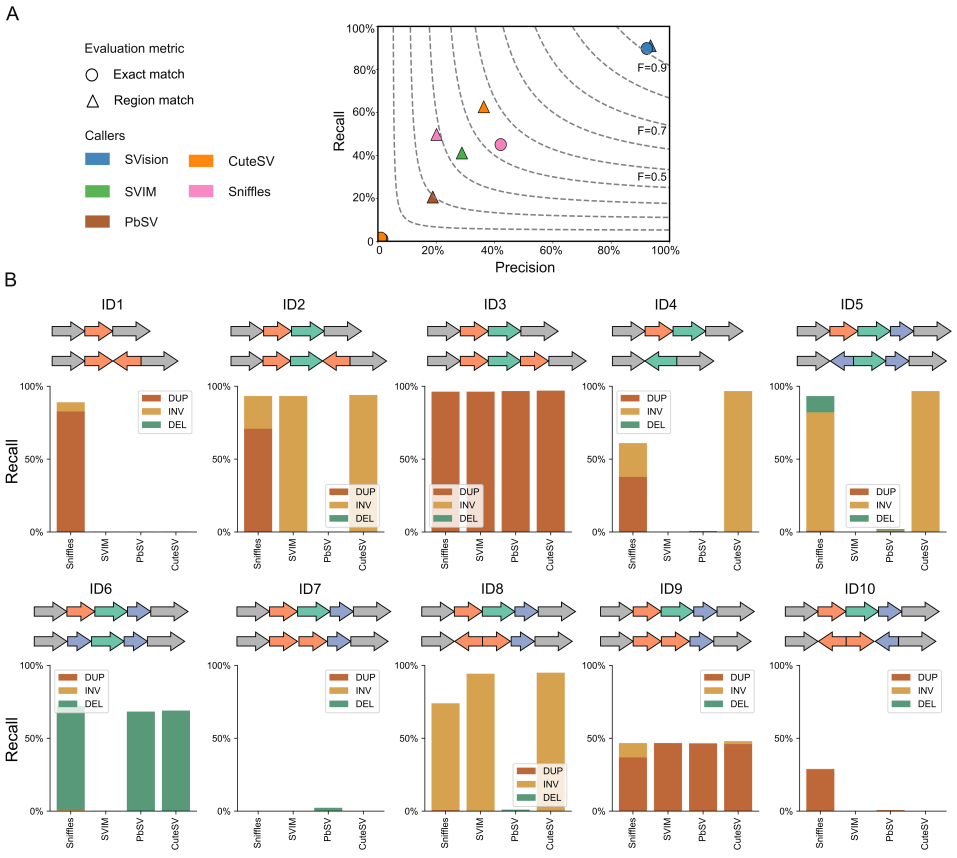


Figure 3.3: Performance of detecting simulated complex structural variants. (A) Performance of detecting simulated complex structural variants (CSVs), which was evaluated with recall (vertical axis), precision (horizontal axis) and F-score (F, dashed line). (B) The recall of model-based callers for detecting subcomponents (i.e., DUP-duplication, DEL-deletion, INV-inversion) of CSV evaluated with region-match. Briefly, for a region matched discovery, we evaluated the recall of the reported types by each caller.

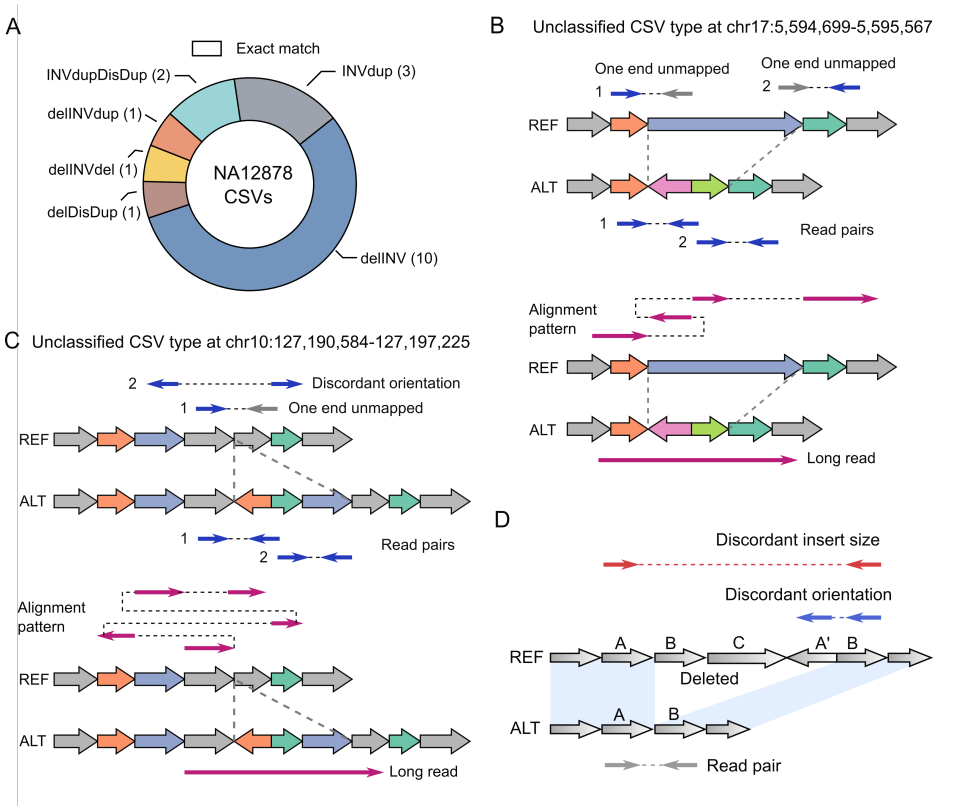


Figure 3.4: Performance of detecting complex structural variants in NA12878. (A) Performance of SVision detecting CSVs from NA12878 evaluated by exact match, where SVision detected all complex events. (B) A deleted sequence replaced with dispersed duplication and inverted duplication, which is correctly characterized by SVision. (C) SVision characterized a complex insertion, consisting of two dispersed duplications and one inverted duplication. Both (B) and (C) are labeled as NA in the published calls. The top panels of (B) and (C) are the discordant alignments derived from short-read sequencing (i.e., one end unmapped and discordant alignment). The bottom panels of (B) and (C) describe the abnormal alignment from long-read alignment. (D) Diagram of misinterpreted complex event from short-read data, while SVision correctly detected it as simple deletion.

3.3.3 CSV mediated gene structure change and genome evolution

To explore novel CSV loci and types, we further applied SVision to HG00733 (PacBio HiFi, ≈30X), where the CSVs were not well characterized. SVision detected 80 high-quality CSVs of 25 unique types, where 20 CSV graphs were novel types, accounting for half of the high-quality CSVs, and another five graphs matched reported CSV types. Moreover, 18 and 28 CSV loci overlapped genes and regulatory elements, respectively. We then introduced computational and experimental approaches to validate the structure and breakpoint junctions of the high-quality CSVs.

Firstly, the GraphAligner [34] was used to assess the internal structure and breakpoints of CSVs by aligning ONT reads [9] to SVision CSV graph. The graph alignments showed that single reads cover the entire paths of 79 CSV graphs, while one CSV graph path was covered by two different reads. Secondly, the haplotype contigs used by Phased Assembly Variant (PAV) [10] for SV discovery were used to examine the CSV internal structures. Among the 73 PAV overlapping CSVs, 90% of them could be successfully reconstructed via manual inspection, while others were challenging to characterize visually but could be verified via GraphAligner (Figure 3.5A). In addition, 20 CSVs were randomly selected for experimental validation. Specifically, eight CSVs failed PCR due to repetitive sequence or high GC content and the other 12 events were successfully confirmed by PCR and Sanger. The above validations indicated that SVision can detect and characterize CSV reliably from long-read data. Compared with long-read calls, short-reads revealed 42% of the CSV loci evaluated by region-match, where internal structures of 12% CSV loci could be completely characterized via exact-match (Figure 3.5B).

Furthermore, we noticed that 18 CSV loci overlapped genes. For instance, one CSV of novel type revealed by SVision (chr11:99,819,283-99,820,576), consisting of tandem and inverted duplications, was missed by short-read [10] and identified as a simple insertion by PAV [10] (Figure 3.5C). This CSV modified the structure of an important nervous system development gene, CNTN5, of which we identified both CSV allele and insertion allele of different frequency among populations (Figure 3.5D). We also observed the duplicated exon signature in the RNAseq data of human primary visual cortex and precuneus [86].

Additionally, SVision identified an insertion-inversion-insertion event (chr9:74,283,222-74,283,473), which was detected as a 1,737bp insertion by PAV but completely missed in previous long-read call sets [9, 87] (Figure 3.6A). This event was also re-genotyped by PanGenie, and it found 80% allele

frequency among 3,202 1KGP cohort [10]. The inserted sequence of this CSV was also identified in primate genomes (Figure 3.6B), such as gorilla, indicating the inserted state was ancestral and the reference was derived via deletion and inversion.

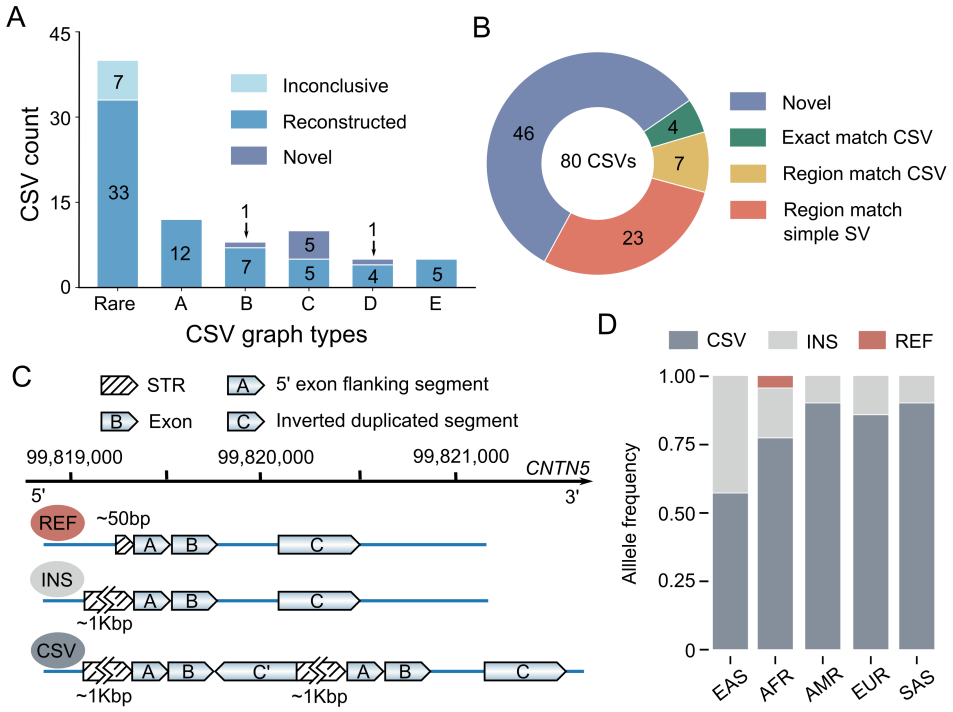


Figure 3.5: Application of SVision on HG00733 HiFi data. (A) SVision detected complex structural variants (CSVs) overlapped with Phased Assembly Variant (PAV) calls and reconstructed with HiFi haplotype contigs. The rare type represented a graph type containing less than five complex events. Graph type A, B, C, D and E corresponded to graph ID 12, 15, 23, 27 and 28, respectively. (B) Comparing SVision detected CSVs with short-read based discoveries, evaluating with region match and exact match, respectively. (C) The diagram of a novel CSV type revealed by SVision, and three allele states (i.e., REF allele, CSV allele and INS allele) were identified at this locus among the population. (D) The allele frequency of the complex event locus shown in (C).

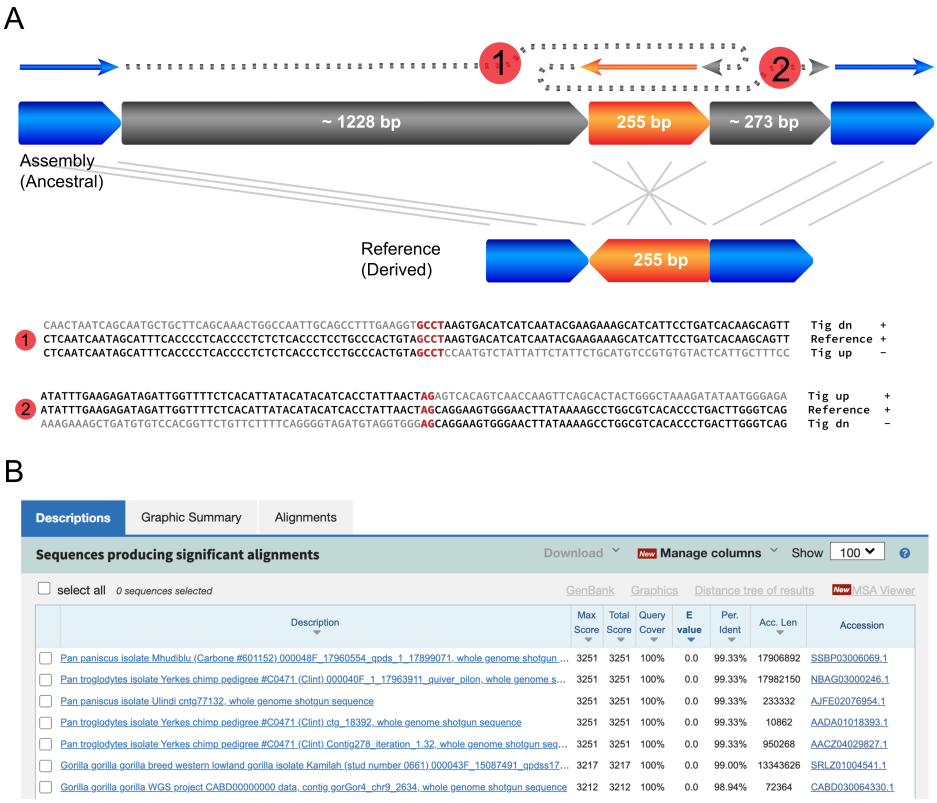


Figure 3.6: Complex structural variant revealed ancestral state. (A) The structure and breakpoint junction sequence of the variant is derived from HiFi assembly. (B) Blast results of mapping the inserted sequence to primate genomes, where the top hits include pan troglodytes and gorilla.

3.4 Conclusion

In recent years, long-read sequencing technologies have revolutionized SV detection and revealed two times more variation than short-reads [10]. While long-read SV detection tools have improved considerably in the past six years, none of them is able to correctly characterize multi-breakpoint events and thereby leaving CSVs either uncalled or misinterpreted as simple SVs. SVision fills this gap by applying a multi-object recognition framework to the denoised image to detect both simple and complex SVs, and autonomously identifies their structures without relying on predefined models. Future work will focus on tumor SV detection, especially complex events and subclonal SVs. Taken together, SVision is a valuable tool to facilitate the study of complicated and novel CSVs, paving the way for the analysis of healthy and cancer genomes in the future.

Chapter 4

SpotSV: An automated approach for simple and complex structural variants validation

Abstract In the past several years, comparing with structural variants (SVs) detection algorithms, there are a few approaches that have been developed to evaluate the quality of detected SVs. As the decrease of long-read sequencing price, accurate detection of SV breakpoints and type is critical to promote long-read applications in both clinical and research settings. However, current manually involved or experimental validation approaches is not applicable at scale in the big data era.

In this chapter, we present SpotSV, an effective algorithm that automatically validates SVs through denoised segments obtained from long-read sequencing data. SpotSV evaluates each via two major modules: 1) selection of variant overlapping reads; 2) collecting denoised segments and calculating validation score. We assessed the performance of SpotSV with both simulated and real genomes across different sequence depths. The evaluation results suggested that SpotSV is able to accurately characterize the breakpoints and type of both simple and complex SVs with low read depth. Moreover, by introducing denoised segments, SpotSV is able to assess SVs at repetitive regions as accurate as those located at simple genomic regions. Recently, long-read sequencing has been widely used in various genomic studies at scale, such as different disease and species. SpotSV provides an option to automatically and systematically assess the quality of detected SVs in high-throughput.

4.1 Introduction

Structural variants (SVs) are among the major forms of genetic variations in human genomes, affecting more than 50bp of the genomes compared with single-nucleotide-variants (SNVs) and small insertions and deletions [1, 8]. SVs comprise different subclasses, such as deletions, insertions and complex structural variant (CSV), which play important roles in numerous diseases including cancers and genetic diseases [8]. In the past decade, a large number of SV detection algorithms have developed for short-read and long-read data [41], promoting our understanding of SV functional impact as well as its role in adaptive selection in population [5]. Though long-read algorithms have been proved to outperform short-read callers in terms of sensitivity and specificity [9], some complex variant types or SVs at repetitive regions are usually misinterpreted by existing algorithms. Therefore, orthogonal or downstream SV validation methods are required to curate callsets generated by different callers, especially for clinical applications.

Currently, experimental validation through PCR and Sanger sequencing is considered as gold standard to validating detected SVs. However, experimental validation is usually time consuming, and most importantly, it is difficult to validate challenging variant classes and SVs at repeat regions. This promotes the development of a high-throughput orthogonal validation approach for detected SVs, including the breakpoint position and variant type. Nowadays, several visualization methods have been developed for researchers to manually assess the quality of detected SVs by either short-read or long-read callers. For example, Samplot [88] creates images that display the read depth and discordant alignments to validate SVs detected by short-read via a machine learning approach. In addition, given that an increasing number of CSVs have been identified, visualization methods, such as Ribbon [89], are developed to view and assess large scale complex events detected in tumor samples [90]. Note that these two representative approaches are not able to accurately characterize the breakpoint for focal complex events (i.e., event length smaller than 100kbp), which is important to understand the internal structure of complex events and their formation mechanism.

Another approach is inspired by the sequence Dotplot [84], which essentially visualizes the recurrence k-mer matrix of two sequences. Most importantly, Dotplot enables precise variant structure interpretation, including breakpoints, compared with the above-mentioned approaches. In the past decade, this approach has been widely used to investigate the genome rearrangements between different species, while it requires long sequence which is not applicable for short-read data. With the rapid development of

long-read sequencing technologies, creating a sequence Dotplot becomes a common approach to manually assess the predicted SVs, especially complex events [5]. Briefly, the alternative sequence (i.e., long-read sequencing of individual genome) is compared against the reference sequence through a fixed size sliding window, called k-mer, and the matches are plotted for visual confirmation purpose. However, this manual curation, coupled with expert-level knowledge of SV structure, are time-consuming and inefficient at large scale for high-throughput validation. VaPoR [72] is the first method that investigates and scores each SV prediction by autonomously analyzing the k-mers within a read against both an unmodified reference sequence at that loci as well as rearranged referencing pertaining to the predicted SV structure.

Moreover, it has been shown that tandem repeat regions, such as Variable Number Repeat Region (VNTR), are hotspots for SVs [87], and long-read sequencing greatly improves the detection compared with short-read sequencing, especially for insertions. Though long reads facilitate insertion detection, it is difficult for detection algorithms to characterize the internal structure of insertion that might consist of duplications. Furthermore, distinguishing insertions from duplications is critical to understand how SVs affect gene structure, thereby enabling precise analysis of functional impact. In addition, an increasing number of detected CSVs and novel CSV types [6, 12] have been reported from healthy and disease genomes, which introduces another layer of difficulty for validating SVs. Altogether, there is an urgent demand of developing novel method for validating SVs at complex genomic regions and CSVs.

Here, we present an effective sequence-based validation tool, SpotSV, that uses either long reads or assemblies to assess each predicted SV. In general, SpotSV characterizes each predicted SV by examining the denoised segments obtained from 1) SV modified sequence (PRED) against long read sequence (READ) comparison and 2) reference sequence (REF) against READ. Accordingly, a correct prediction would maximize the difference in REF-to-READ comparison, while minimize the difference in PRED-to-READ comparison. Notably, to overcome the difficulties of validating SVs at complex genomic regions, the denoised segments could be isolated by removing REF-to-REF from the PRED-to-READ and REF-to-READ because the reference context is presented in both PRED-to-READ and REF-to-READ. Afterwards, a validation score derived from denoised segments is used to assess the correctness of the predicted SV. We then evaluate the performance of SpotSV on a series of simulated and real datasets. The results suggest that our approach could accurately distinguish positive and negative predictions of

simple and complex SVs, especially SVs at repetitive regions, and it is also able to assess and refine the breakpoint of predicted SVs.

In Section 4.2, materials and related methods are described in details. Moreover, results are discussed in Section 4.3 and conclusions are drawn in Section 4.4.

4.2 Material and methods

In this section, we introduce the workflow of SpotSV and its three major components. Then, we use both simulated data and publicly available real data to assess the performance of SpotSV.

4.2.1 Overview of SpotSV

SVs modify the reference sequence (REF) based on detected type and breakpoint position, thus the modified sequence, referring to as predicted sequence (PRED), is identical to long reads (READ). Accordingly, we define SV validation as a problem of maximizing the differences of READ and REF sequence, while minimizing the differences of READ and PRED. SpotSV is developed to assess each SV with three major steps (Figure 4.1): (i) creating k-mer recurrence matrices for REF against READ and PRED against READ; (ii) collecting denoised segments from REF-to-READ k-mer matrix and PRED-to-READ k-mer matrix separately; (iii) calculating SV validation score and assessing breakpoints. Specifically, a k-mer recurrence matrix is created by sliding a fixed-size substring (k-mer) with single steps through each sequence to mark positions where two sequences are identical.

Given the k-mer recurrence matrix, SpotSV removes identical sequence substrings that appeared in the same position on the reference sequence, resulting in so-called REF-to-READ and PRED-to-READ k-mer recurrence matrices. Then, SpotSV obtains denoised segments from REF-to-READ and PRED-to-READ k-mer recurrence matrices for assessing the quality of predicted SVs. The denoised segments enable accurate characterization of SVs at repetitive regions as well as CSVs. Finally, SpotSV adds validation score and refined breakpoints for each predicted SV in a Variant Call Format (VCF) file. Moreover, SpotSV provides REF-to-READ Dotplots and denoised REF-to-READ Dotplots based on the k-mer recurrence matrix for visual confirmation.

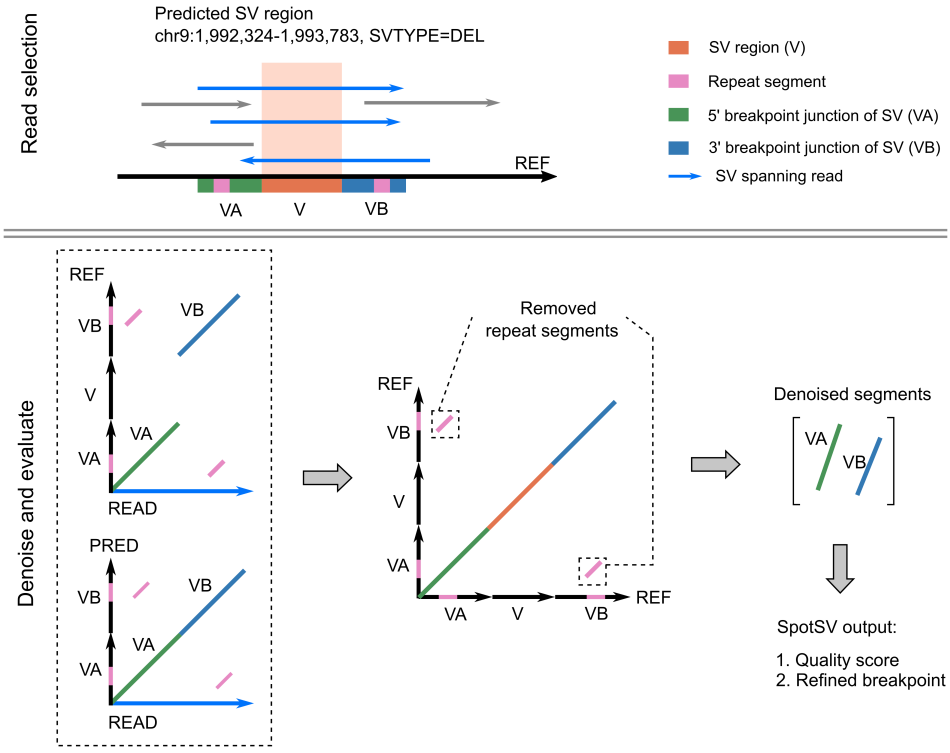


Figure 4.1: Overview of SpotSV. SpotSV consists of two major modules: 1) Read selection and 2) Denoise and evaluate. Module 1) is designed to select variant overlapping reads, containing reads across entire events and those only covering the breakpoint junctions. Module 2) consists of two steps. Firstly, selected reads are realigned and denoised to obtain denoised segments. Secondly, SpotSV uses denoised segments to assess the quality of detected SVs.

4.2.2 Modify reference sequence with predicted structural variants

SpotSV uses predicted SV type and genomic position to modify the reference sequence at the predicted locus, which is referred to as predicted sequence (PRED). Specifically, given predicted SV breakpoints $[\ell, r]$ and size len, SpotSV extracts the segment between $[\ell - 1000, r + 1000]$ from the reference genome to obtain the reference sequence (REF). Then, the segment between $[1000, 1000+len]$ from REF is modified to create PRED based on predicted SV type and length. The above process is applied to SVs containing more than two breakpoints on reference genome, including deletion, inversion, duplication and other complex SV types. For example, if a deletion of size 1,000bp is detected at $[20000, 21000]$, its corresponding REF is extracted between $[19000, 22000]$ from the reference genome and PREF sequence is obtained by deleting the sequence from 1000 to 2000 in the REF. To modify the reference genome containing duplications, especially dispersed duplications, SpotSV uses left most position ℓ as source position, from which the sequence of length len is copied and inserted to the rightmost position r, the destination position. For insertion with a single breakpoint on the reference genome, SpotSV extracts REF from $p - 1000$ to $p + 1000$ on reference genome and obtains PRED by inserting the sequence of size len at position 1000 on REF. The REF and PRED sequences are then used to create REF-to-READ and PRED-to-READ k-mer recurrence matrices, respectively.

4.2.3 Generating denoised segments based on k-mers

SpotSV identifies cooccurrence of substrings (k-mers) in two sequences and generates a raw REF-to-READ and PRED-to-READ k-mer recurrence matrix, which is visualized as sequence Dotplot in SpotSV outputs. By default, SpotSV uses k-mers of length 31bp and requires an exact match between sequences by comparing consecutive k-mers. Once encountering an unmatched k-mer, SpotSV generates a segment of length $k + n$ consisting of n matched k-mers, where k is the length of the k-mer. To resolve repetitive regions, SpotSV introduces a novel process to isolate and boost the SV signature by removing reference background. Firstly, SpotSV uses REF to create a k-mer recurrence matrix representing reference context, from which a set of repeated segments and their position on the reference genome is obtained. Secondly, SpotSV traverses all segments obtained from raw REF-to-READ according to the segment positions on the reference genome, and remove segments that have been identified as repeated segments in reference

sequence comparison. For two identical sequences, the k-mer recurrence matrix only has values on main diagonal, while repeat sequences add values to other cells in the matrix. Compared with repeat sequences, SVs break the continuity of the values on the main diagonal at predicted breakpoint position, and move values right after a breakpoint position to either horizontal axis or vertical axis direction by SV length. For example, if vertical axis and horizontal axis of a recurrence matrix indicate the reference sequence and read sequence, respectively, a deletion manipulates the recurrence matrix by shifting the values along the vertical axis by length L. It should be noted that segments on the main diagonal at the 5' breakpoint position flanking regions and segments on the 3' breakpoint shifted by SV length are retained during repeats removal. This repeat elimination process is applied to each read spanning predicted SV, from which denoised segments are obtained for further assessment. Since DNA is double stranded, containing forward and minus strand, the above process is also applied to the reverse complementary sequence to find potential matches on the minus strand, enabling validation of inversions. In addition, denoised segments in READ-to-REF are used to determine breakpoints of a predicted SV. Finally, denoised segments are also used to create a Dotplot in SpotSV outputs for visual confirmation.

4.2.4 Calculating structural variant validation score

Given a denoised segment set, the difference of two sequences could be measured by calculating distance between segments and diagonal. In principle, distance would approach zero when measuring two identical sequences, while SVs alter the sequence and thus would produce large distance. Specifically, assuming a predicted SV s is spanned by m reads, for a read i containing n denoised segments, the distance d of denoised segment j is defined as vertical distance to diagonal, which is calculated as:

$$\begin{aligned} d_{s,i,j} &= \frac{1}{3}((x_{s,i,j,start} - y_{s,i,j,start}) \\ &\quad +(x_{s,i,j,mid} - y_{s,i,j,mid}) + (x_{s,i,j,end} - y_{s,i,j,end})) \end{aligned}$$

Here $x_{s,i,j,start}$ and $y_{s,i,j,start}$ are the start position of segment j on x-axis and y-axis, respectively, $x_{s,i,j,mid}$ and $y_{s,i,j,mid}$ are the middle position of segment j on x-axis and y-axis, respectively, while $x_{s,i,j,end}$ and $y_{s,i,j,end}$ are the end position of segment j on x-axis and y-axis, respectively. Then, the average distance of all segments belonging to a read is calculated as:

$$d_{s,i,avg} = \frac{1}{n}\sum_{j=1}^{n} d_{s,i,j}$$

Since correct SV prediction maximizes difference of REF-to-READ and minimizes difference of PRED-to-READ, the SV validation score is comprised of two parts. The average distance of REF-to-READ is calculated as $d_{s,i,avg,ref} \in [0, +\infty)$. Similar to $d_{s,i,avg,ref}$, we define the average distance of PRED-to-READ as $d_{s,i,avg,predict} \in [0, +\infty)$. Then, SpotSV normalizes these two scores to assess the predicted SV:

$$Score_{s,i} = \begin{cases} 1 - d_{s,i,avg,predict}/d_{s,i,avg,ref} & \text{if } d_{s,i,avg,ref} > 0 \\ 0 & \text{otherwise} \end{cases}$$

Moreover, for $Score_{s,i} < 0$, it is set to $Score_{s,i} = 0$, thus $Score_{s,i} \in [0, 1]$. Read i is not supporting the predicted SV if $Score_{s,i} = 0$, while $Score_{s,i} = 1$ indicates read i supports the predicted SV. Finally, for a predicted SV spanned by m reads, SpotSV uses the highest score as final validation score in the output:

$$Score_{s,highest} = \max([Score_{s,1}, \ldots, Score_{s,i}, \ldots, Score_{s,m}])$$

However, due to sequencing errors, we consider that read i supports a predicted SV if $Score_{s,i} > Score_{\text{threshold}}$, where $Score_{\text{threshold}} = 0.8$, from which SpotSV identifies the number of reads that support SV s and estimates the gentype.

4.2.5 Data availability

Using the same simulation workflow as described in Chapter 2 and Chapter 3, non-overlapping simple deletions, inversions, insertions and duplications as well as five CSV types are independently incorporate into GRCh38 in both heterozygous and homozygous states. Notably, four subtypes of duplications are simulated, including tandem duplication (tDUP), inverted tandem duplication (itDUP), dispersed duplication (dDUP) and inverted dispersed duplication (idDUP), where itDUP and idDUP are classified as complex event according to previous studies. Moreover, we include another three well-characterized types from previous studies, i.e., deletion associated with insertion (Del-Inv), deletion associated with dispersed duplication (Del-dDUP) and deletion associated with inverted dispersed duplications (Del-idDUP). In total, we simulate 20,000 SV events at whole genome scale, and the number of events is equally distributed for the simulated SV types. The number of SVs for each chromosome (from chromosome 1 to chromosome X) is selected based on the ratio of chromosome length. The 20,000 simulated SVs are kept in BED format and used as positive cases for performance

evaluation, while another 1,000 negative cases not overlapping with the positive ones are added to the benchmark BED file, making a benchmark that contains 20,000 positive and 1,000 negative cases. The types of negative cases are randomly assigned based on simulated SV types. It should be noted that the 1,000 negative cases are not implanted into the simulated genome containing 20,000 positive cases, thus negative cases should be validated as false prediction. The simulated genome is further sequenced to different HiFi read depth, ranging from 5X to 30X, with default parameter specified in VISOR [83]. The HiFi reads are aligned to the reference GRCh38 with pbmm2 (https://github.com/PacificBiosciences/pbmm2) default settings.

For the real dataset, both the HiFi and ONT data for HG002 are obtained from ftp://ftp.ncbi.nlm.nih.gov/giab/ftp/data/AshkenazimTrio/HG002_NA24385_son, which were initially sequenced by the Genome-In-A-Bottle (GIAB) and the high-quality benchmark for HG002 used in this chapter is obtained from ftp://ftptrace.ncbi.nlm.nih.gov/giab/ftp/data/AshkenazimTrio/analysis/NIST_SVs_Integration_v0.6/. We use both HiFi and ONT data to compare SpotSV with VaPoR on validating SVs in the benchmark. Since VaPoR is not able to run on chromosome 4 of real data due to a coding error, we only examine the performance on other autosomes as well as sex chromosomes.

4.3 Results

In this section, we first evaluate SpotSV on validating simulated data that contains both simple and complex SVs. Then, using the high-quality benchmark set of HG002, we compare the performance of SpotSV and VaPoR by assessing the number of correctly validated SVs.

4.3.1 Evaluating SpotSV with simulated data

We first examined the impact of aligners on SpotSV, where SpotSV was applied to simulated reads aligned by pbmm2, minimap2 [17] and ngmlr [18], respectively. The results showed that percentage of SpotSV validated SVs was independent of aligners, such as 97.91%, 97.40%, 97.39% of SVs were validated on pbmm2, minimap2 and ngmlr aligned data at validation score cutoff 0.9, respectively (Figure 4.2A). We then investigated the performance of SpotSV on pbmm2 aligned simulation data. Since the simulated dataset contained 20,000 positive events (Table 4.1), it was expected that the majority of SpotSV validation scores ranged from 0.8 to 1 for both homozygous and heterozygous events across different coverages (Figure 4.2B). Using a high

validation score 0.9 as cutoff, SpotSV was able to successfully validate 85% of SVs even with 5X low-coverage data, and 95% SVs could be validated with a validation score cutoff 0.8 (Figure 4.2C). Additionally, we identified 336 simulated SVs at repetitive regions and examined the sensitivity of validation for these SVs. By introducing denoised segments, the average sensitivity difference of validating SVs inside and outside repeat regions was around 2% across different coverages at a validation score cutoff of 0.8. For example, applying SpotSV on 20X coverage data, 93% and 95% of SVs inside and outside repeat regions were validated, respectively (Figure 4.2). Moreover, SpotSV could validate heterozygous SVs located at repetitive regions as sensitive as homozygous SVs.

	DEL	INS	INV	tDUP	itDup	dDUP	idDUP	DEL+ INV	DEL+ dDUP	DEL+ idDUP
chr1	164	164	164	164	164	164	164	164	164	163
chr2	160	160	160	160	160	160	160	160	160	159
chr3	131	131	131	131	131	131	131	131	131	130
chr4	126	126	126	126	126	126	126	126	126	125
chr5	120	120	120	120	120	120	120	120	120	119
chr6	113	113	113	113	113	113	113	113	113	112
chr7	105	105	105	105	105	105	105	105	105	104
chr8	96	96	96	96	96	96	96	96	96	95
chr9	91	91	91	91	91	91	91	91	91	90
chr10	88	88	88	88	88	88	88	88	88	87
chr11	89	89	89	89	89	89	89	89	89	88
chr12	88	88	88	88	88	88	88	88	88	87
chr13	76	76	76	76	76	76	76	76	76	75
chr14	71	71	71	71	71	71	71	71	71	70
chr15	67	67	67	67	67	67	67	67	67	66
chr16	60	60	60	60	60	60	60	60	60	59
chr17	55	55	55	55	55	55	55	55	55	54
chr18	53	53	53	53	53	53	53	53	53	52
chr19	39	39	39	39	39	39	39	39	39	38
chr20	42	42	42	42	42	42	42	42	42	41
chr21	31	31	31	31	31	31	31	31	31	31
chr22	34	34	34	34	34	34	34	34	34	34
chrX	103	103	103	103	103	103	103	103	103	103

Table 4.1: Number of simulated structural variants at different chromosomes.

We further assessed the true positive rate and false positive rate at different validation score cutoffs for all simulated events. The results showed

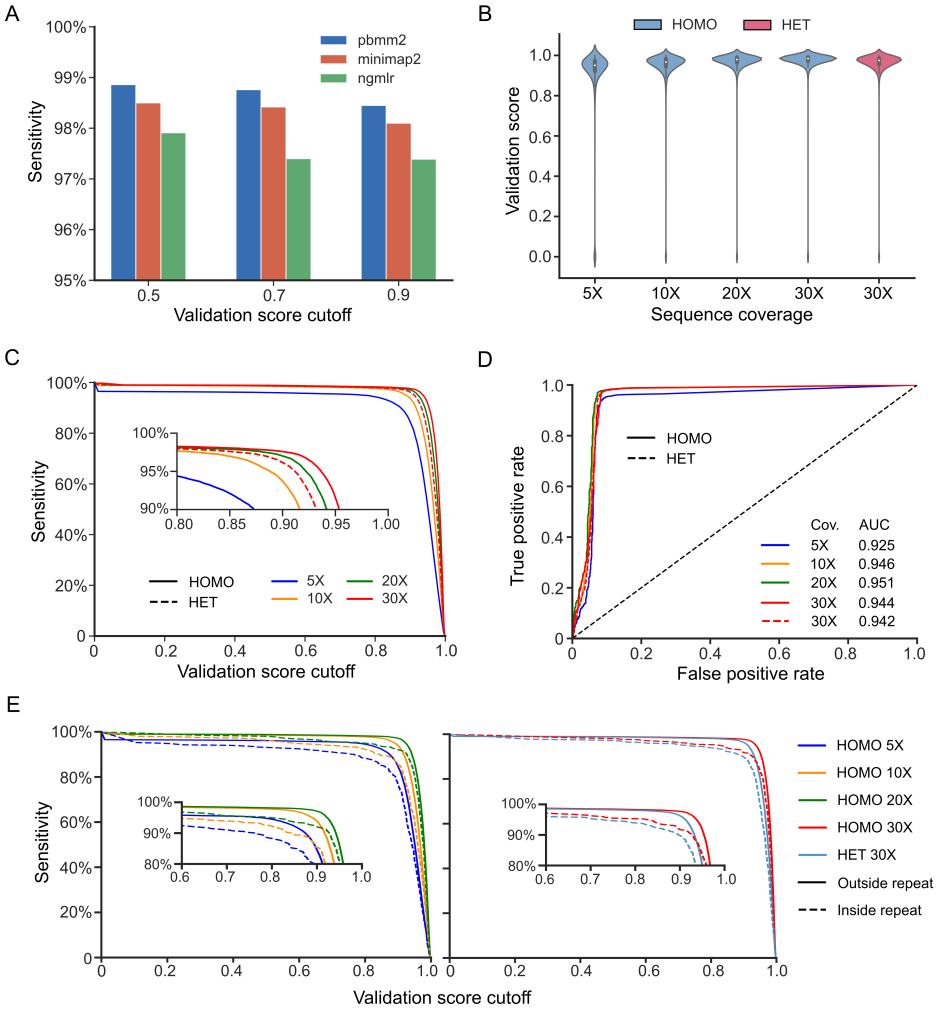


Figure 4.2: Performance of validating simulated structural variants across different coverages. (A) Sensitivity of validating simulated structural variants (SVs) at different validation score cutoffs using long reads mapped with different aligners. (B) The distribution of validation score of homozygous and heterozygous SVs at different sequence coverages. (C) The sensitivity of validation simulated SVs at different validation score cutoffs across different coverages. (D) The true positive rate and false positive rate of validating simulated SVs at different validation score cutoffs. (E) The sensitivity of validating SVs inside and outside of repetitive regions.

that the AUC (Area Under Curve) was 0.92 for homozygous SVs while using 5X low-coverage data, and it increased to 0.94 for 20X coverage data (Figure 4.2D), which was evaluated as optimal coverage for efficient and effective SV detection [91].

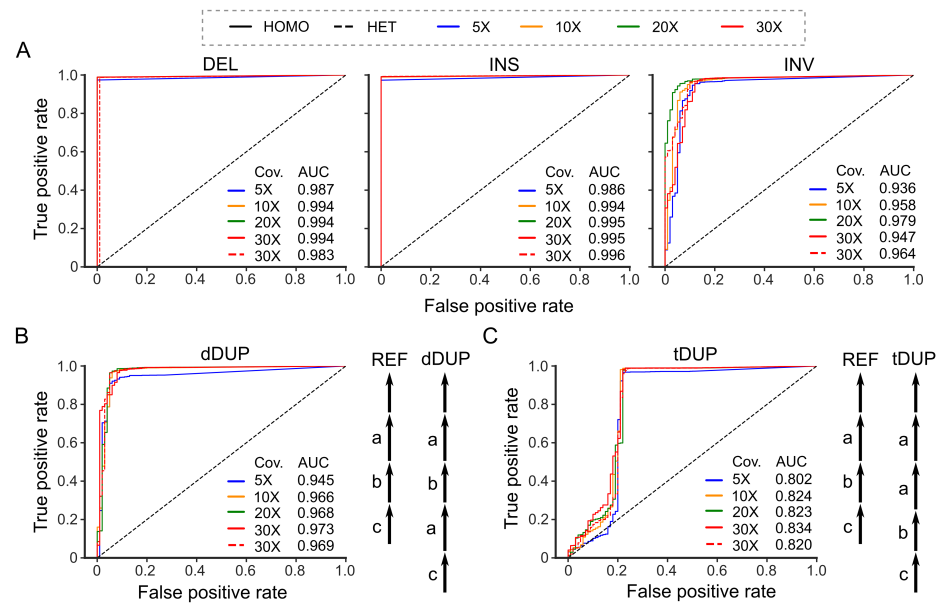


Figure 4.3: Receiver operating characteristic curve of validating five simple structural variant types. (A) The true positive rate and false positive rate of validating deletion (DEL), insertion (INS) and inversion (INV) across different coverages. (B) The true positive rate and false positive rate of validating dispersed duplication (dDUP) and tandem duplication (tDUP) across different coverages.

We then examined the performance of validating SVs of different types. For homozygous SV of different types, even using 5X coverage data, AUC of SpotSV could reach 0.98, 0.98 and 0.93 for validating deletion, insertion and inversion, respectively (Figure 4.3A). Duplication was a special form of insertion, where the inserted sequence either originated from the segment adjacent to the insertional breakpoint or from a remote position, forming so-called tandem duplication and dispersed duplication. It was usually challenging to distinguish insertions from duplications as well as to identify tandem and dispersed duplications for existing callers. SpotSV was able to

correctly validate tandem duplications and dispersed duplications in high AUC, i.e., 0.80 and 0.96, respectively, making it a valuable method to curate duplications (Figure 4.3B, Figure 4.3C). In terms of homozygous complex SVs of five types, the average AUC was 0.91 while applied to 30X coverage data, and the highest AUC of five types was 0.99 for validating deletion associated inversions (Figure 4.4). We also observed that there were no significant changes of AUC for validating heterozygous simple and complex SVs at 30X coverage data.

Altogether, the above results indicate that SpotSV could accurately validate both simple and complex SV types even with 5X coverage data.

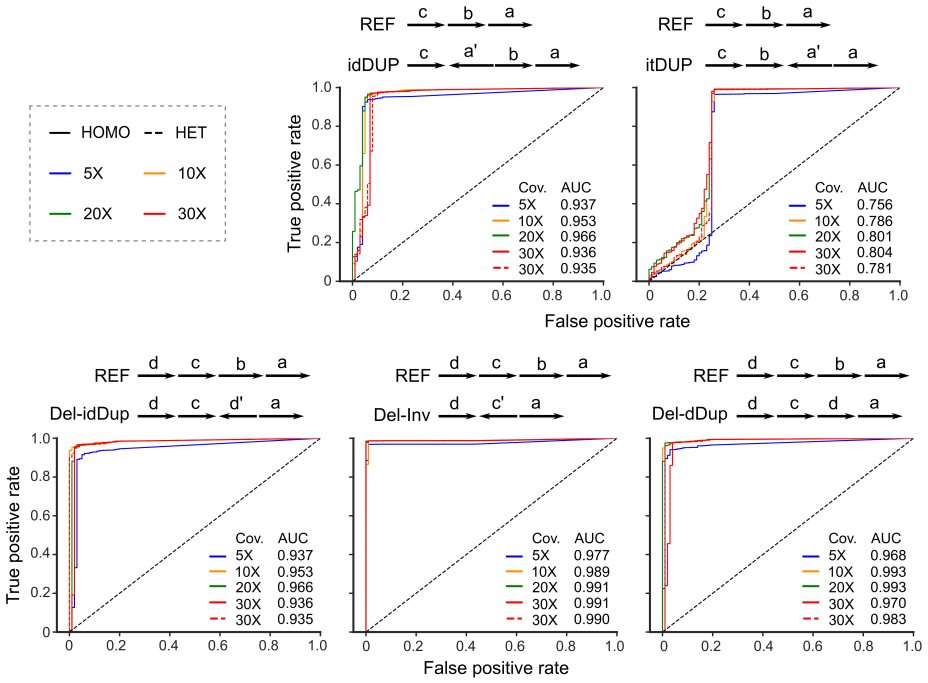


Figure 4.4: Receiver operating characteristic curve of validating five complex structural variant types. idDUP: inverted dispersed duplication, itDUP: inverted tandem duplication, Del-idDup: deletion associated with inverted dispersed duplication, Del-Inv: deletion associated with inversion, Del-dDup: deletion associated with dispersed duplication.

4.3.2 Validating structural variants in a well-characterized genome

We next compared the sensitivity of SpotSV and VaPoR using high-confident SVs in HG002 released by the Genome in a Bottle (GIAB) Consortium [92]. The HG002 callset contains 14,588 deletions and 15,432 insertions, and each deletion or inversion is assigned to a different 'RETYPE' according to sequence features at variant loci (Table 4.2). For example, a deletion (DEL) is defined as 'SIMPLEDEL' if this variant deleted an unique sequence, otherwise it is defined as 'CONTRACT', indicating deletion of a sequence entirely similar to the remaining sequence. We evaluate the sensitivity of validating all 30,020 SVs using HiFi and ONT data at different sequence coverages. As a result, SpotSV was able to examine 96% and 98% of SVs when applied to 5X HiFi and ONT data, respectively, and other SVs were not able to be assessed due to lack of variant spanning reads (Figure 4.5A). While using high-coverage HiFi and ONT data, 99% of SVs could be examined by SpotSV. Comparably, VaPoR was able to assess around 40% SVs and others were labeled as 'NA' while using ONT data and low coverage HiFi data (Figure 4.5A). For SVs that could be assessed by SpotSV and VaPoR, we investigated the sensitivity under various validation score cutoffs. Though sensitivity was negatively correlated with validation score cutoff, SpotSV consistently outperformed VaPoR across different coverages and validation score cutoffs (Figure 4.5B). The performance was especially prominent for ONT data, where SpotSV correctly validated 40% more SVs that VaPoR (Figure 4.5B).

SVTYPE	REPTYPE						
	SIMPLE-DEL	SIMPLE-INS	SUBS-DEL	SUBS-INS	DUP	CON-TRACT	SUM
DEL	8334	0	976	2	4	5171	14588
INS	209	7008	69	1243	6849	53	15432

Table 4.2: Number of structural variants in the HG002 benchmark set.

Moreover, we noticed a significant sensitivity decrease of VaPoR and SpotSV at a validation score around 0.1, while the sensitivity of VaPoR also decreased significantly at a validation score around 0.5 across different coverages and platforms (Figure 4.5B). We then examined the performance of validating 927 DEL and 921 INS events that are located at highly repetitive regions from HG002 benchmark. The results show that SpotSV was able

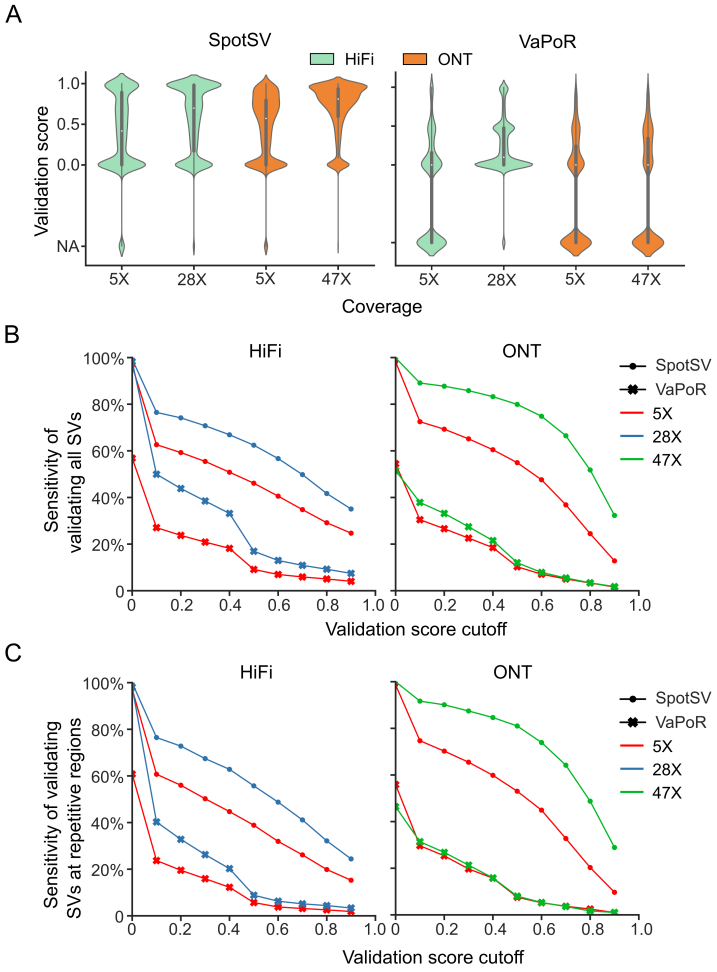


Figure 4.5: Performance of validating structural variants in HG002. (A) The distribution of validation score assessed by SpotSV and VaPoR for all structural variants (SVs) in the HG002 benchmark. NA indicates SVs that coule not be assessed. (B) The sensitivity of validating all SVs in HG002 using different validation score cutoffs. (C) The sensitivity of validating SVs at repetitive regions using different validation score cutoffs.

to validate SVs at highly repetitive regions as sensitive as those outside of repeats, while the average sensitivity decrease for VaPoR was around 10% when applied to SVs located at repetitive regions (Figure 4.5C). For example, a deletion at a highly repetitive region of validation score 1.0 was correctly validated by SpotSV because SpotSV used the denoised segment for validation (Figure 4.6A), while VaPoR assigned a validation score of 0.3 (Figure 4.6B).

Furthermore, we found VaPoR was not able to assess two adjacent SVs, while SpotSV not only validate this event but also identifies an extra SV breakpoint (Figure 4.6C). Our results demonstrated that SpotSV was able to effectively validate SVs at genomic regions of different complexity, especially for tandem repeat regions.

4.3.3 Structural variant breakpoint validation and accuracy

One of the challenges of SV discovery is the precise determination of breakpoint positions at single nucleotide resolution. Some of the previous short-read algorithms, such as Pindel [36] and Manta [42], could detect single nucleotide resolution breakpoints, but their SV detection capability was limited by the read length and repetitive elements. Moreover, a recent study conducted by the 1000 Genomes Project (1KGP) reported that the median confidence interval of breakpoints identified by short-read callers was ±85bp across all events [5]. We therefore assessed whether SpotSV was able to identity accurate breakpoints by using simulated SVs and SVs from the HG002 benchmark set. Briefly, breakpoints of HG002 SVs were used as ground truth breakpoints, which were only compared to SpotSV identified breakpoints because validated breakpoints were not included in VaPoR outputs.

The results showed that most of SpotSV identified breakpoints were ±200bp apart from breakpoints of ground truth calls, with a small portion of breakpoint offset ranging from 200bp to 500bp (Figure 4.7A). Though distribution of breakpoints identified from 5X ONT data was flattened compared with 5X HiFi data, high coverage ONT data facilitated accurate breakpoint detection of SpotSV, leading to similar results compared to 27X HiFi data (Figure 4.7A). In addition, we assessed the breakpoint accuracy of SVs at genomic regions of different complexity. Specifically, 'DEL-SIMPLEDEL' and 'INS-SIMPLEINS' were SVs identified at simple genomic regions, while DEL and INS classified as other 'REPTYPE' were considered at complex regions, referred to as 'DEL-Complex' and 'INS-Complex'. By comparing breakpoint offsets of these two groups of calls, we found that SpotSV was able to identify breakpoints of SVs at complex genomic regions as accurate

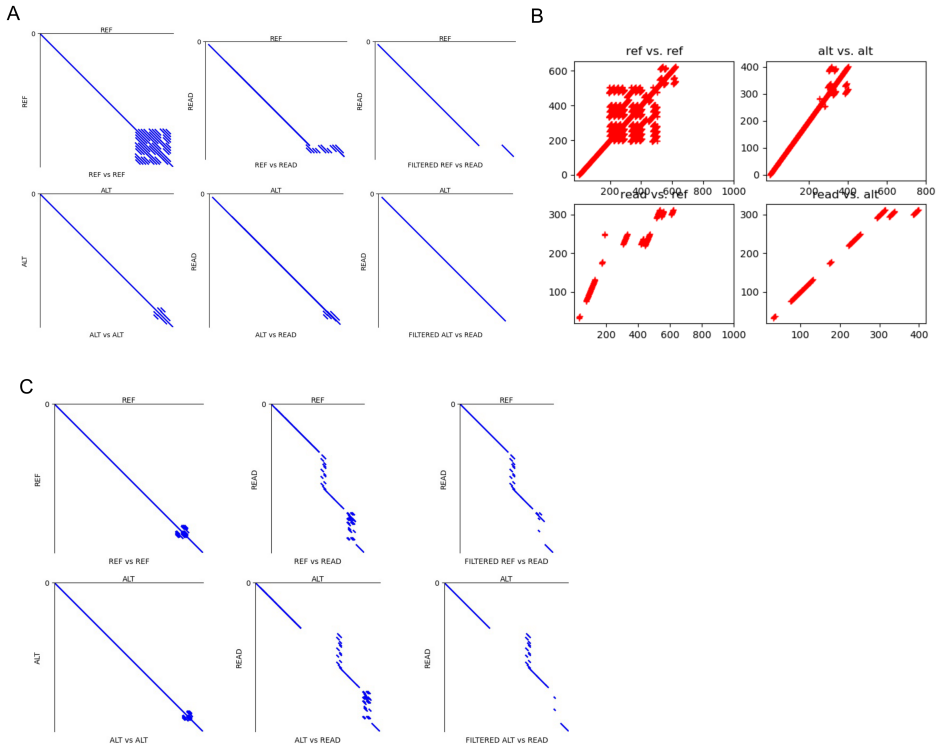


Figure 4.6: Examples of SpotSV validated SVs. (A) SpotSV validates a deletion at a tandem repeat region of validation score 1.0, while VaPoR (B) calculates a validation score of 0.3 for this event. (C) SpotSV identifies a variant locus containing two insertions, but this event was labeled as 'NA' by VaPoR.

as those at simple genomic regions (Figure 4.7B).

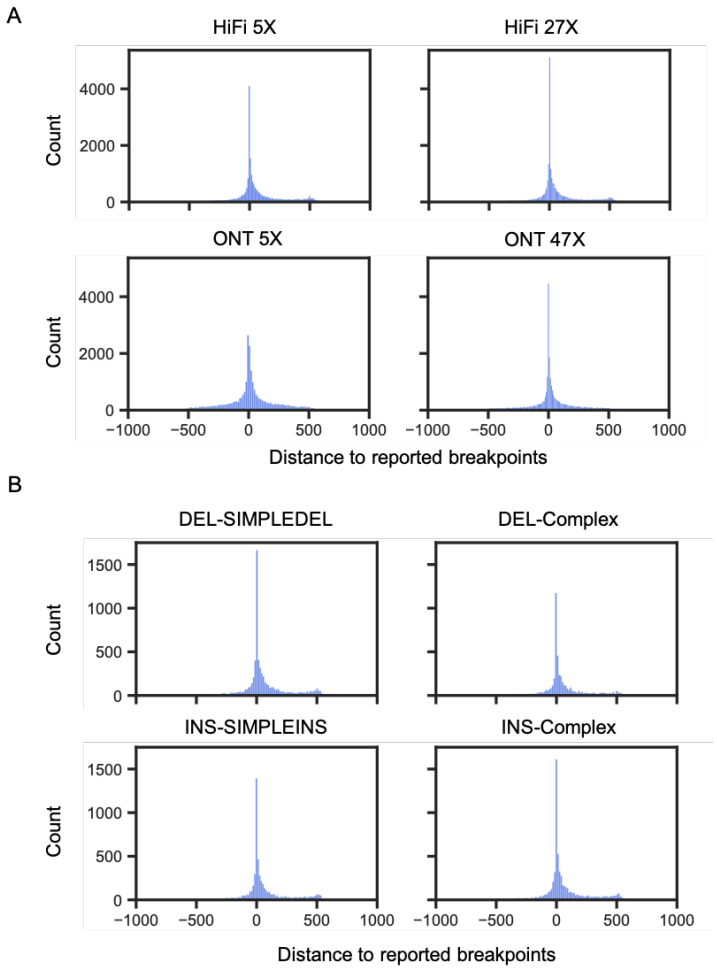


Figure 4.7: Breakpoint accuracy of SpotSV on HG002 calls. (A) Overall breakpoint offsets evaluated on HiFi and ONT data. (B) The distance of benchmark breakpoints to breakpoint identified by SpotSV from 27X HiFi data. The breakpoint comparison is grouped by the complexity of variant loci. Specifically, 'INS-SIMPLEINS' and 'DEL-SIMPLEDEL' are considered as variant occurred at simple genomic region, while 'INS-Complex' and 'DEL-Complex' are labeled as other 'REPTYPE' instead of 'SIMPLEDEL' or 'SIMPLEINS'.

4.4 Conclusion

In this chapter, we presented an automated simple and complex SV assessment approach based on denoised segments, named SpotSV, for validating predicted SVs using long-read sequencing data. SpotSV obtains denoised segments by subtracting reference context from predicted sequences modified with the profile of SVs, thereby reducing the impact of repeat sequences on SV validation that are usually inaccessible by existing methods. Moreover, SpotSV implements the functions to discriminate several subclasses of duplications from insertions, such as tandem and dispersed duplications, which are particular challenging to validate and important for functional analysis. The performance assessed on simulated and real data suggests that SpotSV can accurately validate SVs inside and outside of repetitive regions, with the capability of discriminating genomic loci containing incorrect discoveries or correct detection with inaccurate SV profiles (i.e., type and breakpoints). Future work will focus on optimizing local sequence realignment, especially for detected SV loci containing multiple breakpoints.

Recently, genome assembly based on long-reads has become a popular approach for genomic study, and SV validation from reads is an important orthogonal approach to assess SVs detected from assemblies of different species. Moreover, as the long-read sequencing price decreases, there is an urgent need of assessing SVs from clinical perspectives. Therefore, SpotSV is a valuable method that enables efficient SV assessment for different genomic studies.

Chapter 5

Assessing reproducibility of long-read structural variant detection algorithms

Abstract Recent advances in long-read sequencing and haplotype-aware assemble have enabled phased structural variants (SV) detection and improved SV detection at complex genomic regions. The assembly-based approach for tumor SV detection is further complicated due to heterogeneous cell populations and polyploid tumor genomes. Though a number of alignment-based methods that are more robust to complex tumor genomes have been developed, they lacked systematic evaluation of reproducibility, especially at complex genomic regions, which is critical for promoting long-read application in clinical practices. In this study, we benchmark six alignment-based methods on four real datasets produced by PacBio and Oxford Nanopore sequencers for recall, precision, SV breakpoints and type consistency as well as capability of detecting SVs at repetitive regions. Our results first highlight the important role of aligners in determining SV breakpoint concordance of detection algorithms. Secondly, our analysis based on phased assembly reveals that tandem repeat regions are hotspots for discordant calls of each algorithm detected from different aligners and platforms combinations. In addition, the analysis of tumor-normal paired samples suggest that the number of different SV types varies from tumor unique calls identified from each caller, and integration of tumor unique calls from each caller would substantially improve somatic SV detection. As the importance of SVs are increasingly recognized in disease genomes, our analysis provides important guidelines for selecting dataset, aligner and algorithms for efficient SV detection, and reveals valuable hints for future algorithm development, thereby shedding light on cutting-edge genomic studies and clinical applications.

5.1 Introduction

Structural variants (SVs) comprise different subclasses that consist of unbalanced copy number variants, including deletion, duplication and insertion, as well as balanced rearrangements, such as inversion and translocation [8]. SVs could also have complex internal structures, consisting of multiple combinations of the above-mentioned simple forms of SVs, and this complex form of SV is referred to as complex SV (CSV) [11, 12, 57]. In the past decade, researchers have made great progress in discovering and genotyping SVs in diverse populations and generated phased reference panels of SVs with short-read data. Moreover, researchers found that SVs are enriched for expression quantitative trait loci (eQTLs) up to 50-fold compared with single nucleotide variations, indicating the important role of SVs in regulating gene expression. Remarkably, the widespread application of single-molecule sequencing (SMS) technologies, including Pacific Bioscience (PacBio) and Oxford Nanopore Technology (ONT), greatly improves the sensitivity and precision of detecting SVs comparing with short-read [9, 41]. A study revealed that PacBio long-reads were approximately three times more sensitive than a short-read ensemble achieved, and a large set of SVs, ranging from 50 to 2000bp were unresolvable without long reads [8]. Recently, the haplotype-aware phased assembly facilitated the direct detection of phased SVs [9, 10], enabling systematic analysis of functional impact of SVs as well as SV candidates for adaptive selection within the human population.

Moreover, long-read sequencing also facilitates the analysis and manual curation of CSVs that are usually inaccessible via short-read data. For instance, in 2015, the 1000 Genomes Project (1KGP) published the first previously unexplored CSV classes by integrating both short- and long-read sequencing. Additionally, long-read sequencing revealed SVs in genetic diseases [93, 94, 95] and cancers [45, 90, 96, 97, 98, 99, 100] that are usually undetectable via short-read data. For instance, the ONT data reveals 10,000bp Alzheimer's disease associated ABCA7 Variable Number Tandem Repeats (VNTR) expansion [101] and the PacBio long-read data reveals 10 times more SVs than that of short-read in breast cancer. Additionally, the somatic SVs in tumor are a valuable genetic source to understand tumorigenesis, such as a study showed that long reads could detect two times more somatic SVs than previous short-read study [82].

Detecting SVs from SMS data usually consists of two steps. Firstly, the variant signatures are identified and gathered from two types of aberrant alignments: intra-read and inter-read. Intra-read alignments are derived from reads spanning the entire SV locus, resulting in deletion and insertion

signatures. Inter-read alignments are usually obtained from the supplementary alignments and SV signatures that could be identified from inconsistencies in orientation, location and size during mapping, analogous to read-pair signatures, from which translocation as well as large deletion, duplication and inversion signatures are identified. Secondly, callers typically cluster and merge similar signatures from multiple aberrant alignments, and delineate proximal signatures that support putative SV. Nearly all alignment-based algorithms developed in the past five years, such as Sniffles [18], pbsv, CuteSV [102], SVIM [103], NanoVar [104], NanoSV [105] and Picky [96], detect SVs through combinations of signatures obtained from inter-read and intra-read alignments but differ in their signature clustering heuristics. For example, Sniffles evaluates the signature similarities by examining the signature position and size, and additionally clusters SV supported by the same set of alignments to detect nested SVs. Some methods, such as Phased Assembly Variant (PAV) and SVIM-ASM [103] use the alignment of whole genome assembled contigs as input, referred to as assembly-based approaches, from which aberrant inter-contig and intra-contig alignments are used for SV detection.

Moreover, somatic SVs are driver events for tumorigenesis and they are usually detected by identifying SVs present in tumor but absent from its matched normal sample. For instance, CAMPHOR [82], a computational pipeline, detects somatic SVs by removing SVs present in a 'normal panel'. A similar process can also be completed by SURVIVOR, which identifies putatively somatic SVs that are only present in tumor [90]. However, affected by repetitive sequences and human reference genome defects [87], intensive breakpoint filtering and an external normal reference SV set are required to obtain high-quality somatic SVs [106, 107].

Previous studies have estimated that at least 30% of cancers have a known pathogenic SVs used in diagnosis or treatment [108], and germline variants in cancer predisposition genes underline 5–10% of all cancers [109, 110, 111]. However, the prevalence of SVs in cancer is likely underestimated due to low sensitivity and specificity for short-read based SV discovery at regions of repetitive elements, low sequence complexity and strong GC bias. Recently, long-read assembly approach significantly increased the sensitivity of detecting SVs at complex genomic regions compared to that of short-read data [9, 10], but precise detection of germline SVs and distinguishing tumor unique SVs from germline is further complicated due to tumor heterogeneity and polyploidy. Compared with assembly approaches, alignment-based detection methods are more robust to amplificated tumor genomes that originate from mixed cell populations, while inconsistencies in breakpoints and variant

types confound tumor SV detection, especially somatic SV. Therefore, it is critical to assess the detection consistency of alignment-based algorithms, especially at complex genomic regions, thereby enabling accurate and comprehensive germline and somatic SV detection. In this study, using multiple datasets of two platforms (i.e., HiFi and ONT) mapped by two aligners (i.e., minimap2 and ngmlr), we evaluated the recall, precision, variant breakpoints and type consistency of five alignment-based SV detection algorithms and assess the alignment-based algorithms for tumor SV detection.

In Section 5.2, materials and related methods are described in details. Moreover, results are discussed in Section 5.3 and conclusions are drawn in Section 5.4.

5.2 Materials and methods

In this section, we introduce the datasets and methods used in the evaluation.

5.2.1 Read mapping and SV detection

In this chapter, HiFi and ONT data are obtained for HG002, NA19240, HG00733 and HG00514, while ONT data was used for tumor-normal paired sample COLO829. Then, minimap2 [17] (v2.20) and ngmlr [18] (v0.2.7) were used to map the long-read data of HG002 and COLO829 to hg19 due to the reference version of the benchmark set. The long-read data of NA19240, HG00733 and HG00514 were mapped to reference version GRCh38. For minimap2, parameters '`-a -H -k 19 -O 5,56 -E 4,1 -A 2 -B 5 -z 400,50 -r 2000 -g 5000`' were applied to align HiFi reads, while '`-a -z 600,200 -x map-ont`' were used for ONT reads. For ngmlr, parameters '`-x pacbio`' and '`-x ont`' were used to align HiFi and ONT reads, respectively. For the detection algorithms, SVision (v1.3.6), CuteSV (v1.0.10), pbsv (v2.2.2), SVIM (v1.4.0), Sniffles (v1.0.12) and NanoVar (v1.4.1) were applied to the minimap2 and ngmlr aligned data, respectively. We used default settings for all callers, while at least five supporting reads were required for SV detection in NA19240, HG00733, HG00514 as well as normal-tumor paired COLO829 samples.

5.2.2 Evaluating recall and precision of each algorithm

We first used the evaluation method Truvari (https://github.com/spiralgenetics/truvari) developed by Genome-In-A-Bottle (GIAB) to examine the performance of each algorithm on HG002. The specific steps of SV

calling and processing for SVIM, Sniffles, CuteSV and pbsv were given by CuteSV (https://github.com/tjiangHIT/sv-benchmark). Furthermore, for SVision, SV with 'Covered' filter was considered as passed calls in the algorithm, and we replaced the 'Covered' with 'PASS' for the usage of option '`--passonly`' in Truvari. The raw calls of NanoVar were directly used as input for Truvari evaluation.

Moreover, the PAV call sets of NA19240, HG00733 and HG00514 were used to evaluate each algorithm. Note that the breakends, such as translocations, were first excluded from the raw detections and SVs ranging from 50bp to 100kbp were included in the analysis. BEDtools [85] (v2.30.0) was used to find the correct detections via the 50% reciprocal overlap test, while those failing the overlap test were considered as false detections. Specifically, we used command '`bedtools intersect -c -a pav.bed -b algorithm.bed -f 0.5 -r`' to count the unique number of matched ground truth calls. Given the number of ground truth calls (N), number of detections (D) and number of correct detections (D), the Recall, Precision and F-score were calculated as follows:

$$\text{Precision} = C/D$$

$$\text{Recall} = C/N$$

$$\text{F-score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

5.2.3 Identification and classification of PAV calls missed by each algorithm

Using command '`bedtools intersect -c -a pav.bed -b algorithm.bed -f 0.5 -r`', the missed PAV calls of each algorithm were labeled as zero matches in the last column of the output. Then, the simple repeats and Repeat Masker files obtained from UCSC Genome Browser were used to label the repeat element and calculate the percentage of repeat overlap. For simple repeats, the VNTR was assigned if the repeat unit length was longer than 7bp, otherwise, it was considered as STR. In this study, we only used repeat element LINE, SINE, LTR, VNTR and STR, while other repeat elements were classified as Others.

Additionally, we developed a pipeline to classify missed PAV calls according to the read mapping signatures. Firstly, the missed PAV calls were classified to three types of regions according to the average read mapping quality (avg_{mapq}), including i) no read mapping region (No_reads), ii) low

mapping quality regions (Low_mapq, $avg_{mapq} < 20$) and high confident mapping regions (High_mapq, $avg_{mapq} \geq 20$). The average mapping quality threshold was set according to the default minimum read quality used for SV detection algorithms. Secondly, we extracted the potential SV signature reads that span the PAV calls in the high confident mapping quality regions. In general, the 'I' and 'D' tags in the CIGAR string, and the primary reads and their supplementary alignments were collected and used to identify deletion (DEL), insertion (INS), inversion (INV) and duplication (DUP) signatures. The total number of SV signature reads spanning PAV calls was referred to as signature count. Afterwards, we applied the same implementation as Truvari to match PAV calls and detected SV signature reads. Specifically, for a given SV signature read with start and end position, we calculated the minimum distance between this signature and PAV call as well as their size similarity. If the minimum distance and the size similarity of a signature read was smaller than 500bp and larger than 0.5, respectively, it was considered as the nearest signature.

5.2.4 Evaluating breakpoint accuracy

To evaluate the breakpoint accuracy of each caller, the correct detection, compared with the benchmarks (i.e., PAV calls and short-read calls) was considered as the nearest one with similar size, where the distance and size similarity threshold were 500bp and 0.5, respectively. Note that for short-read benchmark calls, we used Manta with default settings to detected SVs from Illumina reads and evaluate the minimum breakpoint shift of overlapped detections as described above. We calculated the minimum breakpoint shift of the concordant detections to evaluate the breakpoint accuracy of each caller. For the breakpoint assessment of recurrent SVs, SURVIVOR [112] was used to identify the recurrent SVs among three samples for each caller with command '`SURVIVOR 500 3 0 0 0 50`', while translocations were excluded in breakpoint accuracy assessment. For other SV types, the breakpoint accuracy was evaluated by calculating the standard deviation of variant start and end position in the merged VCF file. If the standard deviation of both start and end position was smaller than 50bp, the corresponding recurrent SV was considered as accurate detection.

5.2.5 Examine call set overlaps between platforms and aligners

For each caller, the overlapped and unique calls of different platforms and aligners were identified with SURVIVOR, running command '`SURVIVOR 500 1 0 0 0 50`'. In particular, we only examined whether an SV was detected at a specific region of different aligners or platforms, while the SV type was not considered. For example, the ngmlr and minimap2 unique and overlapped calls detected by SVision on HiFi reads was obtained from the '`SUPP_VEC`' value of SURVIVOR merged output. Specifically, '`SUPP_VEC=11`' indicates overlapped calls, while '`SUPP_VEC=10`' or '`SUPP_VEC=01`' represents aligner unique detections. This comparison between aligners of identical platform was termed as fixed-platform, and the same process was applied to compare the detections between different platforms mapped with identical aligner, referring as fixed-aligner. Afterwards, the same repeat annotation procedure was applied to annotate the unique calls from fixed-platform and fixed-aligner. This process was also applied to identify tumor unique calls, which were obtained from variant of '`SUPP_VEC=10`'.

5.2.6 Data availability

Both the HiFi and ONT data for HG002 are obtained from ftp://ftp.ncbi.nlm.nih.gov/giab/ftp/data/AshkenazimTrio/HG002_NA24385_son, and the benchmark [92] for HG002 used in this chapter is from ftp://ftp-trace.ncbi.nlm.nih.gov/giab/ftp/data/AshkenazimTrio/analysis/NIST_SVs_Integration_v0.6/. The HiFi data for NA19240, HG00733 and HG00514 are obtained from http://ftp.1000genomes.ebi.ac.uk/vol1/ftp/data_collections/HGSVC2/working/, and the ONT data [9] for these samples are available at http://ftp.1000genomes.ebi.ac.uk/vol1/ftp/data_collections/hgsv_sv_discovery/working/20181210_ONT_rebasecalled/. The Phased Assembly Variant (PAV, v1.1.2) [10] for NA19240, HG00733 and HG00514 are downloaded from http://ftp.1000genomes.ebi.ac.uk/vol1/ftp/data_collections/HGSVC2/working/20210806_PAV_VCF/. The normal ONT data for COLO829 is obtained from Sequence Read Archive (SRA) with ERR2752451, and the tumor ONT data is downloaded with ERR2752452. The somatic SV truth set of COLO829 is obtained from https://github.com/UMCUGenetics/COLO829_somaticSV .

5.3 Results

In this section, we first assess the impact of aligners and platforms on SV detection consistency of each alignment-based detection methods. Then, we examine the recall and precision of each method affecting by aligners and platforms. Moreover, we systematically compare SVs detected by alignment-based approach and assembly approach, especially their breakpoint consistency. Finally, using tumor-normal paired sample, we assess the impact of aligners on detecting germline and somatic SVs.

5.3.1 Evaluating the impact of aligners and platforms on detection algorithms

Platform and aligner independency is one of the important features for detection algorithm in clinical usage. The detection consistency was thus assessed with three well-characterized samples (i.e., NA19240, HG00733 and HG00514) sequenced by HiFi and ONT technologies. As a result, more SVs were detected from minimap2 aligned data than that of ngmlr, and such difference was even significant for ONT data (Figure 5.1A). Though the percentage of detected deletions and insertions per genome varied across platform and aligner combinations, 20% more insertions and deletions were detected from minimap2 alignments than that of ngmlr. Notably, approximately 98% of SVIM discoveries were insertions or deletions from minimap2 aligned HiFi data, which was 15% and 38% more than pbsv and NanoVar detected, respectively (Figure 5.2).

Further analysis showed that a large number of duplications (around ≈7,000 without aligner or platform bias) detected by NanoVar was the major factor leading to a lower proportion of detected insertions and deletions (Figure 5.1C). We also noticed that the large number of duplications detected from ngmlr aligned data contributed to 20% difference of detected insertions and deletions between aligners for each caller (Figure 5.1C). Though pbsv, CuteSV, Sniffles and NanoVar could distinguish duplications from insertions, SVIM was the first algorithm that was capable of detecting tandem duplications (DUP:TANDEM) and dispersed duplications (DUP:INT), where around 10 dispersed duplications and 100 tandem duplications per genome were identified. Note that SVision and Sniffles were capable of identifying CSVs, where SVision reported ≈100 CSVs per sample and Sniffles identified three types of CSV (i.e., DEL/INV, DUP/INS and INVDUP) (Figure 5.1C).

We then examined the impacts of aligners on SV detection from different platforms, termed fixed-platform evaluation. The overlapping calls between

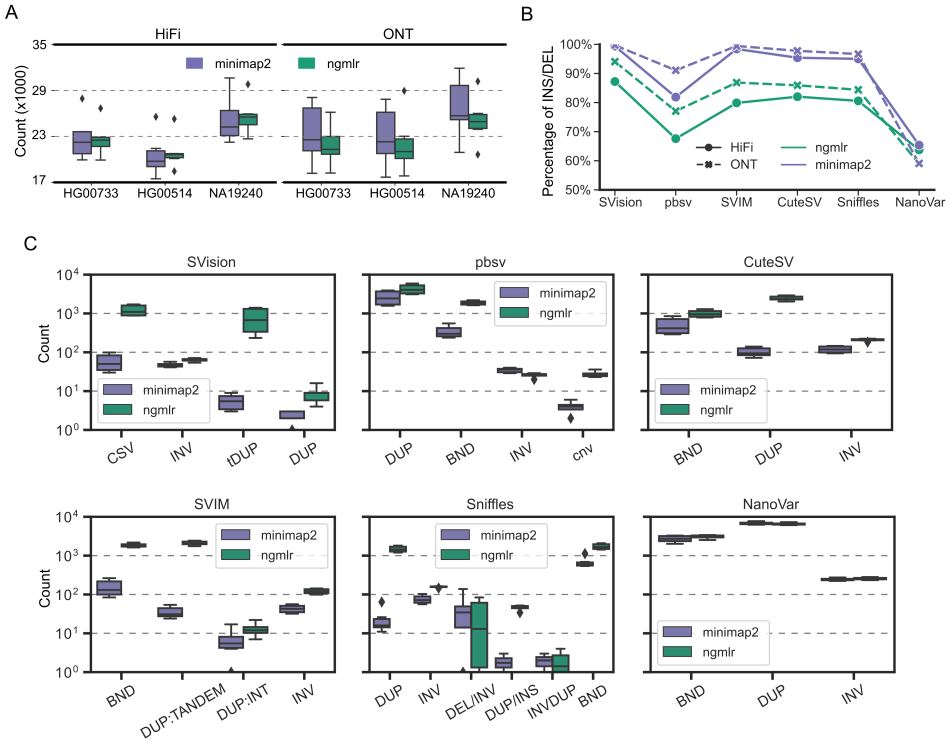


Figure 5.1: Overview of structural variants detected by six callers from three samples. (A) Number of structural variants of three samples detected from data generated by different aligners and platforms. (B) Percentage of deletions and insertions detected by each caller. (C) Number of detected structural variants of different types, excluding insertions and deletions.

two aligners were around 80% for both ONT and HiFi reads (Figure 5.2A), and breakpoint difference of most aligner concordant calls was less than 20bp (Figure 5.2B). Notably, breakpoint difference of pbsv calls was closer to 0bp on both platforms compared with other callers, indicating SV breakpoints reported by pbsv were less affected by aligners. Further analysis of aligner discordant calls revealed that all callers identified more duplications from ngmlr aligned HiFi and ONT data (Figure 5.2C), which was consistent with our previous observation on overall discoveries (Figure 5.1C), suggesting SV types reported by callers were depend on aligners. We reasoned that this limitation was largely due to the model-based SV detection approach, so that more duplications were detected from duplication like abnormal alignments observed in ngmlr aligned data.

In addition, we evaluated the platform influences, referred to as fixed-aligner evaluation, where the percentage of platform concordant calls ranged from 70% to 90% for different callers (Figure 5.2D). Though the platform concordant call took 90% of SVIM HiFi discoveries, three times more ONT unique calls were observed than HiFi unique calls (Figure 5.2D). Moreover, consistent with fixed-platform evaluation, pbsv produced concordant SV breakpoints of platform concordant calls (Figure 5.2E), suggesting pbsv was able to report consistent SV breakpoints that are less affected by aligners or platforms. Altogether, our results suggested that aligners played an important role in producing consistent SV breakpoints and types across platforms for each caller.

5.3.2 Evaluation recall and precision of detection algorithms using different benchmarks

Furthermore, it was critical to understand the sensitivity and specificity of detection algorithms for clinical applications. Therefore, we first benchmarked SVision, pbsv, CuteSV, Sniffles, NanoVar and SVIM with ground truth SVs of sample HG002. The ground truth set was an integration of multiple platforms and released by Genome-In-A-Bottle (GIAB), containing high-confident deletion and insertion calls, which had been widely used to evaluate the performance of SV detection algorithms [92]. The callers were applied to 30X HiFi data and 47X ONT data aligned with minimap2 and ngmlr, respectively. The results showed that SVision, pbsv, SVIM, CuteSV and Sniffles outperformed NanoVar across platforms and aligners. In addition, we noticed that all callers achieved the best performance on minimap2 aligned HiFi and ONT reads, and CuteSV achieved the highest F-score, followed by SVision, Sniffles and pbsv (Figure 5.3A). Though callers produced fewer

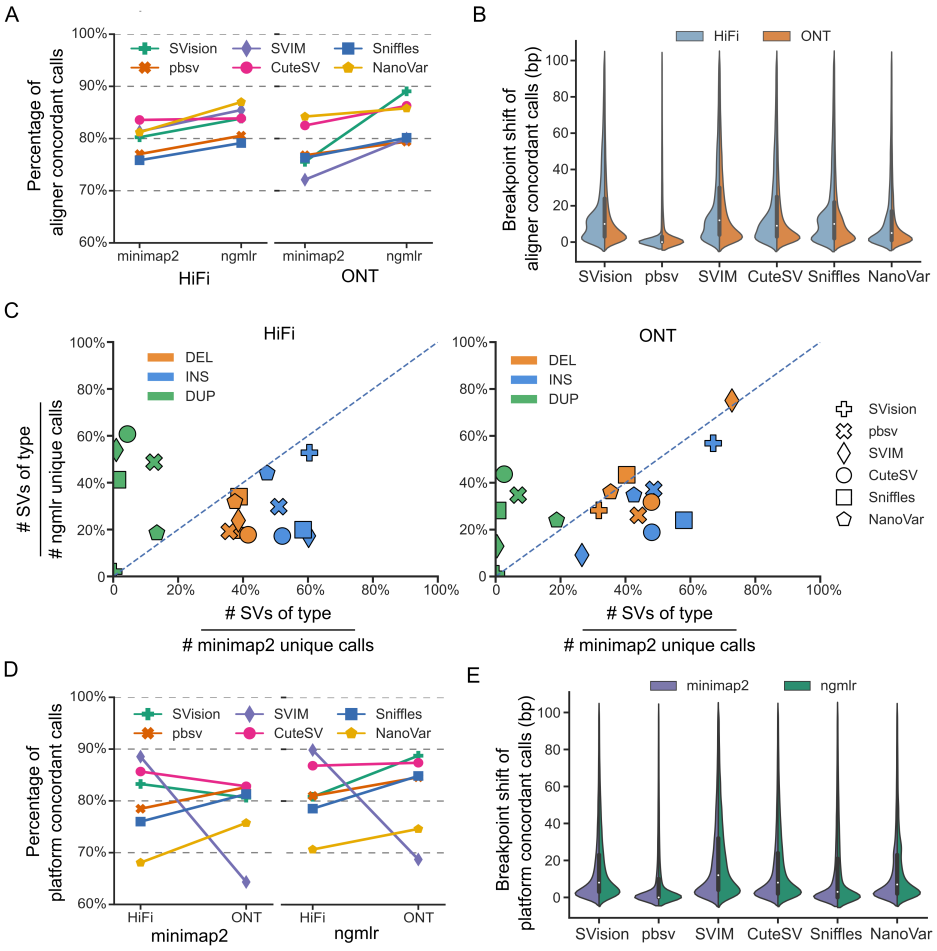


Figure 5.2: Effects of aligners and platforms on structural variants detection. (A-C) Fixed-platform evaluation of each caller. (A) Percentage of aligner concordant calls among all discoveries detected from ngmlr (vertical axis) and minimap2 (horizontal axis) alignments. (B) Breakpoint difference of aligner concordant calls. (C) Percentage of structural variant (SV) types among aligner discordant calls, i.e., minimap2 (horizontal axis) and ngmlr (vertical axis). (D-E) Fixed-aligner evaluation of each caller. (D) Percentage of platform concordant calls detected among all SVs detected from ONT (vertical axis) or HiFi (horizontal axis) reads. (E) Breakpoint difference of platform concordant calls.

correct detections on ngmlr aligned data, the precision of the six callers was comparable to minimap2 or even higher on ONT reads. For example, the precision of SVision detections on the minimap2 aligned ONT data was 80.5%, which increased to 89.9% on the ngmlr aligned ONT data (Figure 5.3A).

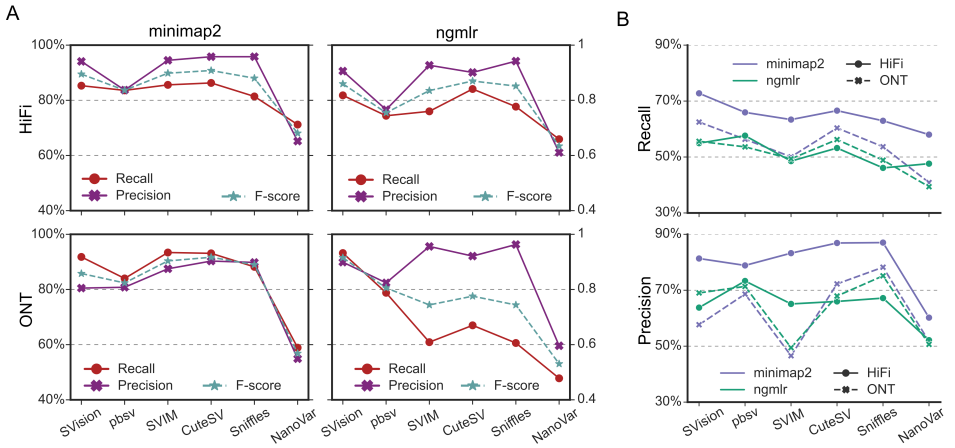


Figure 5.3: Evaluating recall and precision of six callers using different benchmarks. (A) Performance evaluated on sample HG002 HiFi and ONT data. (B) Average recall and precision evaluated on HG00514, HG00733 and NA19240.

In addition, PAV callsets of HG00514, HG00733 and NA19240 were used as ground truth to assess recall and precision of each caller. The PAV calls were detected from the highly contiguous haplotype assemblies released by HGSVC [10], which significantly improved the SV discoveries at repetitive regions compared with the HG002 truth set. Thus, the PAV callset was able to evaluate SV detection algorithms at both simple and complex genomic regions. Briefly, the SVs detected from mapped reads (i.e., HiFi and ONT aligned with minimap2 and ngmlr) of each caller were compared with the PAV calls by examining the reciprocal overlaps. Since translocation (BND) was not included in PAV calls, the BNDs from the raw calls from each caller were excluded and SVs ranging from 50bp to 100kbp were used for the performance assessment. As a result, all algorithms achieved their own best performance on minimap2 aligned HiFi reads, where SVision and pbsv ranked first on minimap2 and ngmlr aligned HiFi reads across samples, respectively (Figure 5.4B). We reasoned that this biased performance was largely due to the method of detecting PAV calls, i.e., detecting from the minimap2

aligned HiFi assemblies with extra alignment trimming. Though SV detection performance on ONT reads was not comparable with HiFi reads, the F-score of each caller based on different aligners were approximately equal, indicating less impact from aligners. Altogether, our results indicated that aligners affect more than platforms on recall and precision, where Sniffles, SVision, pbsv and CuteSV showed similar performance and consistently outperformed NanoVar across different platforms and aligners.

5.3.3 Features of PAV calls missed by detection algorithms

We then examined PAV calls missed by each caller on three samples (i.e., NA19240, HG00733 and HG00514), aiming to understand limitations of alignment-based SV detection algorithms. The missed PAV calls were considered those without matched detections via the reciprocal overlap test, and the best recall of detecting PAV calls was around 70% (Figure 5.3B). Among missed PAV calls, 70% and 28% of missed PAV calls were insertions and deletions, respectively (Figure 5.4A). Moreover, 80%, 70% and 60% of NanoVar, pbsv and CuteSV uniquely missed PAV calls were insertion, respectively, whereas more than 60% of SVIM and Sniffles missed PAV calls were deletions (Figure 5.4B). Further repeat annotation revealed that a large majority (≈70%) of missed SVs overlapped with VNTR regions, followed by STR regions (≈10%) (Figure 5.4C). These results suggested that an assembly-based approach significantly increased the sensitivity of detecting insertions and SVs in tandem repeat regions (i.e., VNTR and STR) compared with alignment-based detection. The above results were consistent with the conclusion drawn by HGSVC, where the predominant increase of PAV was among small SVs (<250bp) localized to simple repeat sequences.

Though the assembly-approach achieved remarkable results on SV detection, it was difficult to generalize for tumor genomes because of heterogeneity and aneuploidy. Therefore, we investigated whether the missed PAVs were detectable from alignment-based approaches. Firstly, we noticed that 80% of missed PAVs were located at high mapping quality regions (Figure 5.4D), providing confident alignments for SV signature reads identification. Afterwards, for missed PAV calls at high mapping quality regions, variant spanning reads were extracted and analyzed to find SV signatures. The results showed that the percentage of missed PAV calls with SV signature reads was independent of aligners for both HiFi and ONT reads, where NanoVar failed to report SVs from 88% and 77% of the genomic regions with SV signatures (Figure 5.4E).

Furthermore, we examined the nearest SV signatures, providing the direct evidence of detecting missed PAV calls. In principle, missed PAVs were not

able to be discovered from read mapping if we cannot identify the nearest SV signatures. On average, approximately 55% of missed PAVs contained nearest signatures for HiFi reads aligned with both aligners, whereas ONT reads were likely to produce more nearest signatures when aligned with minimap2 (Figure 5.4E). This indicated that half of missed PAV calls in high mapping quality regions could be recovered, while they were missed by routine SV callers due to the inaccurate breakpoints in repeat regions. Specifically, the nearest signatures could be identified from 90% of the missed PAV regions contained signatures, and the highest average PAV recall rate (≈70%) was achieved by minimap2 aligned HiFi reads, and we thus reasoned 17% more PAVs in high mapping quality regions could be detected based on signatures. Our analysis indicated that most of the PAV missed calls at simple repeat regions contain SV signature reads, and these PAV calls could be detected with proper breakpoint fine mapping.

5.3.4 Examining the effects of platforms and aligners on breakpoint accuracy

In addition, accurate breakpoints are critical to the downstream SV functional annotation such as gene annotation and known pathogenetic variant annotation, and we thus investigated the breakpoint accuracy of each caller by comparing with two independent call sets generated via orthogonal approaches, i.e., phased assembly and short-read. For phased assembly evaluation, using PAV calls, the breakpoint difference of ≈80% concordant calls were smaller than 50bp for minimap2 and ngmlr across different callers (Figure 5.5A). Moreover, consistent with the fixed-platform (Figure 5.2B) and fixed-aligner (Figure 5.2E) evaluation, pbsv achieved the most accurate breakpoints (breakpoint difference smaller than 10bp) without aligner and platform bias (Figure 5.5A). We next divided the concordant calls into two groups: i) accurate detections (breakpoint difference smaller than 50bp, Figure 5.5B) and ii) inaccurate detections (breakpoint difference larger than 50bp), and found that a significant number of inaccurate detections were located at VNTR regions (78%) (Figure 5.5C). This suggested that the breakpoints of SVs detected from read and assembly were largely different at simple repeat regions, especially in VNTR. Due to the aligner bias of PAV calls, breakpoint accuracy was further evaluated with short-read data, of which pbsv also showed the most accurate breakpoints and it was independent of aligners and platforms (Figure 5.6A).

On the contrary, the breakpoint accuracy of other callers was dependent on aligners, where we found the percentage of breakpoint shift smaller than

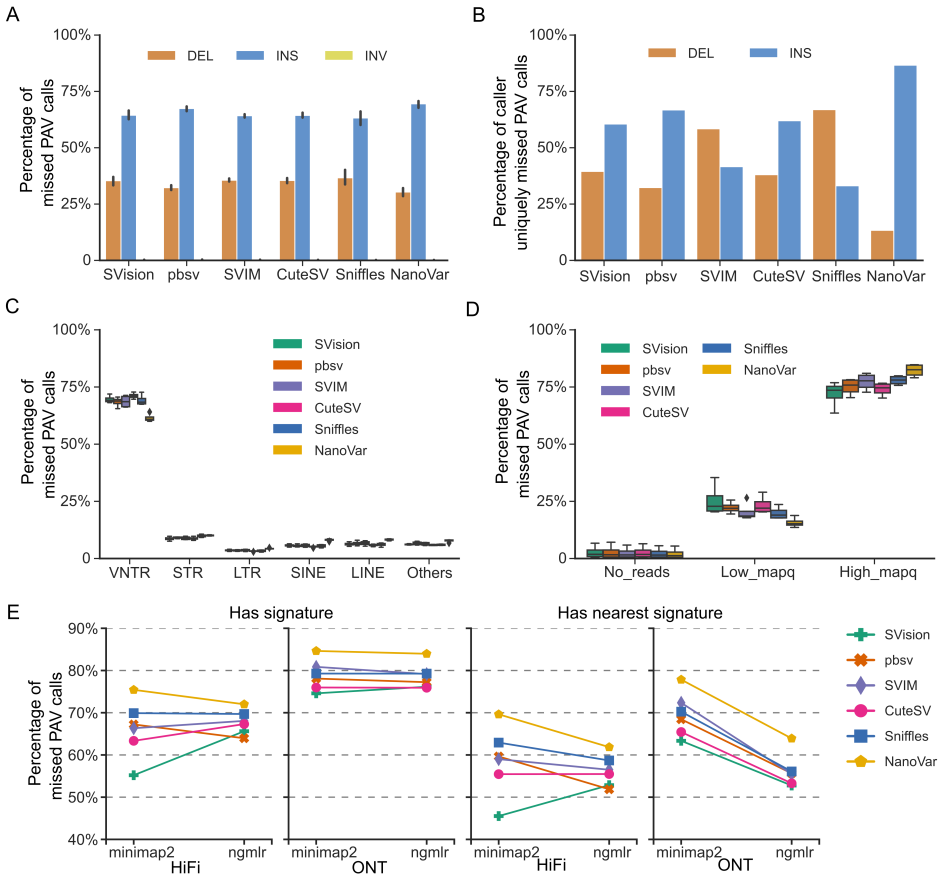


Figure 5.4: Features of missed Phased Assembly Variant by six callers. (A) Distribution of missed Phased Assembly Variants (PAVs) detected from different aligners and platforms. (B) Types of caller uniquely missed PAV calls. (C) Repeat annotation of missed PAVs. (D) Mapping quality of the missed PAV loci, including no read mapping (No_reads), low mapping quality (Low_mapq) and high mapping quality (High_mapq). (E) Missed PAV loci that had signatures and nearest signature identified from long reads aligned with minimap2 and ngmlr.

10bp increased 30% on minimap2 aligned reads (Figure 5.6A). Both PAV and short-read data revealed that HiFi data paired with minimap2 would produce the most accurate breakpoints for all callers. To avoid potential aligner bias of the benchmarks, we assessed the breakpoint accuracy of different callers by comparing the breakpoints of recurrent SVs among different samples. As a result, SVs detected by callers except SVision were likely to have consistent breakpoints on minimap2 aligned HiFi or ONT data, where Sniffles outperformed other callers among different platforms and aligners (Figure 5.6B). Our results suggested that the selection of aligner was critical to get consistent breakpoints for routine SV detection algorithms, while tandem repeat regions (i.e., VNTR) required extra breakpoint refinement if the caller was applied to repeat expansion related diseases, such as Huntington disease.

5.3.5 Effects of aligners on tumor SV detection

The above results suggested that aligners play an important role for consistent SV detection. We then evaluated the impact of aligner for tumor genome analysis, especially the performance of detecting somatic SVs from tumor unique calls. Briefly, each routine SV caller was used to detect SVs from tumor (ONT, ≈60X coverage) and normal (ONT, ≈40X coverage) data of COLO829 separately, and the filtering-based approach was applied to identify tumor unique calls, which are also called putatively somatic SVs. As a result, the total number of SVs detected by NanoVar from tumor and normal tissues was independent of aligners, whereas Sniffles, CuteSV and SVIM detected more SVs from minimap2 alignments comparing to ngmlr, thereby leading to 5% more minimap2 unique detections than that of ngmlr (Figure 5.7A). Furthermore, we investigated the impact of aligners on identifying tumor unique calls, which is one of the critical steps to obtain somatic SVs. The results showed that the percentage of tumor unique calls obtained from NanoVar and Sniffles was less affected by aligners (Figure 5.7B), and NanoVar had the largest number of tumor unique calls, i.e., 7,626 and 7,676 from minimap2 and ngmlr alignments, respectively.

On average, 50% of the tumor unique calls were inside the repetitive regions, of which the majority of them were annotated as SINE or LINE. As for the SV types of tumor unique calls, ≈4,500 putatively somatic deletions were identified from SVIM calls detected based on minimap2 alignments, which was four times more than detected insertions (≈1,000 events) (Figure 5.7C). Comparably, approximately 3,300 of the tumor unique calls identified from NanoVar was translocations, attributing to 44% of the tumor unique calls,

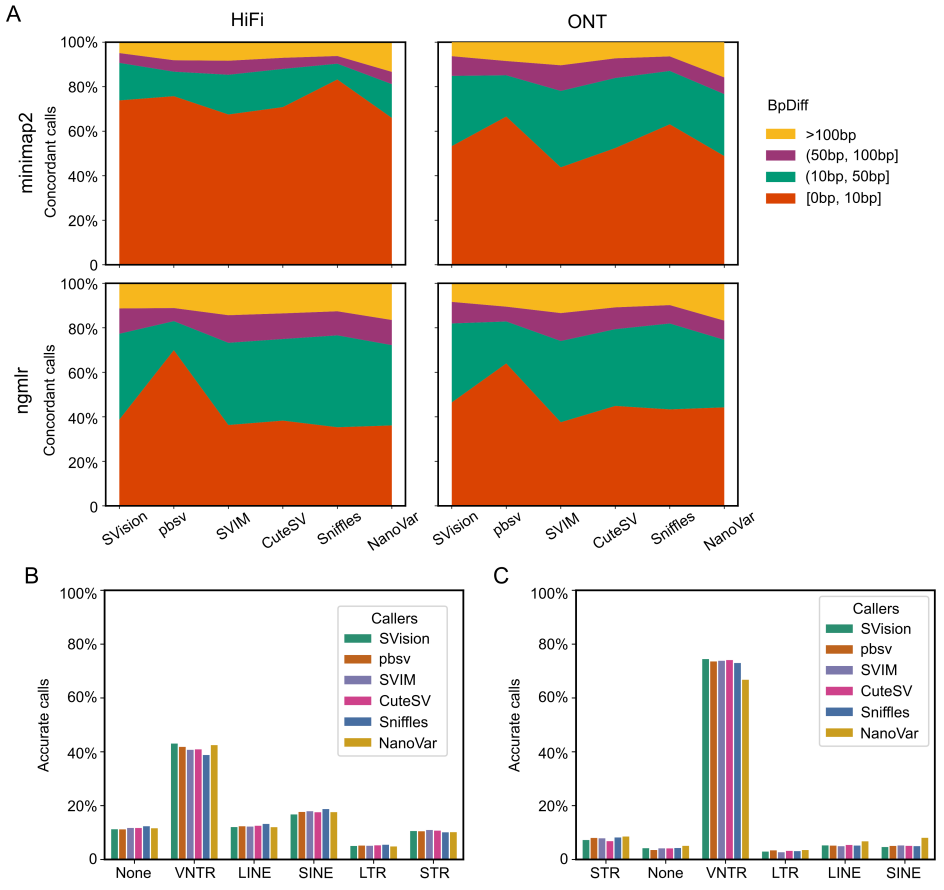


Figure 5.5: Evaluating the breakpoint accuracy of structural variants detected by six callers with Phased Assembly Variant. (A) The breakpoint difference (*BpDiff*) of concordant calls between callers' detections and Phased Assembly Variants (PAVs). (B) The repeat annotation of accurate calls (*BpDiff* $\leq$ 50bp). (C) The repeat annotation of inaccurate calls (*BpDiff* $>$ 50bp).

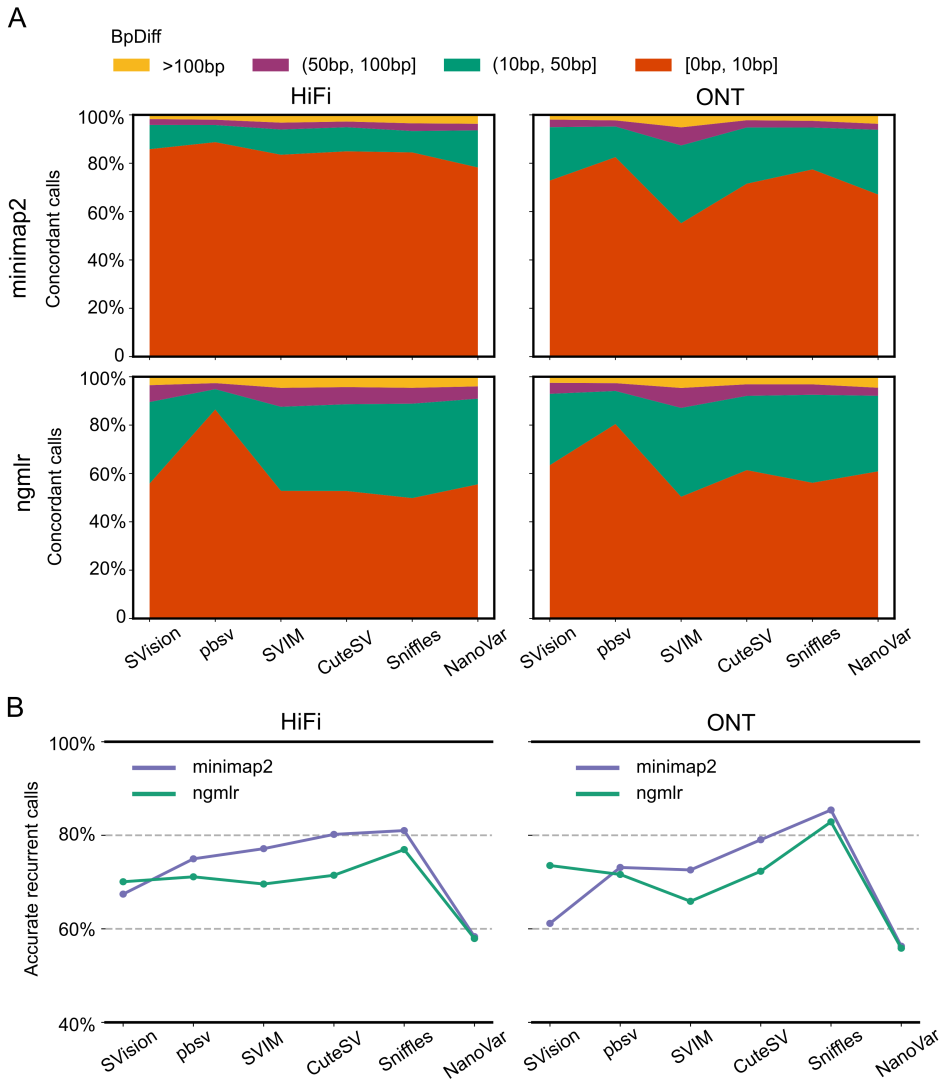


Figure 5.6: Evaluating the breakpoint accuracy with short-read data and assessing breakpoints of recurrent structural variants. (A) The breakpoint difference (BpDiff) of structural variants (SVs) detected by six callers and those detected by short-read data. (B) The breakpoint accuracy of recurrent SVs among three samples (i.e., NA19240, HG00733 and HG00514).

and it was independent of aligners (Figure 5.7C). Furthermore, 1,500 putatively somatic translocations were identified from pbsv calls using ngmlr alignments, which was 15 times more than translocations identified from minimap2 alignments. In addition, we assessed the putative somatic SVs with the COLO829 somatic benchmark, containing 78 (i.e., 38 deletions, 13 translocations, 7 duplications, 7 inversions and 3 insertions) high-quality SVs released by a multi-platform study. As a result, though 57 ground truth somatic SVs were missed by one of the five callers, all somatic insertions were correctly detected. In addition, 35 out of 57 ground truth SVs, consisting of six translocations, 21 deletions, five inversions and three duplications, could not be detected by any combination of callers and aligners. We thus reasoned that integration of discoveries from different callers might substantially increase the detection sensitivity.

5.4 Conclusion

SVs are important types of genomic alterations to form population diversity [5] and to drive disease progression, such as tumorigenesis [6], but are more difficult to detect than small variants from short-read data due to the limited read length. In the past five years, the long-read sequencing technologies and the newly developed algorithms greatly facilitate the detection of SVs from both healthy [113] and tumor genomes [114], improving our understanding of the functional impact of SVs. Remarkably, the SV detection based on haplotype-resolved assembly enables the haplotype-aware germline SV detection, and significantly improves the detection at complex genomic regions, such as segmental duplication and variable number tandem repeat (VNTR) [9, 10]. Though studies have attempted to evaluate the performance of routine SV detection algorithms, we explored the major factors affecting the ability of different algorithms in detecting SVs in complex genomic regions and somatic SVs. Overall, using public HiFi and ONT data from four healthy genomes and ONT data from a normal-tumor paired sample, we evaluated multiple aligners and SV callers to assess the routine SV detection algorithms by comparing with PAV calls and high-quality somatic truth set.

In this chapter, we examined the performance of each SV caller with two aligners (i.e., minimap2 and ngmlr). The alignment time and memory usage had been systematically evaluated in other studies [115], which was out of the scope of this study. For both HiFi and ONT platforms, all callers tend to detect more SVs on minimap2 aligned data than that of ngmlr, while SVIM produced more ONT unique calls on both aligners. Since the same parameters

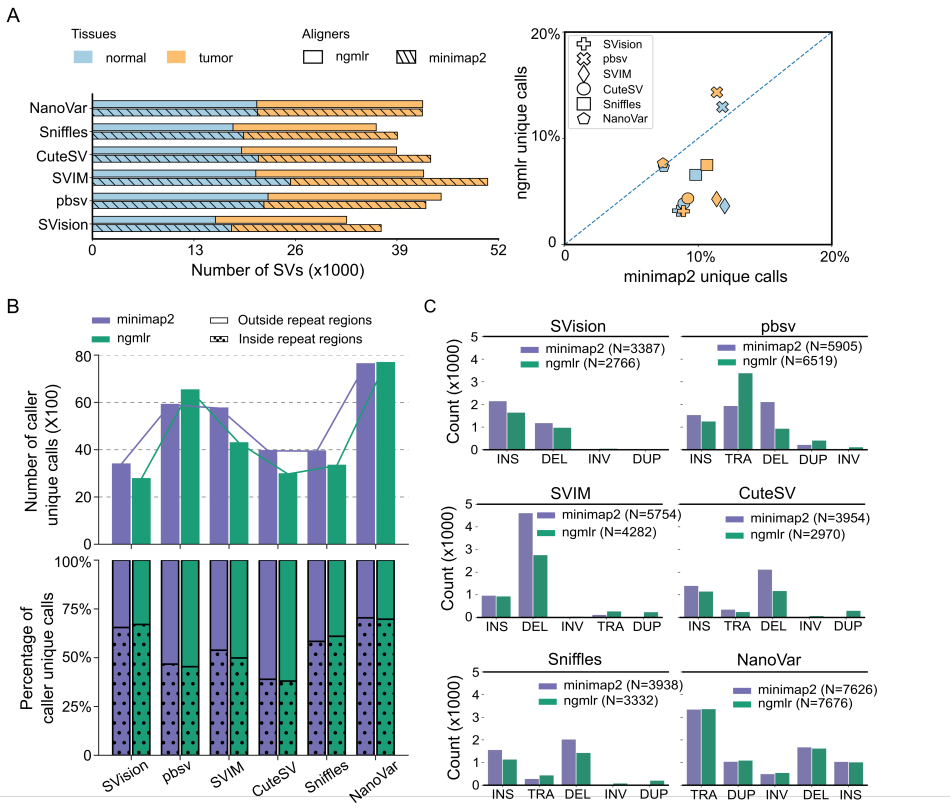


Figure 5.7: Evaluating the filtering-based somatic structural variants detection of the six callers. (A) Comparison of structural variants (SVs) detected from both tumor and normal tissues (left panel) and percentage of aligner unique calls detected by each caller (right panel). (B) Percentage of tissue unique calls detected by each caller (top panel) and repeat annotation of caller unique calls (bottom panel). (C) SV types of tumor unique calls identified from each caller.

were used for each caller on different platforms and aligners, SVIM might need specific parameter tuning for ONT data. Moreover, we found that aligner was the major factor affecting the number of detected SVs and their breakpoint accuracy, whereas the breakpoint of pbsv were less affected by aligners and platforms. Therefore, we recommend using pbsv with either minimap2 or ngmlr for the initial SV for a new sample. In terms of the recall and precision of callers, both the GIAB and PAV benchmarking suggested the bias of minimap2 paired with HiFi data. Though these two benchmarks showed limitations for evaluation, they suggested that SVision, Sniffles, pbsv, CuteSV and SVIM showed similar performance and outperformed NanoVar. In addition, Sniffles and CuteSV showed the highest precision for all of the HiFi and ONT data tested, while SVision call sets generally had a higher recall rate. Therefore, Sniffles and CuteSV should be used when high precision was the priority, pbsv was recommended when accurate breakpoint were required, and SVision should be considered if high sensitivity was desired.

Additionally, and uniquely to this study, we investigated the features of PAV calls missed by read-based detection to assess whether read-based calling was capable of generating comprehensive call set. It was expected that most of the missed PAV calls were found at VNTR regions and insertion was the major SV type missed by read-based detection. While our results suggested that the majority of the missed PAV loci contained SV signature reads, and most importantly, this was not depending on aligners, indicating the read-based detection would recover most of the PAV calls.

Moreover, since we also observed high SV breakpoint concordance on different platforms using identical aligner, the selection of sequencing platform would have less impact on SV detection for a new sample. However, it should be noted that the majority of the inaccurate and inconsistent calls were found at tandem repeat regions, so that disease associated with repeat expansion requires extra downstream analysis or specific algorithms, such as Straglr [116] and NanoSatellite [101]. Another critical step in studying tumor genomes was to characterize the somatic SVs, which were considered closely related to the tumorigenesis. Due to lack of long-read based somatic SV detection algorithms, we only evaluated the recall of detecting somatic SVs in tumor unique calls. However, this approach identified ground truth somatic SVs in low precision, suggesting an urgent demand of standalone somatic SV detection algorithms in the community.

Altogether, our analysis suggested that alignment-based callers would uncover a near comprehensive and high-quality call set of a genome, while the filtering-based approach for somatic SV discovery was suboptimal, leading to high false positive rate. Thus, as the detection of SVs from long-reads becomes

routine and gradually applied to investigate tumor genomes, it is imperative to start to consider and work towards developing robust pipelines or algorithms for SV detection in tumors. Moreover, we expect resources from ONT and PacBio to accumulate as the technology improves and the sequencing price decreases, which leaves great opportunities for better somatic benchmark generation and future algorithm development for clinical applications.

Chapter 6

Conclusions and perspectives

In this chapter, we present our conclusions and provide perspectives for future research.

6.1 Conclusions

It is a fact that the research discipline of computational genomics largely emerged from sequence analysis. Indeed, deciphering the language of life from DNA, RNA or protein sequences has been greatly facilitated by the advanced sequencing technologies. From short-read DNA sequencing to single-molecule-sequencing (SMS) DNA sequencing, methods for sequence alignment, genome structural variants detection, etc., are actively developed by numerous researchers in the past decade. This thesis focuses on developing novel algorithms for several pivotal parts in applying sequencing technology to SV detection in clinical settings, including detection, characterization and validation. Moreover, this thesis provides a systematic evaluation of factors affecting clinical applications.

With the rapid development of high-throughput sequencing (HTS) technology, genomic rearrangements or structural variants (SVs) have been recognized to affect more than SNPs or Indels in genome evolution and disease progression. Recently, an increasing number of simple SVs are found to be complex events, which not only misleads downstream analysis but also introduces another layer of difficulty for SV detection. So far, most of the methods detect SVs by following a model and match approach, i.e., a sequencing data specific alignment model is first created for different SV types and further matched with the observations from sequence alignment for discovery. Though mode-based approach is well-performed for detecting simple SVs

(i.e., deletions, inversions, duplications, insertions and translocations), it is neither effective nor efficient to resolve complex events due to the complicated internal structure for modeling. On the other hand, complex events are largely unexplored, which also limits the detection through a model-based approach. Therefore, novel algorithms that can detect complex events or curate existing discoveries are in great demand, especially for sequencing oriented clinical diagnosis.

In this thesis, we first design two novel algorithms, graph based and deep learning based, to detect complex structural variants (CSVs) without predefined models. Secondly, we systematically assess the reproducibility of current SV detection methods among different datasets, helping users select proper methods and datasets for their applications. In this way, we address our main research questions, dealing with detection and assessment of structural variants.

Since short-read sequencing has been widely used in large cohort studies, a graph based approach was first developed, aiming to profile complex events at a large scale. Specifically, the graph was used to represent alternative connections derived from an individual genome, from which CSVs were detected as frequent local maximal subgraphs. However, due to the limited read length, the graph-based approach based on short-read data was not able to resolve the accurate internal structure.

As the price of long-read sequencing decreases, its usage for both research and clinical settings is expected to increase dramatically in the next few years. Therefore, we further developed SVision, a deep-learning based multi-object recognition framework, to automatically detect and characterize both simple and complex SVs from sequence image. In addition, since vast amounts of sequence data and SV callsets have become available, a high-throughput orthogonal validation approach is also in demand. We thus developed a novel algorithm, SpotSV, to assess the quality of predicted SVs, including their breakpoints and type. SpotSV uses the denoised segment to examine the breakpoints of predicted SVs, improving the assessment of complex events and SVs at repetitive regions. Our results suggest that the novel detection algorithms and the validation algorithm outperformed the state-of-the-art methods.

Furthermore, it is expected that HTS based SV detection will become a routine clinical diagnosis approach, especially for complex diseases, such as cancer. We then systematically evaluated the robustness of detection algorithms by using different sequence alignment algorithms and sequencing platforms, of which the sensitivity, specificity and breakpoint accuracy were examined.

6.2 Perspectives

This section contains directions for future research.

Flexible connection graph data structure for SV detection
In Chapter 2, a graph, representing alternative connections, has been successfully used to detect simple and complex events from an individual genome. Recently, long-read sequencing has revolutionized the detection and study of SVs, and it would add extra connections to the graph built on short-read. Moreover, long-reads are able to accurately resolve the CSV internal structure as we show in Chapter 3. Thus, a flexible data structure that could integrate both the advantage of short-read and long-read sequencing is expected to improve the detection, such as finding the accurate breakpoints induced by short-reads and internal structure characterized by long-reads.

Frequent subgraph mining among population genome graph
Since a large amount of sequencing data is available for both healthy and disease genomes, one of the biggest issues is how to detect SVs at a large scale, which is critical to understand evolution and disease progression. In principle, each individual genome mapping to the reference genome could be converted to a connection graph, thereby leading to a population-scale genome connection graph. Afterwards, frequent subgraphs representing certain types of SV or CSV could be detected based on the subgraph topology. Most importantly, a population-scale genome connection graph would enable rare SV detection in personal genome because the rare SV of an individual genome might be frequent among population. This feature makes it a valuable data structure to compare SVs within population or between populations, and it could also facilitate the analysis of undiagnosed disease.

Compare and merge SVs at population-scale
In Chapter 4, we develop a novel algorithm to assess the quality of predicted SVs, especially for complex ones and SVs at repetitive regions. Similar to SV quality evaluation of an individual genome, comparison and merging SVs at population-scale is another challenging computational problem, affecting downstream analysis, such as SV formation and Mendelian disease. In general, SV comparison is difficult because they vary across individuals and are discovered through different data and methods. Therefore, an approach that could detect and merge SVs simultaneously at population-scale is able to avoid the issue of detecting from different data and methods. We would also adopt the idea of a population-scale connection graph, integrating both short-read and long-read data, for SV comparison and merging at population-scale. Specifically, if one SV is common in population, it is expected to detect

a local subgraph of dense alternative connections derived from different individuals, and these connections are approximately equal based on specific edge attributes. Finally, a merged SV call set of multiple samples could be derived from detected subgraphs in the connection graph.

SV graph for somatic SV detection

In Chapter 3, a graph is used to represent the CSV internal structure, which also enables the graph based validation via graph alignment. So far, a number of studies have shown the strength of using long-reads to analyze tumor genomes compared with short-read data, whereas algorithms for somatic SV detection based on long-reads are underdeveloped. The graph implemented in Chapter 3 provides an important hint to isolate somatic SVs from the genetic background of a patient. Briefly, the germline SVs (i.e., genetic background) represented as mini graph are first embedded into the linear reference genome, resulting in a germline graph. Secondly, the sequencing data from matched tumor tissue could be aligned to the germline graph, from which the newly formed subgraph or path is identified to be a somatic SV and the augmentation graph could be built. Since tumor tissue might contain cells originated from different clones, detecting SVs from different clones is important to help understand the tumorigenesis. The above step could be done recursively to detect subclonal SVs, and this recursive process is terminated when new path could not be identified from the graph alignments. This is a complicated computational approach, which requires optimized graph augmentation and alignment algorithm for effective SV detection.

A structural variants analysis system for clinical settings

In Chapter 5, we have evaluated the factors that might affect SV detection from the clinical perspectives. On the other hand, the downstream analysis, such as SV quality assessment (Chapter 4) and SV merging, is also critical for clinical diagnosis. Therefore, a SV analysis system from detection to result interpretation would fill the gap between research output and clinical application, which becomes even important as the number of undiagnosed cases increases and the price of sequencing decrease. Moreover, a user-friendly interface for doctors or non-computing experts is preferred and valuable to expand the usage of sequencing technology assisted diagnosis. Therefore, as an algorithm designer and implementer, future research will continue to develop algorithms for challenging biological or clinical problems. In addition, we aim to carefully implement the algorithms and provide user-friendly graphic interfaces, enabling the application of HTS technology in clinical settings.

Bibliography

[1] Alkan C, Coe BP, Eichler EE: Genome structural variation discovery and genotyping. Nat Rev Genet 2011, 12:363–376.

[2] Li H, Durbin R: Fast and accurate short read alignment with Burrows-Wheeler transform. Bioinformatics 2009, 25:1754–1760.

[3] Langmead B, Trapnell C, Pop M, Salzberg SL: Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. Genome Biol 2009, 10:R25.

[4] Poplin R, Chang PC, Alexander D, Schwartz S, Colthurst T, Ku A, Newburger D, Dijamco J, Nguyen N, Afshar PT, et al.: A universal SNP and small-indel variant caller using deep neural networks. Nat Biotechnol 2018, 36:983–987.

[5] Sudmant PH, Rausch T, Gardner EJ, Handsaker RE, Abyzov A, Huddleston J, Zhang Y, Ye K, Jun G, Fritz MH, et al.: An integrated map of structural variation in 2,504 human genomes. Nature 2015, 526:75–81.

[6] Li Y, Roberts ND, Wala JA, Shapira O, Schumacher SE, Kumar K, Khurana E, Waszak S, Korbel JO, Haber JE, et al.: Patterns of somatic structural variation in human cancer genomes. Nature 2020, 578:112–121.

[7] Collins RL, Brand H, Karczewski KJ, Zhao X, Alfoldi J, Francioli LC, Khera AV, Lowther C, Gauthier LD, Wang H, et al.: A structural variation reference for medical and population genetics. Nature 2020, 581:444–451.

[8] Ho SS, Urban AE, Mills RE: Structural variation in the sequencing era. Nat Rev Genet 2020, 21:171–189.

[9] Chaisson MJP, Sanders AD, Zhao X, Malhotra A, Porubsky D, Rausch T, Gardner EJ, Rodriguez OL, Guo L, Collins RL, et al.: Multi-platform discovery of haplotype-resolved structural variation in human genomes. Nat Commun 2019, 10:1784.

[10] Ebert P, Audano PA, Zhu Q, Rodriguez-Martin B, Porubsky D, Bonder MJ, Sulovari A, Ebler J, Zhou W, Serra Mari R, et al.: Haplotype-resolved diverse human genomes and integrated analysis of structural variation. Science 2021, 372.

[11] Quinlan AR, Hall IM: Characterizing complex structural variation in germline and somatic genomes. Trends Genet 2012, 28:43–53.

[12] Collins RL, Brand H, Redin CE, Hanscom C, Antolik C, Stone MR, Glessner JT, Mason T, Pregno G, Dorrani N, et al.: Defining the diverse spectrum of inversions, complex structural variation, and chromothripsis in the morbid human genome. Genome Biol 2017, 18:36.

[13] Spielmann M, Lupianez DG, Mundlos S: Structural variation in the 3D genome. Nat Rev Genet 2018, 19:453–467.

[14] Carvalho CM, Lupski JR: Mechanisms underlying structural variant formation in genomic disorders. Nat Rev Genet 2016, 17:224–238.

[15] Telli ML, Timms KM, Reid J, Hennessy B, Mills GB, Jensen KC, Szallasi Z, Barry WT, Winer EP, Tung NM, et al.: Homologous Recombination Deficiency (HRD) score rredicts response to Platinum-containing neoadjuvant chemotherapy in patients with Triple-Negative Breast Cancer. Clin Cancer Res 2016, 22:3764–3773.

[16] Kurtz S, Phillippy A, Delcher AL, Smoot M, Shumway M, Antonescu C, Salzberg SL: Versatile and open software for comparing large genomes. Genome Biol 2004, 5:R12.

[17] Li H: Minimap2: pairwise alignment for nucleotide sequences. Bioinformatics 2018, 34:3094–3100.

[18] Sedlazeck FJ, Rescheneder P, Smolka M, Fang H, Nattestad M, von Haeseler A, Schatz MC: Accurate detection of complex structural variations using single-molecule sequencing. Nat Methods 2018, 15:461–468.

[19] Miga KH, Koren S, Rhie A, Vollger MR, Gershman A, Bzikadze A, Brooks S, Howe E, Porubsky D, Logsdon GA, et al.: Telomere-to-telomere assembly of a complete human X chromosome. Nature 2020, 585:79–84.

[20] Li H, Handsaker B, Wysoker A, Fennell T, Ruan J, Homer N, Marth G, Abecasis G, Durbin R, Genome Project Data Processing S: The Sequence Alignment/Map format and SAMtools. Bioinformatics 2009, 25:2078–2079.

[21] Zerbino DR, Birney E: Velvet: algorithms for de novo short read assembly using De Bruijn graphs. Genome Res 2008, 18:821–829.

[22] Ruan J, Li H: Fast and accurate long-read assembly with wtdbg2. Nat Methods 2020, 17:155–158.

[23] Cheng H, Concepcion GT, Feng X, Zhang H, Li H: Haplotype-resolved de novo assembly using phased assembly graphs with hifiasm. Nat Methods 2021, 18:170–175.

[24] Simpson JT, Durbin R: Efficient de novo assembly of large genomes using compressed data structures. Genome Res 2012, 22:549–556.

[25] Zepeda-Mendoza CJ, Morton CC: The iceberg under water: unexplored complexity of chromoanagenesis in congenital disorders. Am J Hum Genet 2019, 104:565–577.

[26] Li Z, Chen Y, Mu D, Yuan J, Shi Y, Zhang H, Gan J, Li N, Hu X, Liu B, et al.: Comparison of the two major classes of assembly algorithms: overlap-layout-consensus and De-Bruijn-graph. Brief Funct Genomics 2012, 11:25–37.

[27] Marschall T, Costa IG, Canzar S, Bauer M, Klau GW, Schliep A, Schonhuth A: CLEVER: clique-enumerating variant finder. Bioinformatics 2012, 28:2875–2882.

[28] Sherman RM, Salzberg SL: Pan-genomics in the human genome era. Nat Rev Genet 2020, 21:243–254.

[29] Li H, Feng X, Chu C: The design and construction of reference pangenome graphs with minigraph. Genome Biol 2020, 21:265.

[30] Hickey G, Heller D, Monlong J, Sibbesen JA, Siren J, Eizenga J, Dawson ET, Garrison E, Novak AM, Paten B: Genotyping structural

variants in pangenome graphs using the vg toolkit. Genome Biol 2020, 21:35.

[31] Garrison E, Siren J, Novak AM, Hickey G, Eizenga JM, Dawson ET, Jones W, Garg S, Markello C, Lin MF, et al.: Variation graph toolkit improves read mapping by representing genetic variation in the reference. Nat Biotechnol 2018, 36:875–879.

[32] Sibbesen JA, Maretty L, Danish Pan-Genome C, Krogh A: Accurate genotyping across variant classes and lengths using variant graphs. Nat Genet 2018, 50:1054–1059.

[33] Rautiainen M, Makinen V, Marschall T: Bit-parallel sequence-to-graph alignment. Bioinformatics 2019, 35:3599–3607.

[34] Rautiainen M, Marschall T: GraphAligner: rapid and versatile sequence-to-graph alignment. Genome Biol 2020, 21:253.

[35] Hadi K, Yao X, Behr JM, Deshpande A, Xanthopoulakis C, Tian H, Kudman S, Rosiene J, Darmofal M, DeRose J, et al.: Distinct classes of complex structural variation uncovered across thousands of cancer genome graphs. Cell 2020, 183:197–210 e132.

[36] Ye K, Schulz MH, Long Q, Apweiler R, Ning Z: Pindel: a pattern growth approach to detect break points of large deletions and medium sized insertions from paired-end short reads. Bioinformatics 2009, 25:2865–2871.

[37] Rausch T, Zichner T, Schlattl A, Stutz AM, Benes V, Korbel JO: DELLY: structural variant discovery by integrated paired-end and split-read analysis. Bioinformatics 2012, 28:i333–i339.

[38] Layer RM, Chiang C, Quinlan AR, Hall IM: LUMPY: a probabilistic framework for structural variant discovery. Genome Biol 2014, 15:R84.

[39] Chen K, Wallis JW, McLellan MD, Larson DE, Kalicki JM, Pohl CS, McGrath SD, Wendl MC, Zhang Q, Locke DP, et al.: BreakDancer: an algorithm for high-resolution mapping of genomic structural variation. Nat Methods 2009, 6:677–681.

[40] Cameron DL, Di Stefano L, Papenfuss AT: Comprehensive evaluation and characterisation of short read general-purpose structural variant calling software. Nat Commun 2019, 10:3240.

[41] Kosugi S, Momozawa Y, Liu X, Terao C, Kubo M, Kamatani Y: Comprehensive evaluation of structural variation detection algorithms for whole genome sequencing. Genome Biol 2019, 20:117.

[42] Chen X, Schulz-Trieglaff O, Shaw R, Barnes B, Schlesinger F, Kallberg M, Cox AJ, Kruglyak S, Saunders CT: Manta: rapid detection of structural variants and indels for germline and cancer sequencing applications. Bioinformatics 2016, 32:1220–1222.

[43] Gao R, Davis A, McDonald TO, Sei E, Shi X, Wang Y, Tsai PC, Casasent A, Waters J, Zhang H, et al.: Punctuated copy number evolution and clonal stasis in triple-negative breast cancer. Nat Genet 2016, 48:1119–1130.

[44] Yates LR, Knappskog S, Wedge D, Farmery JHR, Gonzalez S, Martincorena I, Alexandrov LB, Van Loo P, Haugland HK, Lilleng PK, et al.: Genomic evolution of breast cancer metastasis and relapse. Cancer Cell 2017, 32:169–184 e167.

[45] Nattestad M, Goodwin S, Ng K, Baslan T, Sedlazeck FJ, Rescheneder P, Garvin T, Fang H, Gurtowski J, Hutton E, et al.: Complex rearrangements and oncogene amplifications revealed by long-read DNA and RNA sequencing of a breast cancer cell line. Genome Res 2018, 28:1126–1135.

[46] Sanchis-Juan A, Stephens J, French CE, Gleadall N, Megy K, Penkett C, Shamardina O, Stirrups K, Delon I, Dewhurst E, et al.: Complex structural variants in Mendelian disorders: identification and breakpoint resolution using short- and long-read genome sequencing. Genome Med 2018, 10:95.

[47] Greer SU, Nadauld LD, Lau BT, Chen J, Wood-Bouwens C, Ford JM, Kuo CJ, Ji HP: Linked read sequencing resolves complex genomic rearrangements in gastric cancer metastases. Genome Med 2017, 9:57.

[48] Lee JJ, Park S, Park H, Kim S, Lee J, Lee J, Youk J, Yi K, An Y, Park IK, et al.: Tracing oncogene rearrangements in the mutational history of lung adenocarcinoma. Cell 2019, 177:1842–1857 e1821.

[49] Baca SC, Prandi D, Lawrence MS, Mosquera JM, Romanel A, Drier Y, Park K, Kitabayashi N, MacDonald TY, Ghandi M, et al.: Punctuated evolution of prostate cancer genomes. Cell 2013, 153:666–677.

[50] Korbel JO, Campbell PJ: Criteria for inference of chromothripsis in cancer genomes. Cell 2013, 152:1226–1236.

[51] Sanders AD, Meiers S, Ghareghani M, Porubsky D, Jeong H, van Vliet M, Rausch T, Richter-Pechanska P, Kunz JB, Jenni S, et al.: Single-cell analysis of structural variations and complex rearrangements with tri-channel processing. Nat Biotechnol 2019.

[52] Carvalho CMB, Lupski JR: Mechanisms underlying structural variant formation in genomic disorders. Nature Reviews Genetics 2016, 17:224–238.

[53] Malhotra A, Lindberg M, Faust GG, Leibowitz ML, Clark RA, Layer RM, Quinlan AR, Hall IM: Breakpoint profiling of 64 cancer genomes reveals numerous complex rearrangements spawned by homology-independent mechanisms. Genome Res 2013, 23:762–776.

[54] Ye K, Wang J, Jayasinghe R, Lameijer EW, McMichael JF, Ning J, McLellan MD, Xie M, Cao S, Yellapantula V, et al.: Systematic discovery of complex insertions and deletions in human cancers. Nat Med 2016, 22:97–104.

[55] Zhang CZ, Leibowitz ML, Pellman D: Chromothripsis and beyond: rapid genome evolution from complex chromosomal rearrangements. Genes Dev 2013, 27:2513–2530.

[56] Soylev A, Le TM, Amini H, Alkan C, Hormozdiari F: Discovery of tandem and interspersed segmental duplications using high-throughput sequencing. Bioinformatics 2019, 35:3923–3930.

[57] Zhao X, Emery SB, Myers B, Kidd JM, Mills RE: Resolving complex structural genomic rearrangements using a randomized approach. Genome Biol 2016, 17:126.

[58] Cameron DL, Schroder J, Penington JS, Do H, Molania R, Dobrovic A, Speed TP, Papenfuss AT: GRIDSS: sensitive and specific genomic rearrangement detection using positional de Bruijn graph assembly. Genome Res 2017, 27:2050–2060.

[59] Arthur JG, Chen X, Zhou B, Urban AE, Wong WH: Detection of complex structural variation from paired-end sequencing data. bioRxiv 2017:200170.

[60] Liao VCC, Chen MS: DFSP: a Depth-First SPelling algorithm for sequential pattern mining of biological sequences. Knowledge and Information Systems 2014, 38:623–639.

[61] Tsai HP, Yang DN, Chen MS: Mining group movement patterns for tracking moving objects efficiently. Ieee Transactions on Knowledge and Data Engineering 2011, 23:266–281.

[62] Huang Y, Zhang LQ, Zhang PS: A framework for mining sequential patterns from spatio-temporal event data sets. Ieee Transactions on Knowledge and Data Engineering 2008, 20:433–448.

[63] Ye K, Kosters WA, Ijzerman AP: An efficient, versatile and scalable pattern growth approach to mine frequent patterns in unaligned protein sequences. Bioinformatics 2007, 23:687–693.

[64] Pei J, Han J, Wang W: Constraint-based sequential pattern mining: the pattern-growth methods. Journal of Intelligent Information Systems 2007, 28:133–160.

[65] Pei J, Han JW, Mortazavi-Asl B, Wang JY, Pinto H, Chen QM, Dayal U, Hsu MC: Mining sequential patterns by pattern-growth: the PrefixSpan approach. IEEE Transactions on Knowledge and Data Engineering 2004, 16:1424–1440.

[66] Li H, Homer N: A survey of sequence alignment algorithms for next-generation sequencing. Brief Bioinform 2010, 11:473–483.

[67] Li H, Durbin R: Fast and accurate short read alignment with Burrows-Wheeler transform. Bioinformatics 2009, 25:1754–1760.

[68] Bolognini D, Sanders A, Korbel JO, Magi A, Benes V, Rausch T: VISOR: a versatile haplotype-aware structural variant simulator for short and long read sequencing. Bioinformatics 2019.

[69] McPherson A, Wu C, Wyatt AW, Shah S, Collins C, Sahinalp SC: nFuse: discovery of complex genomic rearrangements in cancer using high-throughput sequencing. Genome Res 2012, 22:2250–2261.

[70] Dzamba M, Ramani AK, Buczkowicz P, Jiang Y, Yu M, Hawkins C, Brudno M: Identification of complex genomic rearrangements in cancers using CouGaR. Genome Res 2017, 27:107–117.

[71] Delcher AL, Phillippy A, Carlton J, Salzberg SL: Fast algorithms for large-scale genome alignment and comparison. Nucleic Acids Res 2002, 30:2478–2483.

[72] Zhao X, Weber AM, Mills RE: A recurrence-based approach for validating structural variation using long-read sequencing technology. Gigascience 2017, 6:1–9.

[73] Ottaviani D, LeCain M, Sheer D: The role of microhomology in genomic structural variation. Trends Genet 2014, 30:85–94.

[74] Kramara J, Osia B, Malkova A: Break-induced replication: the where, the why, and the how. Trends Genet 2018, 34:518–531.

[75] Hartlerode AJ, Willis NA, Rajendran A, Manis JP, Scully R: Complex breakpoints and template switching associated with non-canonical termination of homologous recombination in mammalian cells. PLoS Genet 2016, 12:e1006410.

[76] Zhou W, Zhang F, Chen X, Shen Y, Lupski JR, Jin L: Increased genome instability in human DNA segments with self-chains: homology-induced structural variations via replicative mechanisms. Hum Mol Genet 2013, 22:2642–2651.

[77] Yang L, Luquette LJ, Gehlenborg N, Xi R, Haseley PS, Hsieh CH, Zhang C, Ren X, Protopopov A, Chin L, et al.: Diverse mechanisms of somatic structural variations in human cancer genomes. Cell 2013, 153:919–929.

[78] Chen W, McKenna A, Schreiber J, Haeussler M, Yin Y, Agarwal V, Noble WS, Shendure J: Massively parallel profiling and predictive modeling of the outcomes of CRISPR/Cas9-mediated double-strand break repair. Nucleic Acids Res 2019, 47:7989–8003.

[79] Allen F, Crepaldi L, Alsinet C, Strong AJ, Kleshchevnikov V, De Angeli P, Palenikova P, Khodak A, Kiselev V, Kosicki M, et al.: Predicting the mutations generated by repair of Cas9-induced double-strand breaks. Nat Biotechnol 2018.

[80] Quigley DA, Dang HX, Zhao SG, Lloyd P, Aggarwal R, Alumkal JJ, Foye A, Kothari V, Perry MD, Bailey AM, et al.: Genomic hallmarks and structural variation in metastatic prostate cancer. Cell 2018, 175:889.

[81] Fraser M, Sabelnykova VY, Yamaguchi TN, Heisler LE, Livingstone J, Huang V, Shiah YJ, Yousif F, Lin X, Masella AP, et al.: Genomic hallmarks of localized, non-indolent prostate cancer. Nature 2017, 541:359–364.

[82] Fujimoto A, Wong JH, Yoshii Y, Akiyama S, Tanaka A, Yagi H, Shigemizu D, Nakagawa H, Mizokami M, Shimada M: Whole-genome sequencing with long reads reveals complex structure and origin of structural variation in human genetic variations and somatic mutations in cancer. Genome Med 2021, 13:65.

[83] Bolognini D, Sanders A, Korbel JO, Magi A, Benes V, Rausch T: VISOR: a versatile haplotype-aware structural variant simulator for short- and long-read sequencing. Bioinformatics 2020, 36:1267–1269.

[84] Krumsiek J, Arnold R, Rattei T: Gepard: a rapid and sensitive tool for creating dotplots on genome scale. Bioinformatics 2007, 23:1026–1028.

[85] Quinlan AR, Hall IM: BEDTools: a flexible suite of utilities for comparing genomic features. Bioinformatics 2010, 26:841–842.

[86] Guennewig B, Lim J, Marshall L, McCorkindale AN, Paasila PJ, Patrick E, Kril JJ, Halliday GM, Cooper AA, Sutherland GT: Defining early changes in Alzheimer's disease from RNA sequencing of brain regions differentially affected by pathology. Sci Rep 2021, 11:4865.

[87] Audano PA, Sulovari A, Graves-Lindsay TA, Cantsilieris S, Sorensen M, Welch AE, Dougherty ML, Nelson BJ, Shah A, Dutcher SK, et al.: Characterizing the major structural variant alleles of the human genome. Cell 2019, 176:663–675 e619.

[88] Belyeu JR, Chowdhury M, Brown J, Pedersen BS, Cormier MJ, Quinlan AR, Layer RM: Samplot: a platform for structural variant visual validation and automated filtering. Genome Biol 2021, 22:161.

[89] Nattestad M, Aboukhalil R, Chin CS, Schatz MC: Ribbon: intuitive visualization for complex genomic variation. Bioinformatics 2021, 37:413–415.

[90] Aganezov S, Goodwin S, Sherman RM, Sedlazeck FJ, Arun G, Bhatia S, Lee I, Kirsche M, Wappel R, Kramer M, et al.: Comprehensive analysis of structural variants in breast cancer genomes using single-molecule sequencing. Genome Res 2020, 30:1258–1273.

[91] Jiang T, Liu S, Cao S, Liu Y, Cui Z, Wang Y, Guo H: Long-read sequencing settings for efficient structural variation detection based on comprehensive evaluation. BMC Bioinformatics 2021, 22:552.

[92] Zook JM, Hansen NF, Olson ND, Chapman L, Mullikin JC, Xiao C, Sherry S, Koren S, Phillippy AM, Boutros PC, et al.: A robust benchmark for detection of germline large deletions and insertions. Nat Biotechnol 2020, 38:1347–1355.

[93] Sone J, Mitsuhashi S, Fujita A, Mizuguchi T, Hamanaka K, Mori K, Koike H, Hashiguchi A, Takashima H, Sugiyama H, et al.: Long-read sequencing identifies GGC repeat expansions in NOTCH2NLC associated with neuronal intranuclear inclusion disease. Nat Genet 2019, 51:1215–1221.

[94] Hiatt SM, Lawlor JMJ, Handley LH, Ramaker RC, Rogers BB, Partridge EC, Boston LB, Williams M, Plott CB, Jenkins J, et al.: Long-read genome sequencing for the molecular diagnosis of neurodevelopmental disorders. HGG Adv 2021, 2.

[95] Pauper M, Kucuk E, Wenger AM, Chakraborty S, Baybayan P, Kwint M, van der Sanden B, Nelen MR, Derks R, Brunner HG, et al.: Long-read trio sequencing of individuals with unsolved intellectual disability. Eur J Hum Genet 2021, 29:637–648.

[96] Gong L, Wong CH, Cheng WC, Tjong H, Menghi F, Ngan CY, Liu ET, Wei CL: Picky comprehensively detects high-resolution structural variants in nanopore long reads. Nat Methods 2018, 15:455–460.

[97] Zhou B, Ho SS, Greer SU, Zhu X, Bell JM, Arthur JG, Spies N, Zhang X, Byeon S, Pattni R, et al.: Comprehensive, integrated, and phased whole-genome analysis of the primary ENCODE cell line K562. Genome Res 2019, 29:472–484.

[98] Sakamoto Y, Xu L, Seki M, Yokoyama TT, Kasahara M, Kashima Y, Ohashi A, Shimada Y, Motoi N, Tsuchihara K, et al.: Long-read sequencing for non-small-cell lung cancer genomes. Genome Res 2020, 30:1243–1257.

[99] Zhou B, Ho SS, Greer SU, Spies N, Bell JM, Zhang X, Zhu X, Arthur JG, Byeon S, Pattni R, et al.: Haplotype-resolved and integrated genome analysis of the cancer cell line HepG2. Nucleic Acids Res 2019, 47:3846–3861.

[100] Peneau C, Imbeaud S, La Bella T, Hirsch TZ, Caruso S, Calderaro J, Paradis V, Blanc JF, Letouze E, Nault JC, et al.: Hepatitis B virus integrations promote local and distant oncogenic driver alterations in hepatocellular carcinoma. Gut 2021.

[101] De Roeck A, De Coster W, Bossaerts L, Cacace R, De Pooter T, Van Dongen J, D'Hert S, De Rijk P, Strazisar M, Van Broeckhoven C, Sleegers K: NanoSatellite: accurate characterization of expanded tandem repeat length and sequence through whole genome long-read sequencing on PromethION. Genome Biol 2019, 20:239.

[102] Jiang T, Liu Y, Jiang Y, Li J, Gao Y, Cui Z, Liu Y, Liu B, Wang Y: Long-read-based human genomic structural variation detection with cuteSV. Genome Biol 2020, 21:189.

[103] Heller D, Vingron M: SVIM: structural variant identification using mapped long reads. Bioinformatics 2019, 35:2907–2915.

[104] Tham CY, Tirado-Magallanes R, Goh Y, Fullwood MJ, Koh BTH, Wang W, Ng CH, Chng WJ, Thiery A, Tenen DG, Benoukraf T: NanoVar: accurate characterization of patients' genomic structural variants using low-depth nanopore sequencing. Genome Biol 2020, 21:56.

[105] Cretu Stancu M, van Roosmalen MJ, Renkens I, Nieboer MM, Middelkamp S, de Ligt J, Pregno G, Giachino D, Mandrile G, Espejo Valle-Inclan J, et al.: Mapping and phasing of structural variation in patient genomes using nanopore sequencing. Nat Commun 2017, 8:1326.

[106] Hiltemann S, Jenster G, Trapman J, van der Spek P, Stubbs A: Discriminating somatic and germline mutations in tumor DNA samples without matching normals. Genome Res 2015, 25:1382–1390.

[107] Franco I, Helgadottir HT, Moggio A, Larsson M, Vrtacnik P, Johansson A, Norgren N, Lundin P, Mas-Ponte D, Nordstrom J, et al.: Whole genome DNA sequencing provides an atlas of somatic mutagenesis in healthy human cells and identifies a tumor-prone cell type. Genome Biol 2019, 20:285.

[108] Dixon JR, Xu J, Dileep V, Zhan Y, Song F, Le VT, Yardimci GG, Chakraborty A, Bann DV, Wang Y, et al.: Integrative detection and

analysis of structural variation in cancer genomes. Nat Genet 2018, 50:1388–1398.

[109] Thibodeau ML, O'Neill K, Dixon K, Reisle C, Mungall KL, Krzywinski M, Shen Y, Lim HJ, Cheng D, Tse K, et al.: Improved structural variant interpretation for hereditary cancer susceptibility using long-read sequencing. Genet Med 2020, 22:1892–1897.

[110] Huang KL, Mashl RJ, Wu Y, Ritter DI, Wang J, Oh C, Paczkowska M, Reynolds S, Wyczalkowski MA, Oak N, et al.: Pathogenic germline variants in 10,389 adult cancers. Cell 2018, 173:355–370 e314.

[111] Grobner SN, Worst BC, Weischenfeldt J, Buchhalter I, Kleinheinz K, Rudneva VA, Johann PD, Balasubramanian GP, Segura-Wang M, Brabetz S, et al.: The landscape of genomic alterations across childhood cancers. Nature 2018, 555:321–327.

[112] Jeffares DC, Jolly C, Hoti M, Speed D, Shaw L, Rallis C, Balloux F, Dessimoz C, Bahler J, Sedlazeck FJ: Transient structural variations have strong effects on quantitative traits and reproductive isolation in fission yeast. Nat Commun 2017, 8:14061.

[113] Beyter D, Ingimundardottir H, Oddsson A, Eggertsson HP, Bjornsson E, Jonsson H, Atlason BA, Kristmundsdottir S, Mehringer S, Hardarson MT, et al.: Long-read sequencing of 3,622 Icelanders provides insight into the role of structural variants in human diseases and other traits. Nat Genet 2021, 53:779–786.

[114] Alvarez EG, Demeulemeester J, Otero P, Jolly C, Garcia-Souto D, Pequeno-Valtierra A, Zamora J, Tojo M, Temes J, Baez-Ortega A, et al.: Aberrant integration of Hepatitis B virus DNA promotes major restructuring of human hepatocellular carcinoma genome architecture. Nat Commun 2021, 12:6910.

[115] Zhou A, Lin T, Xing J: Evaluating nanopore sequencing data processing pipelines for structural variation identification. Genome Biol 2019, 20:237.

[116] Chiu R, Rajan-Babu IS, Friedman JM, Birol I: Straglr: discovering and genotyping tandem repeat expansions using whole genome long-read sequences. Genome Biol 2021, 22:224.

English summary

Structural variants (SVs) are the hidden architecture of the human genome, and are critical for us to understand diseases, evolution, and so on. The development of both sequencing technologies and computational tools greatly facilitates the detection of SVs, while misinterpreting or even missing complex ones. Detecting and characterizing complex events is a typical field requiring multiple disciplines, i.e., domain knowledge and computer science algorithms.

In this thesis, we introduce novel algorithms to detect and validate complex events, and assess the reproducibility of current SV detection pipelines for clinical and research settings.

Chapter 1 begins with the introduction of DNA, various types of SVs and sequencing technologies. Then fundamental techniques and algorithms from computer science related to the thesis are briefly described. Pattern mining, graphs and deep learning are applied to detect and characterize different types of complex SVs (CSVs). CSVs usually contain multiple breakpoints and are often missed or misinterpreted by traditional detection strategies developed for simple SV detection. Most importantly, CSVs are largely underexplored, making them even challenging to detect based on existing knowledge. Considering the sequencing cost and detection accuracy for different application scenarios, we first develop algorithms for both short-read and long-read sequencing technologies without pattern matching against a database of know structures of SVs. Currently, short-read sequencing is significantly reduced in cost and has been widely applied to clinical diagnostics and cohort studies. To detect CSVs from short-read sequencing, we consider that SVs change the connections of adjacent segments with alternative connection derived from abnormally aligned paired-end reads.

Accordingly, in Chapter 2, we propose a frequent maximal subgraph mining approach (Mako) to detect both SVs and CSVs from a graph built from abnormal alignments. This graph is called signal graph, where nodes represent positions of connected genomic segments and edges indicate alternative and reference connections between genomic segments. We then apply a linearized database with prefix index schema to efficiently detect frequent maximal subgraphs from the signal graph, from which SVs and CSVs are derived from detected subgraphs. Compared to other approaches, a graph is able to depict complex genomic segment connections originating from CSVs. Moreover, detected CSV subgraphs are interpretable, making it possible to understand and compare different types of CSVs. However, limited by the read length of short-read sequencing, two simple SVs from different haplotypes might be detected as a single CSV event. On the other hand, short read length would

also be problematic for read mapping at regions with potential CSVs, where breakpoints belonging to a CSV could be potentially missed by callers. With the advances of long-read sequencing, a single read is more likely to span an entire CSV event compared to short-read sequencing. This greatly simplifies the confirmation of CSVs by investigating the difference and similarity of reads and its counterpart sequence from the reference genome. As a result, an increasing number of CSV have been revealed through intensive breakpoint analysis and visual confirmation. However, this is only applicable to small amounts of samples, which would not satisfy the ever-increasing demand of studying CSVs at population scale.

In Chapter 3, we leverage the human intelligence of identifying CSVs from visualization, and develop a multi-object recognition framework (SVision) to detect both SVs and CSVs without previous knowledge of SV structures. We first propose a sequence-to-image coding schema, which not only describes the differences and similarities of two sequences but also removes the background sequence context. This coding strategy enables us to efficiently and effectively detect CSVs even at complex genomic regions. In addition, CSV representation or interpretation is another challenging problem that hinders the definition and cross study of CSVs. Inspired by the graph structure used in Chapter 2, we also use a graph to represent and compare CSVs detected from long-read data, from which we are able to classify different types of CSVs by measuring graph isomorphisms. But different from nodes in the signal graph proposed in Chapter 2, a node of the CSV graph in Chapter 3 represents a matched sequence between two sequences. This feature makes it possible to genotype CSVs based on graph alignment. Moreover, this provides a novel idea of detecting SVs from a SV graph instead of detecting from a biased linear reference. We expected this graph-based SV detection approach will help to detect somatic SVs and SVs from tumor subclones.

Having developed two SV detection algorithms for trending sequencing technologies, we next aim to further explore the possibilities of applying long-read sequencing in various applications. In general, we observe that the high-confident SVs detected from reproducible analysis pipelines are critical for long-read applications in either clinical or research settings. Therefore, we first develop a high-throughput SV validation approach (SpotSV) to identify high-confident SVs in Chapter 4. Different from SV detection, SV validation focuses on exclude false negatives and corrects inaccurate SV characterizations, such as type and breakpoints. The idea of this validation approach is also inspired by the way in which human experts visually characterize SVs. We first apply a light-weighted local realignment method to locate different segments between two sequences. Then, we adopt a simple two-dimensional geometry

calculation to measure the confidence of a detected SV.

Additionally, in Chapter 5, we assess the reproducibility of existing pipelines on detecting germline and somatic SVs. This chapter systematically investigates the difference of assembly-based and alignment-based SV detection, highlighting major factors for discordant discoveries. We expect that this evaluation will help non-experts to understand the difference between methods and thus will help them to select proper analysis pipelines in their own applications.

Finally, in Chapter 6, we mention future research directions regarding the accurate detection of SVs for both research and clinical settings. Notably, we are confident that the combination of BioTech and InfoTech, often referred to as BT-IT, will revolutionize future health care.

Nederlandse samenvatting

Structurele varianten (SV's) vormen eigenlijk de verborgen architectuur van het menselijk genoom, en zijn van cruciaal belang voor ons om ziektes en evolutie te begrijpen. De ontwikkeling van sequencing-technologie en algoritmen maakt de detectie van SV's mogelijk, maar complexe SV's worden soms verkeerd geïnterpreteerd of zelfs over het hoofd gezien. Het ontdekken en karakteriseren van complexe events is een vakgebied dat meerdere disciplines omvat, waaronder domeinkennis en gespecialiseerde algoritmen.

In dit proefschrift introduceren we nieuwe algoritmen om complexe events te detecteren en te valideren, en om de reproduceerbaarheid van huidige SV-detectie pipelines voor klinische toepassingen en onderzoek te beoordelen.

Hoofdstuk 1 begint met de introductie van DNA, verschillende soorten SV's en sequencing-technologie. Vervolgens worden fundamentele technieken en algoritmen uit de informatica die verband houden met het proefschrift kort beschreven. Pattern mining, grafen en deep learning worden toegepast om verschillende soorten complexe SV's (CSV's) te detecteren en te karakteriseren. CSV's bevatten meestal meerdere breakpoints en worden vaak over het hoofd gezien of verkeerd geïnterpreteerd door traditionele strategieën die zijn ontwikkeld voor eenvoudige SV-detectie. Het belangrijkste is dat CSV's grotendeels onderbelicht zijn, waardoor ze zelfs moeilijk te detecteren zijn op basis van bestaande kennis. Rekening houdend met de sequencing-kosten en detectienauwkeurigheid voor verschillende scenario's, hebben we eerst algoritmen ontwikkeld voor zowel short-read als long-read sequencing-technologie zonder patroonovereenkomst ten opzicht van een database met bekende SV-structuren. Momenteel is short-read sequencing aanzienlijk goedkoper en wordt het op grote schaal toegepast in klinische diagnostiek en cohortstudies. Om CSV's via short-read sequencing te detecteren, zijn we van mening dat SV's de verbindingen van aangrenzende segmenten veranderen door middel van een alternatieve verbinding, afgeleid van abnormaal uitgelijnde reads met gepaarde uiteinden.

Daarom stellen we in Hoofdstuk 2 een aanpak (Mako) voor die gebruik maakt van een frequente maximale subgraaf, gebaseerd op abnormale alignments, om zowel SV's als CSV's te detecteren. Deze graaf wordt signal-graaf genoemd, waarbij knopen posities van verbonden genoom-segmenten vertegenwoordigen en takken alternatieve en referentieverbindingen tussen genoomsegmenten aangeven. Vervolgens hebben we een gelineariseerde database met prefix-indexschema toegepast om efficiënt frequente maximale subgrafen in de signal-graaf op te sporen, waaruit SV's en CSV's werden afgeleid uit gedetecteerde subgrafen. In vergelijking met andere benaderingen is een

graaf in staat om complexe genoom-segmentverbindingen weer te geven die afkomstig zijn van CSV's. Bovendien zijn gedetecteerde CSV-subgrafen interpreteerbaar, waardoor het mogelijk wordt om verschillende soorten CSV's beter te begrijpen en te vergelijken. Echter, beperkt door de leeslengte van short-read sequencing, kunnen twee eenvoudige SV's van verschillende haplotypes worden gedetecteerd als een enkele CSV-gebeurtenis. Aan de andere kant zou een korte leeslengte (read length) ook problematisch zijn voor toewijzing in gebieden met potentiële CSV's, waar breakpoints die bij een CSV horen, mogelijk door "callers" zouden kunnen worden gemist. Met de vooruitgang in long-read sequencing, is de kans groter dat een enkele read een hele CSV-gebeurtenis omvat in vergelijking met short-read sequencing. Dit vereenvoudigt de bevestiging van CSV's aanzienlijk door het verschil en de gelijkenis van de read en de corresponderende sequentie op het referentiegenoom te onderzoeken. Als gevolg hiervan is een toenemend aantal CSV's ontdekt door middel van intensieve breakpoint-analyse en visuele bevestiging. Dit is echter alleen van toepassing op kleine hoeveelheden steekproeven, die niet voldoen aan de steeds toenemende vraag naar het bestuderen van CSV's op populatieschaal.

In Hoofdstuk 3 hebben we gebruik gemaakt van menselijke intelligentie voor het identificeren van CSV's op basis van visualisatie, en hebben we een raamwerk voor herkenning van meerdere objecten (SVision) ontwikkeld om zowel SV's als CSV's te detecteren zonder voorafgaande kennis van SV-structuren. We stellen eerst een sequentie-naar-beeld coderingsschema voor, dat niet alleen de verschillen en overeenkomsten van twee sequenties beschrijft, maar ook de context van de achtergrondsequentie verwijdert. Deze coderingsstrategie stelt ons in staat om CSV's efficiënt en effectief te detecteren, zelfs in complexe genoom-gebieden. Verder is de CSV-representatie of -interpretatie een ander uitdagend probleem dat de definitie en studie van CSV's belemmert. Geïnspireerd door de graafstructuur die in Hoofdstuk 2 wordt gebruikt, hebben we ook een graaf gebruikt om CSV's die zijn gedetecteerd uit long-read gegevens weer te geven en te vergelijken, van waaruit we verschillende typen CSV's kunnen classificeren door graafisomorfismen te benutten. Maar anders dan een knoop in de signal-graaf voorgesteld in Hoofdstuk 2, vertegenwoordigt een knoop van de CSV-graaf in Hoofdstuk 3 een gematchte deelsequentie tussen twee sequenties. Deze functie maakt het mogelijk om CSV's te genotyperen op basis van alignment van de graaf. Bovendien biedt dit een nieuw idee voor het detecteren van SV's uit een SV-graaf in plaats van het detecteren van vertekening in de referentie. We verwachten dat deze op grafen gebaseerde SV-detectiebenadering zal helpen om somatische SV's en SV's van tumorsubklonen te detecteren.

Nadat we twee SV-detectiealgoritmen voor trending sequencing-technologie hebben ontwikkeld, willen we vervolgens de mogelijkheden van het gebruik van long-read sequencing in verschillende toepassingen verder onderzoeken. Over het algemeen zijn we van mening dat de meest betrouwbare SV's die zijn gedetecteerd via reproduceerbare analyse-pipelines van cruciaal belang zijn voor long-read toepassingen in klinische of onderzoeksomgevingen. Daarom hebben we in Hoofdstuk 4 eerst een high-throughput SV-validatieaanpak (SpotSV) ontwikkeld om de meest betrouwbare SV's te identificeren. Anders dan SV-detectie, richt SV-validatie zich op het uitsluiten van false negatives en corrigeert onnauwkeurige SV-karakteriseringen, zoals type en breakpoints. Het idee van deze validatieaanpak is ook geinspireerd op de manier waarop menselijke experts SV's visueel karakteriseren. We hebben eerst een eenvoudige lokale herschikkingsmethode toegepast om verschillende segmenten tussen twee sequenties te lokaliseren. Vervolgens hebben we een eenvoudige tweedimensionale berekening gebruikt om de betrouwbaarheid van een gedetecteerde SV te meten.

Daarnaast hebben we in Hoofdstuk 5 de reproduceerbaarheid van bestaande pipelines voor het detecteren van kiembaan SV's en somatische SV's beoordeeld. Dit hoofdstuk onderzoekt systematisch het verschil tussen op assembly gebaseerde en op alignment gebaseerde SV-detectie, waarbij de belangrijkste factoren voor tegenstrijdige ontdekkingen werden benadrukt. We verwachten dat deze evaluatie niet-experts zal helpen om het verschil in methoden te begrijpen en hen zo in staat zal stellen om de juiste analysepipelines in hun eigen toepassingen te selecteren.

Tot slot, in Hoofdstuk 6, beschrijven we toekomstige onderzoeksrichtingen met betrekking tot de nauwkeurige detectie van SV's voor zowel onderzoeks- als klinische instellingen. We zijn er met name van overtuigd dat de combinatie van BioTech en InfoTech, ook wel BT-IT genoemd, een revolutie teweeg zal brengen in de toekomstige gezondheidszorg.

Acknowledgements

I would like to express my gratitude to many people who helped me and supported me in the five years of my graduate life.

My foremost thanks go to my supervisor Dr. Kai Ye from Xi'an Jiaotong University, for introducing me to the wonder and frustrations of scientific study. He has granted me the freedom to establish research questions, explore the underlying truth, and provided timely guidance when I got lost. In addition, I would like to thank Dr. Walter Kosters for his insightful discussion and suggestions on mathematical and computer science algorithms. I really appreciate and value the scientific training and trust they had in me which help me build confidence in pursuing interesting projects and will have a positive effect on my future academic career.

Thirdly, I should acknowledge all my collaborators, for helping me complete my research stories. I want to thank Xiaofei Yang, Peter A. Audano, Christine R. Beck, Tobias Marschall, Xuefang Zhao and other members from the Human Genome Structural Variation Consortium (HGSVC) for their contribution to my complex structural variants detection algorithms. I also thank the HGSVC for providing me the opportunity to contribute to their studies and access to the highly valuable datasets they have created. I have to give thanks to my friends, who shared the happiness as well as helped me out of the hard time. Yongyong Kang, Peng Jia, Tingjie Wang, Tun Xu, Yuan Shen, Songbo Wang, Jing Hai, Yue Wang and many others, I'll never forget the time we played and studied together.

Last but not least, I should thank my parents for their support and love for me and for our small family in the past several years. I very much appreciate the understanding, encouragement, support, cooperation and love of my wife Minjun Xue, who also gave birth to our cute and lovely daughter.

Curriculum vitae

Jiadong Lin, born 1991 in Xi'an, China, received his BSc degree in Electrical Engineering from Xidian University, Xi'an, China in 2013. He graduated with a master's degree in Computer Science in 2015 at Michigan State University, Lansing, United States of America. In 2016, he started his research intern at the Department of Computational Medicine and Bioinformatics from the University of Michigan, Ann Arbor, United States of America. In 2017, he joined the Lab of Bio-data science of Dr. Kai Ye at Xi'an Jiaotong University (XJTU), China. In 2019, sponsored by the China Scholarship Council, he joined the double-PhD program of XJTU and Leiden University, the Netherlands where he was supervised by Dr. Walter Kosters at the Leiden Institute of Advanced Computer Science (LIACS). His research focuses on developing novel algorithms to detect and interpret complex genomic rearrangements from high-throughput-sequencing data.