Journal Pre-proofs

Method

Mako: A Graph-based Pattern Growth Approach to Detect Complex Structural Variants

Jiadong Lin, Xiaofei Yang, Walter Kosters, Tun Xu, Yanyan Jia, Songbo Wang, Qihui Zhu, Mallory Ryan, Li Guo, Chengsheng Zhang, The Human Genome Structural Variation Consortium, Charles Lee, Scott E. Devinel, Evan E. Eichler, Kai Ye

PII: S1672-0229(21)00143-1
DOI: https://doi.org/10.1016/j.gpb.2021.03.007
Reference: GPB 542

To appear in: *Genomics, Proteomics & Bioinformatics*

Received Date: 17 January 2021
Revised Date: 5 March 2021
Accepted Date: 5 March 2021

Please cite this article as: J. Lin, X. Yang, W. Kosters, T. Xu, Y. Jia, S. Wang, Q. Zhu, M. Ryan, L. Guo, C. Zhang, The Human Genome Structural Variation Consortium, C. Lee, S.E. Devinel, E.E. Eichler, K. Ye, Mako: A Graph-based Pattern Growth Approach to Detect Complex Structural Variants, *Genomics, Proteomics & Bioinformatics* (2021), doi: https://doi.org/10.1016/j.gpb.2021.03.007

Mako: A Graph-based Pattern Growth Approach to Detect Complex Structural Variants

Jiadong Lin[1,2,3,4,#], Xiaofei Yang[2,5,#], Walter Kosters[4], Tun Xu[1], Yanyan Jia[1], Songbo Wang[1], Qihui Zhu[6], Mallory Ryan[6], Li Guo[2,‡], Chengsheng Zhang[6,7], The Human Genome Structural Variation Consortium[‡], Charles Lee[6,7], Scott E. Devine[l,8], Evan E. Eichler[9,10,], Kai Ye[1,2,3,11,*]

[1]*School of Automation Science and Engineering, Faculty of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an 710049, China*

[2]*MOE Key Lab for Intelligent Networks & Networks Security, Faculty of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an 710049, China*

[3]*Genome Institute, the First Affiliated Hospital of Xi'an Jiaotong University, Xi'an 710061, China*

[4]*Leiden Institute of Advanced Computer Science, Faculty of Science, Leiden University, Leiden 2311EZ, Netherland*

[5]*School of Computer Science and Technology, Faculty of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an 710049, China*

[6]*The Jackson Laboratory for Genomic Medicine, Farmington, CT 06032, USA*

[7]*Precision Medicine Center, the First Affiliated Hospital of Xi'an Jiaotong University, Xi'an 710061, China*

[8]*Institute for Genome Sciences, University of Maryland School of Medicine, Baltimore, MD 21201, USA*

[9]*Department of Genome Sciences, University of Washington School of Medicine, Seattle, WA 98119-5065, USA*

[10]*Howard Hughes Medical Institute, University of Washington, Seattle, WA 98195, USA*

[11]*The School of Life Science and Technology, Xi'an Jiaotong University, Xi'an 710049, China*

[#]Equal contribution.

[‡]Consortium authors are enumerated at the end of this article..

*Corresponding author.

E-mail: kaiye@xjtu.edu.cn (Ye K).

Running title: *Lin J et al / Graph Based Complex Structural Variants Detection*

Total word counts (from "Introduction" to "Discussion"): 3859

Total figures: 6

Total tables: 3

Total supplementary figures: 23

Total supplementary tables: 10

Total supplementary files: 4

Total references: 53

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. As a result, there has been limited progress for CSV discovery compared with simple structural variants. 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.

KEYWORDS: Next-generation sequencing; Complex structural variants; Pattern growth; Graph mining; Formation mechanism

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 [1–7]. 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 1**A). This model-match strategy has been proved effective for detecting simple SVs, providing us with prominent opportunities to study and understand genome evaluation and disease progression [8–11]. However, recent research has revealed that some rearrangements have multiple, compounded mutational signals and usually cannot fit into the simple SV models [8,12–16] (Figure 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 [8]. 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 [17]. 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* [16]. Though less frequently reported compared with simple SVs, these multiple breakpoint rearrangements were considered as punctuated events, leading to severe genome alterations at once [10,18–21]. This dramatic change of genome provided distinctive evidence to study formation mechanisms of rearrangement and to understand cancer genome evolution [13,14,17,19,21–25].

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 [8,12,16]. 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 [26] proposed sophisticated abnormal alignment models to depict the mutational signals reflected by 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 [27] 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. Whereas GRIDSS [28] represents the assembly-based approach, which detected 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 1B). Moreover, the graph is another approach that has been widely used for simple [2,29] and complex [19,30] SV detection. Notably, ARC-SV [30] 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 study, we proposed 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.

Method

Overview of Mako

Given that a CSV is a single event with multiple breakpoint connections, the breakpoints in the current CSV shall not connect 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**). 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 $V = \{v_1, v_2, ..., v_n\}$ and $E = \{E_{pe}, E_{ae}\}$. Each signal node $v \in V$ is represented as $v = (type, pos, weight)$, 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 $e_{pe} = (v_i, v_j, rp)$ 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 paired-reads or split-reads (*sr*). E_{ae} indicates the adjacent edges induced from the reference genome, connecting two adjacent signal nodes of 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 CSVs types are determined by the edge connection types of the corresponding subgraphs (Figure 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 [4] to avoid using randomly occurred discordant alignment (File S1). Additionally, edges are created alone with the signal nodes, where multiple types of edges might co-exist between two nodes.

Detecting CSVs with pattern growth

Pattern growth has been widely used in many areas [31–36], such as Indel detection in DNA sequences [1,24]. 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). Specifically, 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. Of note, pattern growth via adjacent edges is conditional to the distance constrain (*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).

Given that the signal graph contains millions of nodes at the whole genome scale, we adopt the "seed-and-extension" [37,38] 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). 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, a toy example of excuting the algorithm is shown in Figure S1).

Algorithm I: Detect maximal subgraphs

Input: Signal graph $G = (V, E)$, **parameters** $minFreq$, $minDist$

Output: A set of CSV subgraphs $O = \{g_1, g_2, \dots, g_n\}$, with $freq(g_i) \geq minFreq$

procedure $findMaximalSubgraph(G, minFreq, minDist)$
Initialize $freq\_types$ equals to $type$ frequency of node in V;
Build index-projection $G|_\emptyset$ of G;
for α **in** $freq\_types$ **do**:
    Build index-projection $G|_\alpha$;
    $g_i = \alpha$;
    if $freq(g_i) > minFreq$ **then**
        $multiLocPatternGrowth(O, g_i, G|_\alpha, minFreq, minDist)$;
    end if
end for
end procedure

Algorithm II: Multi-location subgraph growth

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**
        $g' = g + node$;
        $O.append(g')$;
        $multiLocPatternGrowth(O, g', G|_{g'}, minFreq, minDist)$;
    end if
end for
end procedure
procedure $nodeInRange(g, v)$
    Set the nodes in g with respect increasing order of pos value: $v_0, v_1, \dots, v_n$;
    Set $v' = v_n$;
    if $freq(v) > minFreq$ **then**
        if $dist(v', v) < minDist$ **then**
            return True
        else:
            for $i = n$ **to** 0 **do**
                if $\exists\ e_{pe}$ between v and v_i **then**
                    return True
        end if
    end if
    return False
end procedure

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, 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 criteria, 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 (File S1).

Preparing CSV benchmarks for performance evaluation

In this study, we use both simulated and real CSVs to benchmark the performance of different callers. We follow the workflow introduced by the Sniffles [39] to create simulated CSVs (Figure S2). Firstly, VISOR [40] 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 [38]. According to the above-generalized simulation procedures, we create reported CSV types published by previous studies [8,17] and randomized CSV types (File S1).

In terms of the real data, we are not aware of any public CSV benchmarks due to the breakpoint complexity and underdeveloped methods [8,12,27,41,42]. Fortunately, PacBio reads could span multiple breakpoints of CSVs, providing direct evidence to

validate CSVs through sequence Dotplot [43]. 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 (Figure S3 and S4). We then use the threshold that could produce the most clusters for merging clusters, which could potentially reduce the number of missed CSVs (Table S1, Figure S5 and S6). 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.

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 (Figure S7). 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 (Figure S8). The obtained Sanger sequences are aligned against the reference allele of the CSV site and visualized with Gepard for breakpoint inspection (File S1).

As for the computational validation, two orthogonal data obtained from Human Genome Structural Variant Consortium (HGSVC) are used, *e.g.*, Oxford Nanopore sequencing (ONT) and HiFi contigs. We first apply VaPoR [44] on the ONT reads to validate CSVs, referring 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 (Figure S9, File S1).

Furthermore, we manually curate detected CSVs via Dotplots created by Gepard (Figure S10), which is similar to the procedure of creating the benchmark CSV for real data (File S1). For CSVs at highly repetitive regions, we further validate them according

to specific patterns (Figures S11–S13).

Results

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 3**A) and homozygous (HOM) (Figure 3B) 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 3A and B) 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.

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 3C and D), 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 3E F), which was consistent with our previous observation (Figure 3A and B). In particular, at 30x coverage, Mako detected 203% more HET CSVs than that of SVelter (Figure 3E), probably due to the complementary graph edges for accurate CSV site discovery.

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 (Figures S14 and S15), and we found that Mako shared most calls with GRIDSS (**Figure 4**A and B), which was consistent with our observation in simulated data (Figure 3). Furthermore, we examined the discovery completeness of 59 (NA19240) and 21 (SKBR3) benchmark CSVs (**Table 1**, File S2, Table S2). 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 4C). 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 (Table 1). 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 was better performed than that of either randomized model (SVelter) or specific model (TARDIS) with few computational resources (Figure S16).

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 5**A, Figure S17, Table S3). We systemically evaluated all CSV events in HG00733 via experimental and computational validation as well as manual inspection (File S3). For experimental validation, we successfully designed primers for 107 CSVs (Table S4), where 15 out of 21 (71%, **Table 2**) were

successfully amplified and validated by Sanger sequencing (File S4, Table S5, breakpoint details in Table S6). The computational validation (Figure S4, Table S5, breakpoint details of HiFi contigs in Table S7, details of VaPoR validation in Table S8) showed up to 87% accuracy, indicating a combination of methods and external data is necessary for comprehensive CSV validation (**Table 3**). Further analysis showed that the medians of breakpoint shift were 13bp and 26bp comparing to breakpoints given by experimental and computational evaluation (Figure S18). 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 5A). For the connection types, more 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 5B), 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 5C) and chr5:151,511,018–151,516,780 (Figure 5D) 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 (Table S9). 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, from which we can characterize the breakpoint connections for a given CSV event.

Contribution of homology sequence in CSV formation

Given 1568 detected CSVs from three genomes, we further investigated the formation mechanisms of these CSVs. Ongoing studies have revealed that inaccurate DNA repair and the 2-33 bp long microhomology sequence at breakpoint junctions play an important role in CSV formation [18, 45-48]. To further characterize CSVs' internal structure and examine the impact of homology sequence on CSV formation, we

manually reconstructed 1052 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 6**A, Figure S19, Table S10, File S3). 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 (Figure S20). Besides, we found two novel types, which were named adjacent segments swap and tandem dispersed duplication (Figure 6B, Figures S21 and S22). 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 (Table S10). These duplications involved CSVs were mainly caused by microhomology mediated break-induced replication (MMBIR) according to previous studies[18,46,49]. It was known that different homology patterns cause distinct CSV types (Figure 6C and 6D). Surprisingly, one particular pattern of homology sequence yielded multiple CSV types (Figure 6E). In particular situations of the three different homology patterns, DNA double strand break (DSB) occurred after replication of fragment *C*. According to the MMBIR mechanism and template switch [23,46–48], the pattern I (Figure 6C) and pattern II (Figure 6D) yield one output, but pattern III (Figure 6E) 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 [50,51].

Discussion

Currently, short-read sequencing is significantly reduced in cost and has been applied to clinical diagnostics and large cohort studies [16,52,53]. 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 [13,14,39], 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 (Figure S23). 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 20 bp 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.

Code availability

Mako is implemented in Java 1.8, and it is available at https://github.com/xjtu-omics/Mako. It is free for non-commercial use by academic, government, and non-profit/not-for-profit institutions. A commercial version of the software is available and licensed through Xi'an Jiao-tong University. All scripts used in this study are also included in the Github repository, and a detailed description of using these scripts and other tools is provided in File S1.

Data availability

All materials or datasets used in this study are publicly available, and their links are listed in File S1.

Credit author statement

Jiadong Lin: Algorithm design, Software development, Sample analysis, Manual validation, Writing. **Xiaofei Yang:** Algorithm design, Writing. **Walter Kosters:** Algorithm design, Writing. **Tun Xu:** Sample analysis. **Yanyan Jia:** Experimental validation. **Songbo Wang:** Manual and computational validation. **Qihui Zhu:** Experimental validation. **Mallory Ryan:** Experimental validation. **Li Guo:** Writing.

Chengsheng Zhang: Experimental validation, Writing. **HGSVC:** Resources. **Charlse Lee:** Producing data, Writing. **Scott Devine:** Producing data. **Evan Eichler:** Producing data. **Kai Ye:** Conceptualization, Resources, Supervision, Project administration, Funding acquisition. All authors read and approved the final manuscript.

Competing interests

The authors have declared no competing interests.

Acknowledgments

This study was supported by the National Key R&D Program of China (Grand Nos. 2018YFC0910400 and 2017YFC0907500), the National Science and Technology Major Project of China (Grand No. 2018ZX10302205), the National Science Foundation of China (Grand NO. 31671372, 61702406, and 31701739) and the "World-Class Universities and the Characteristic Development Guidance Funds for the Central Universities". Supported by Shanghai Municipal Science and Technology Major Project (Grant No. 2017SHZDZX01).

Authors from HGSVC

Mark B. Gerstein[1], Ashley D. Sanders[2], Micheal C. Zody[3], Michael E. Talkowski[4], Ryan E. Mills[5], Jan O. Korbel[2], Tobias Marschall[6], Peter Ebert[6], Peter A. Audano[7], Bernardo Rodriguez-Martin[8], David Porubsky[7], Marc Jan Bonder[8,9], Arvis Sulovari[7], Jana Ebler[6], Weichen Zhou[5], Rebecca Serra Mari[6], Feyza Yilmaz[10], Xuefang Zhao[4], PingHsun Hsieh[7], Joyce Lee[11], Sushant Kumar[1], Tobias Rausch[8], Yu Chen[12], Zechen Chong[12], Katherine M. Munson[7], Mark J.P. Chaisson[13], Junjie Chen[14], Xinghua Shi[14], Aaron M. Wenger[15], William T. Harvey[7], Patrick Hansenfeld[8], Allison Regier[16], Ira M. Hall[16], Paul Flicek[17], Alex R. Hastie[11], Susan Fairely[17]

[1]*Program in Computational Biology and Bioinformatics, Yale University, BASS 432&437, 266 Whitney Avenue, New Haven, CT 06520, USA*

[2]*European Molecular Biology Laboratory (EMBL), Genome Biology Unit, Meyerhofstr. 1, 69117 Heidelberg, Germany*

[3]*New York Genome Center, New York, NY 10013, USA*

[4]*Center for Genomic Medicine, Massachusetts General Hospital, Department of Neurology, Harvard Medical School, Boston, MA 02114, USA*

[5]*Department of Computational Medicine & Bioinformatics, University of Michigan, 500 S. State Street, Ann Arbor, MI 48109, USA*

[6]*Heinrich Heine University, Medical Faculty, Institute for Medical Biometry and Bioinformatics, Moorenstr. 20, 40225 Düsseldorf, Germany*

[7]*Department of Genome Sciences, University of Washington School of Medicine, 3720 15th Ave NE, Seattle, WA 98195-5065, USA*

[8]*European Molecular Biology Laboratory (EMBL), Genome Biology Unit, Meyerhofstr. 1, 69117 Heidelberg, Germany*

[9]*Division of Computational Genomics and Systems Genetics, German Cancer Research Center (DKFZ), 69120 Heidelberg, Germany*

[10]*The Jackson Laboratory for Genomic Medicine, 10 Discovery Dr, Farmington, CT 06030, USA*

[11]*Bionano Genomics, San Diego, CA 92121, USA*

[12]*Department of Genetics and Informatics Institute, School of Medicine, University of Alabama at Birmingham, Birmingham, AL 35294, USA*

[13]*Molecular and Computational Biology, University of Southern California, Los Angeles, CA 90089, USA*

[14]*Department of Computer & Information Sciences, Temple University, Philadelphia, PA 19122, USA*

[15]*Pacific Biosystems of California, Inc., Menlo Park, CA 94025, USA*

[16]*Washington University, St. Louis, MO 63108, USA*

[17]*European Molecular Biology Laboratory, European Bioinformatics Institute, Wellcome Genome Campus, Hinxton, Cambridge, CB10 1SD, United Kingdom*

ORCID

0000-0002-8116-5901 (Jiadong Lin)

0000-0002-5118-7755 (Xiaofei Yang)

0000-0001-8860-0390 (Walter Kosters)

0000-0003-3194-1834 (Tun Xu)

0000-0002-4966-0574 (Yanyan Jia)

0000-0003-4482-8128 (Songbo Wang)

0000-0003-2401-8443 (Qihui Zhu)

0000-0001-5428-0018 (Mallory Ryan)

0000-0001-6100-3481 (Li Guo)
0000-0002-5238-083X (Chengsheng Zhang)
0000-0001-7317-6662 (Charles Lee)
0000-0001-7629-8331 (Scott E. Devine)
0000-0002-8246-4014 (Evan E. Eichler)
0000-0002-2851-6741 (Kai Ye)
0000-0002-9746-3719 (Mark B. Gerstein)
0000-0003-3945-0677 (Ashley D. Sanders)
0000-0001-6594-7199 (Michael C. Zody)
0000-0003-2889-0992 (Michael E. Talkowski)
0000-0003-3425-6998 (Ryan E. Mills)
0000-0002-2798-3794 (Jan O. Korbel)
0000-0002-9376-1030 (Tobias Marschall)
0000-0001-7441-532X (Peter Ebert)
0000-0002-5187-0415 (Peter A. Audano)
0000-0003-4693-3140 (Bernardo Rodriguez-Martin)
0000-0001-8414-8966 (David Porubsky)
0000-0002-8431-3180 (Marc Jan Bonder)
0000-0003-4354-9020 (Arvis Sulovari)
0000-0002-0382-3702 (Jana Ebler)
0000-0003-4755-1072 (Weichen Zhou)
0000-0002-2812-9653 (Rebecca Serra Mari)
0000-0001-8795-5800 (Feyza Yilmaz)
0000-0003-4036-9577 (Xuefang Zhao)
0000-0001-8294-6227 (PingHsun Hsieh)
0000-0002-3492-1102 (Joyce Lee)
0000-0002-2294-3988 (Sushant Kumar)
0000-0001-5773-5620 (Tobias Rausch)
0000-0002-2037-7337 (Yu Chen)
0000-0001-5750-1808 (Zechen Chong)
0000-0001-8413-6498 (Katherine M. Munson)
0000-0001-5395-1457 (Mark J.P. Chaisson)
0000-0002-0483-303X (Junjie Chen)
0000-0003-4662-3177 (Xinghua Shi)

0000-0003-1183-0432 (Aaron M. Wenger)
0000-0003-0646-7528 (William T. Harvey)
0000-0003-2319-2482 (Patrick Hasenfeld)
0000-0002-1932-8714 (Allison A. Regier)
0000-0003-4442-6655 (Ira M. Hall)
0000-0002-3897-7955 (Paul Flicek)
0000-0001-5829-2649 (Alex R. Hastie)
0000-0001-9425-0788 (Susan Fairley)

References

[1] 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;25:2865–71.

[2] 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:i333–9.

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

[4] Chen K, Wallis JW, McLellan MD, Larson DE, Kalicki JM, Pohl CS, et al. BreakDancer: an algorithm for high-resolution mapping of genomic structural variation. Nat Methods 2009;6:677–81.

[5] 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.

[6] 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.

[7] Chen X, Schulz-Trieglaff O, Shaw R, Barnes B, Schlesinger F, Kallberg M, et al. Manta: rapid detection of structural variants and indels for germline and cancer sequencing applications. Bioinformatics 2016;32:1220–2.

[8] Sudmant PH, Rausch T, Gardner EJ, Handsaker RE, Abyzov A, Huddleston J, et al. An integrated map of structural variation in 2,504 human genomes. Nature 2015;526:75–81.

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

[10] Gao R, Davis A, McDonald TO, Sei E, Shi X, Wang Y, et al. Punctuated copy number evolution and clonal stasis in triple-negative breast cancer. Nat Genet 2016;48:1119–30.

[11] Yates LR, Knappskog S, Wedge D, Farmery JHR, Gonzalez S, Martincorena I, et al. Genomic evolution of breast cancer metastasis and relapse. Cancer Cell 2017;32:169–84 e7.

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

[13] Nattestad M, Goodwin S, Ng K, Baslan T, Sedlazeck FJ, Rescheneder P, et al. Complex rearrangements and oncogene amplifications revealed by long-read DNA and RNA sequencing of a breast cancer cell line. Genome Res 2018;28:1126–35.

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

[15] Greer SU, Nadauld LD, Lau BT, Chen J, Wood-Bouwens C, Ford JM, et al. Linked read sequencing resolves complex genomic rearrangements in gastric cancer metastases. Genome Med 2017;9:57.

[16] Lee JJ, Park S, Park H, Kim S, Lee J, Lee J, et al. Tracing oncogene rearrangements in the mutational history of lung adenocarcinoma. Cell 2019;177:1842–57 e21.

[17] Collins RL, Brand H, Redin CE, Hanscom C, Antolik C, Stone MR, et al. Defining the diverse spectrum of inversions, complex structural variation, and chromothripsis in the morbid human genome. Genome Biol 2017;18:36.

[18] Carvalho CM, Lupski JR. Mechanisms underlying structural variant formation in genomic disorders. Nat Rev Genet 2016;17:224–38.

[19] Baca SC, Prandi D, Lawrence MS, Mosquera JM, Romanel A, Drier Y, et al. Punctuated evolution of prostate cancer genomes. Cell 2013;153:666–77.

[20] Korbel JO, Campbell PJ. Criteria for inference of chromothripsis in cancer genomes. Cell 2013;152:1226–36.

[21] Sanders AD, Meiers S, Ghareghani M, Porubsky D, Jeong H, van Vliet M, et al. Single-cell analysis of structural variations and complex rearrangements with tri-channel processing. Nat Biotechnol 2020;38:343–54.

[22] Carvalho CMB, Lupski JR. Mechanisms underlying structural variant formation in genomic disorders. Nature Reviews Genetics 2016;17:224–38.

[23] Malhotra A, Lindberg M, Faust GG, Leibowitz ML, Clark RA, Layer RM, et al. Breakpoint profiling of 64 cancer genomes reveals numerous complex rearrangements spawned by homology-independent mechanisms. Genome Res 2013;23:762–76.

[24] Ye K, Wang J, Jayasinghe R, Lameijer EW, McMichael JF, Ning J, et al. Systematic discovery of complex insertions and deletions in human cancers. Nat Med 2016;22:97–104.

[25] Zhang CZ, Leibowitz ML, Pellman D. Chromothripsis and beyond: rapid genome evolution from complex chromosomal rearrangements. Genes Dev 2013;27:2513–30.

[26] Soylev A, Le TM, Amini H, Alkan C, Hormozdiari F. Discovery of tandem and interspersed segmental duplications using high-throughput sequencing. Bioinformatics 2019;35:3923–30.

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

[28] Cameron DL, Schroder J, Penington JS, Do H, Molania R, Dobrovic A, et al. GRIDSS: sensitive and specific genomic rearrangement detection using positional de Bruijn graph assembly. Genome Res 2017;27:2050–60.

[29] Marschall T, Costa IG, Canzar S, Bauer M, Klau GW, Schliep A, et al. CLEVER: clique-enumerating variant finder. Bioinformatics 2012;28:2875–82.

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

[31] Liao VCC, Chen MS. DFSP: a Depth-First SPelling algorithm for sequential pattern mining of biological sequences. Knowl Inf Syst 2014;38:623–39.

[32] Tsai HP, Yang DN, Chen MS. Mining group movement patterns for tracking moving objects efficiently. IEEE T Knowl Data En 2011;23:266–81.

[33] Huang Y, Zhang LQ, Zhang PS. A framework for mining sequential patterns from spatio-temporal event data sets. IEEE T Knowl Data En 2008;20:433–48.

[34] Ye K, Kosters WA, Ijzerman AP. An efficient, versatile and scalable pattern growth approach to mine frequent patterns in unaligned protein sequences. Bioinformatics 2007;23:687–93.

[35] Pei J, Han J, Wang W. Constraint-based sequential pattern mining: the pattern-growth methods. J Intell Inf Syst 2007;28:133–60.

[36] Pei J, Han JW, Mortazavi-Asl B, Wang JY, Pinto H, Chen QM, et al. Mining sequential patterns by pattern-growth: the PrefixSpan approach. IEEE T Knowl Data En 2004;16:1424–40.

[37] Li H, Homer N. A survey of sequence alignment algorithms for next-generation sequencing. Brief Bioinform 2010;11:473–83.

[38] Li H, Durbin R. Fast and accurate short read alignment with Burrows-Wheeler transform. Bioinformatics 2009;25:1754–60.

[39] Sedlazeck FJ, Rescheneder P, Smolka M, Fang H, Nattestad M, von Haeseler A, et al. Accurate detection of complex structural variations using single-molecule sequencing. Nat Methods 2018;15:461–8.

[40] 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;36:1267–9.

[41] 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;22:2250–61.

[42] Dzamba M, Ramani AK, Buczkowicz P, Jiang Y, Yu M, Hawkins C, et al. Identification of complex genomic rearrangements in cancers using CouGaR. Genome Res 2017;27:107–17.

[43] Delcher AL, Phillippy A, Carlton J, Salzberg SL. Fast algorithms for large-scale genome alignment and comparison. Nucleic Acids Res 2002;30:2478–83.

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

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

[46] Kramara J, Osia B, Malkova A. Break-induced replication: the where, the why, and the how. Trends Genet 2018;34:518–31.

[47] 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.

[48] 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;22:2642–51.

[49] Yang L, Luquette LJ, Gehlenborg N, Xi R, Haseley PS, Hsieh CH, et al. Diverse mechanisms of somatic structural variations in human cancer genomes. Cell 2013;153:919–29.

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

[51] Allen F, Crepaldi L, Alsinet C, Strong AJ, Kleshchevnikov V, De Angeli P, et al. Predicting the mutations generated by repair of Cas9-induced double-strand breaks. Nat Biotechnol 2019;37:64–72.

[52] Quigley DA, Dang HX, Zhao SG, Lloyd P, Aggarwal R, Alumkal JJ, et al. Genomic hallmarks and structural variation in metastatic prostate cancer. Cell 2018;175:889.

[53] Fraser M, Sabelnykova VY, Yamaguchi TN, Heisler LE, Livingstone J, Huang V, et al. Genomic hallmarks of localized, non-indolent prostate cancer. Nature 2017;541:359–64.

Figure legends

Figure 1 Explanation of simple and complex structure variants alignment models derived from abnormal read-pairs

A. Three common simple SV 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 signature that can be matched by a simple SV alignment model.

Figure 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 E_{ae} edge and two E_{pe} edges of type Del and Inv.

Figure 3 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 (y-axis), precision (x-axis) and F1-score (dotted lines). The right top corner of the plot indicates better performance. The c5–c30 indicates coverage, *e.g.*, c5 indicates 5× coverage.

Figure 4 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.

Figure 5 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 is the forward-forward alignments, green is the reverse-forward alignments, and red indicates the alignment of large insert size. The bottom panel 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 alignment are shown on the reference genome with the corresponding ID in the circle. Noted that a single ID may have more than one corresponding abnormal alignment types on the reference.

Figure 6 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 less 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 diagram explains 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

replicate twice, as shown in (D).

Tables

Table 1 Summary of benchmark CSVs

Table 2 Summary of experimentally validated CSVs

Table 3 Summary of experimental and computational validation as well as

Supplementary material

File S1 Supplementary note for Mako

File S2 IGV view and PacBio reads Dotplot of each benchmark CSVs

File S3 Dotplot used for manually inspection of CSVs from HG00733

File S4 PCR results and visualization of CSV breakpoint validated through Sanger sequencing

Figure S1 A toy example to explain the pattern growth process

Figure S2 Workflow of CSV simulation

Figure S3 Hierarchical clustering tree view of SVs from NA19240 chromosome 1

Figure S4 Hierarchical clustering tree view of SVs from SKBR3 chromosome 1

Figure S5 The curve plot between cluster distance cutoff and number of clusters for SVs from NA19240 autosomes

Figure S6 The curve plot between cluster distance cutoff and number of clusters for SVs from SKBR3 autosomes

Figure S7 Diagram of selecting primers for each CSV

Figure S8 Examples of PCR electrophoretic bands visualized under the UV light

Figure S9 Workflow of HiFi assembly K-mer validation

Figure S10 A screenshot using Gepard to investigate a deletion associated with inversion event

Figure S11 Dotplot patterns used to identify CSVs at highly repetitive regions

Figure S12 Dotplot patterns used to identify SVs at highly repetitive regions

Figure S13 Example call at high repetitive regions that labeled as NA by VaPoR at chr6:165,749,273-165,749,500

Figure S14 Size distribution of SV in the range [50bp, 10Kbp] from NA19240

Figure S15 Size distribution of SV in the range [50bp, 10Kbp] from SKBR3 breast cancer cell line

Figure S16 Running time comparison between different methods

A. Runtime comparison on simulated data at 30× coverage. **B.** Runtime of Mako on real data at different coverage. The time baseline is decided by copying the original BAM to another location. **C.** Memory usage of Mako on real data at different coverage.

Figure S17 Repeat annotation and connection types of Mako detected CSVs from three samples

A. Repeat annotation of CSVs detected from three genomes. **B.** Mako predicted CSV types of three genomes.

Figure S18 Mako detected CSV breakpoint resolution compared to HiFi contig (K-mer) and experiment

Figure S19 Mako detected CSV and PacBio HiFi read refined CSV size distribution

Figure S20 Example of an insertion associated with duplication event (InsDup) at chr6:165,749,273-165,749,500

Figure S21 The IGV view and sequence dot-plot of the adjacent segment swap from NA19240 at Chr7:83,316,809-83,317,466

Figure S22 The IGV view and sequence dot-plot of the tandem dispersed duplication from NA19240 at Chr17:43,359,104-43,365,253

Figure S23 Examples to show the difference of CSV breakpoints from single haplotype or two haplotypes

A. Diagram of two simple SVs at different haplotypes. **B.** Diagram of complex SV at the same haplotype

Table S1 Parameters used for creating the CSV benchmarks for NA19240 and SKBR3

Table S2 CSV benchmarks for NA19240 and SKBR3

Table S3 Mako detected CSVs for HG00733, HG00514, and NA19240

Table S4 CSVs of successfully designed primers

Table S5 Summary of experimental and computational validation as well as manual inspections of HG00733

Table S6 Comparing Mako detected breakpoints with PCR validated

breakpoints

Table S7 Comparing Mako breakpoints with K-mer realigned breakpoints

Table S8 Details of VaPoR validation results

Table S9 Details of breakpoints for the two examples in Figure 5

Table S10 Summary of PacBio HiFi reads refined CSV types

Credit author statement

Jiadong Lin: Algorithm design, Software development, Sample analysis, Manual validation, Writing. **Xiaofei Yang:** Algorithm design, Writing. **Walter Kosters:** Algorithm design, Writing. **Tun Xu:** Sample analysis. **Yanyan Jia:** Experimental validation. **Songbo Wang:** Manual and computational validation. **Qihui Zhu:** Experimental validation. **Mallory Ryan:** Experimental validation. **Li Guo:** Writing. **Chengsheng Zhang:** Experimental validation, Writing. **HGSVC:** Resources. **Charlse Lee:** Producing data, Writing. **Scott Devine:** Producing data. **Evan Eichler:** Producing data. **Kai Ye:** Conceptualization, Resources, Supervision, Project administration, Funding acquisition. All authors read and approved the final manuscript.

Table 1 Summary of benchmark CSVs

Type	Benchmark summaries		Description
	NA19240	SKBR3	
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

DelSpaDel	8	1	Two deletions with inverted or non-inverted spacer
TanDisdup	1	-	Tandem dispersed duplications

Table 2 Summary of experimentally validated CSVs

Chromosome	Start	End	Mako Type
Chr1	81,194,398	81,195,874	DEL, INV
Chr2	119,659,504	119,661,322	DUP, INS
Chr3	146,667,093	146,667,284	DEL, DUP
Chr5	141,480,327	141,483,116	DEL, DUP
Chr7	1,940,931	1,941,009	DUP, INS
Chr9	29,591,409	29,593,057	DEL, INV
Chr10	14,568,488	14,568,677	DUP, INS
Chr12	71,315,482	71,316,928	DEL, INV
Chr12	77,989,900	77,994,324	DEL, INV
Chr13	74,340,759	74,342,810	DEL, DUP
Chr16	78,004,459	78,007,456	DEL, DUP
Chr17	34,854,438	34,855,851	DEL, INV
Chr17	48,538,270	48,540,171	DEL, DUP
Chr18	72,044,575	72,045,937	DEL, DUP
Chr21	26,001,844	26,001,844	DEL, INV

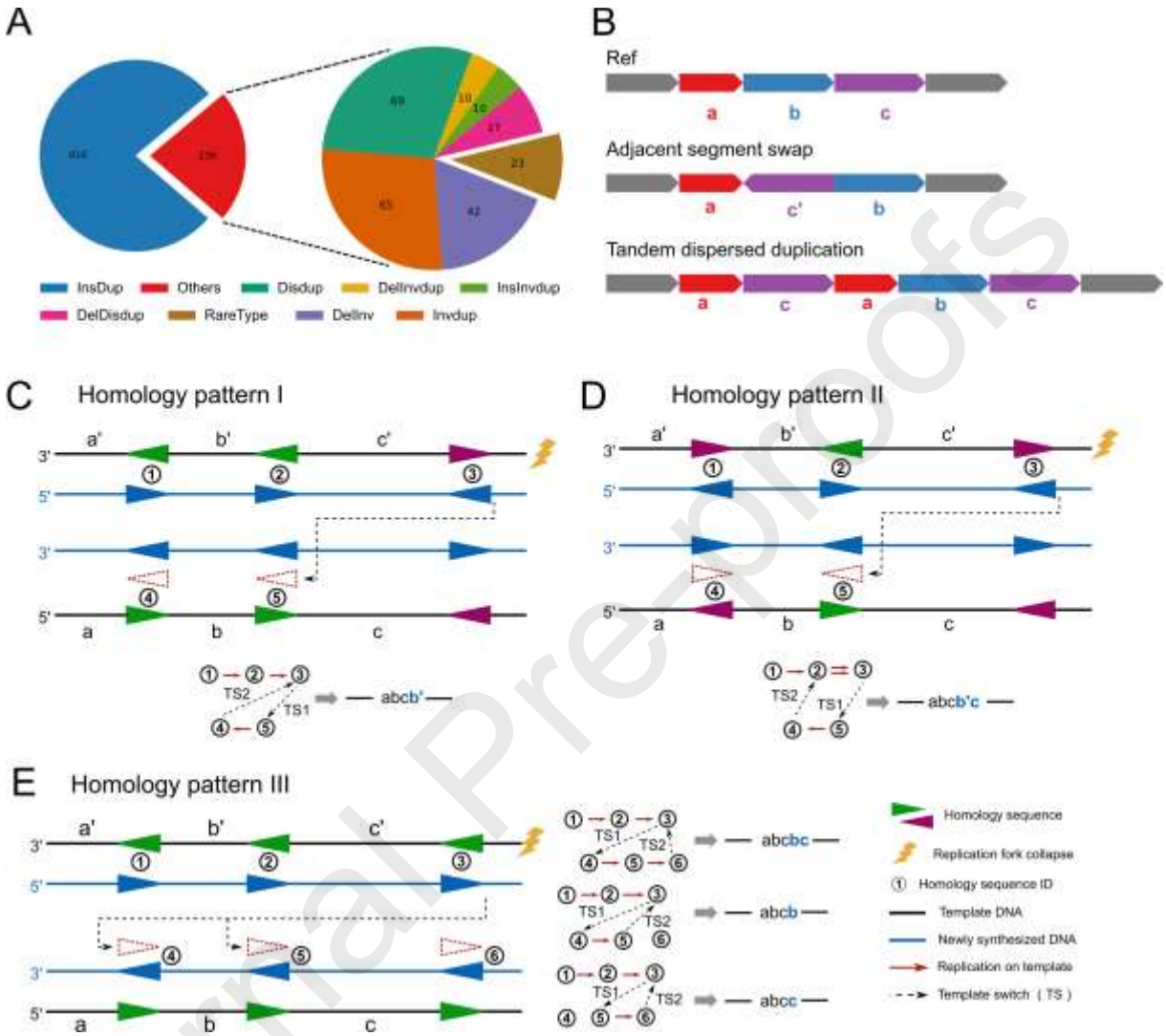
Note: DEL, deletion; INS, insertion; DUP, duplication; INV, inversion.

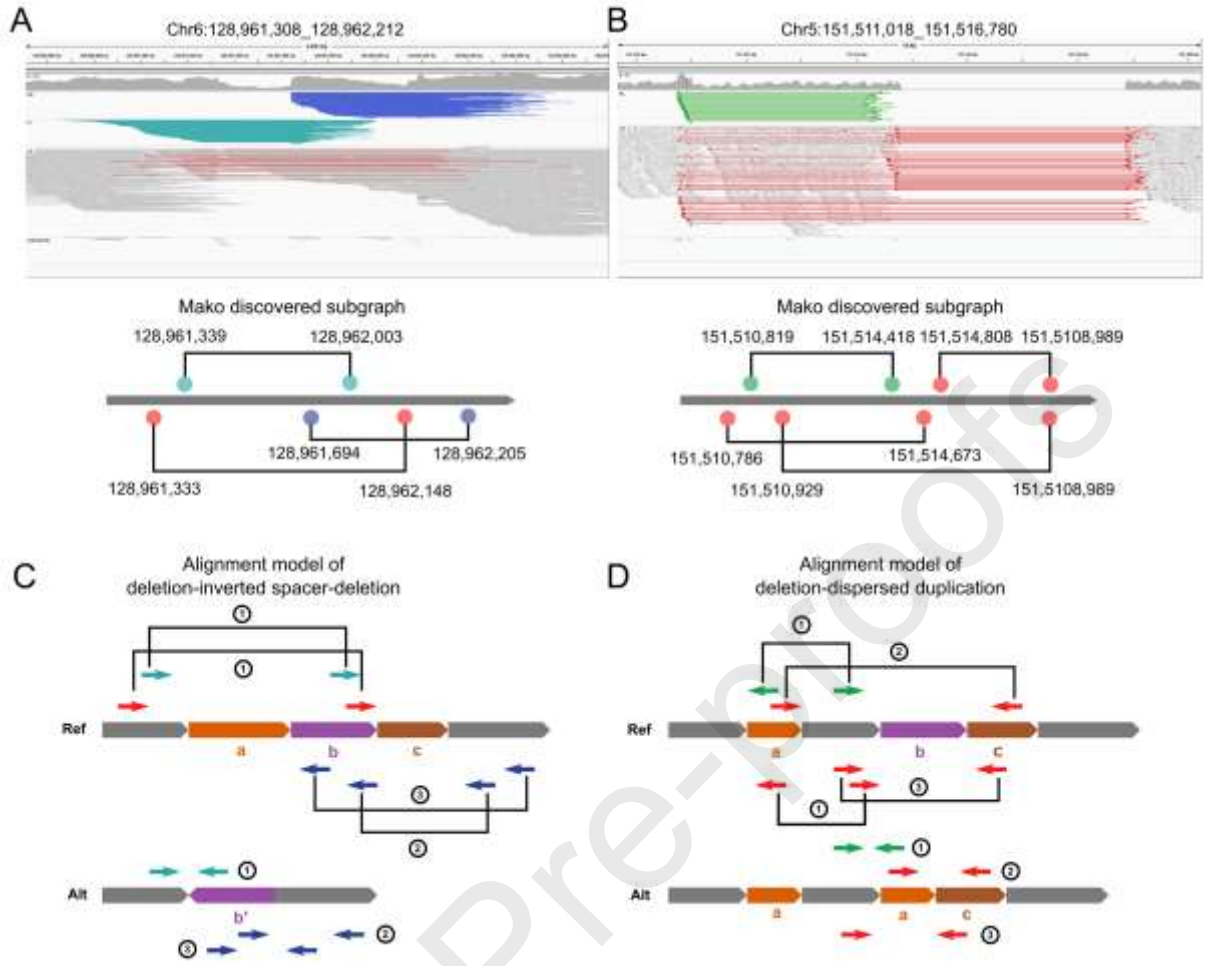
Table 3 Summary of experimental and computational validation as well as manual inspection for CSVs

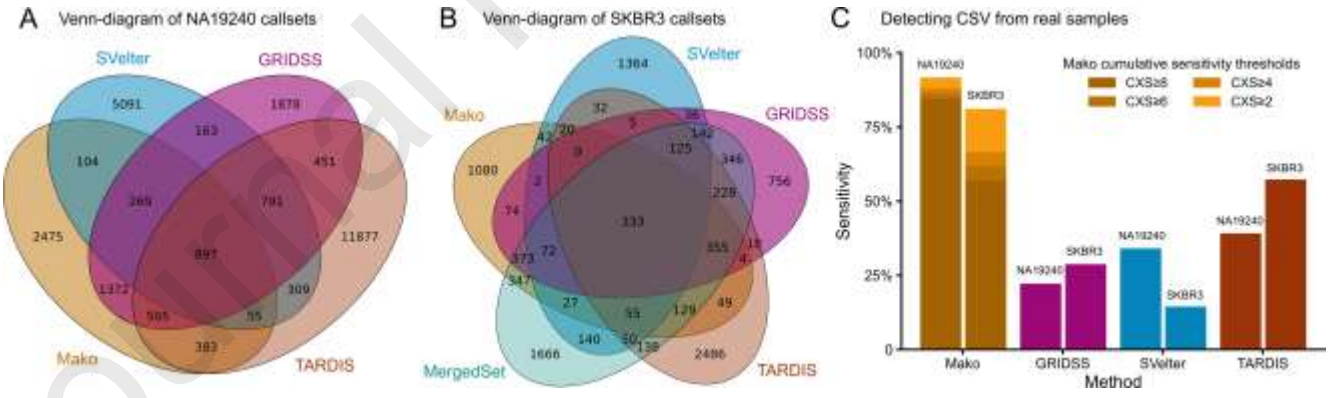
Validation Strategy		Total	Valid	Invalid	Inconclusive
Experimental (PCR succeeded)		21	15 (71%)	6 (29%)	-
Computational	ONT reads	609	256 (42%)	-	353 (58%)
	HiFi contig		414 (68%)	191 (32%)	-

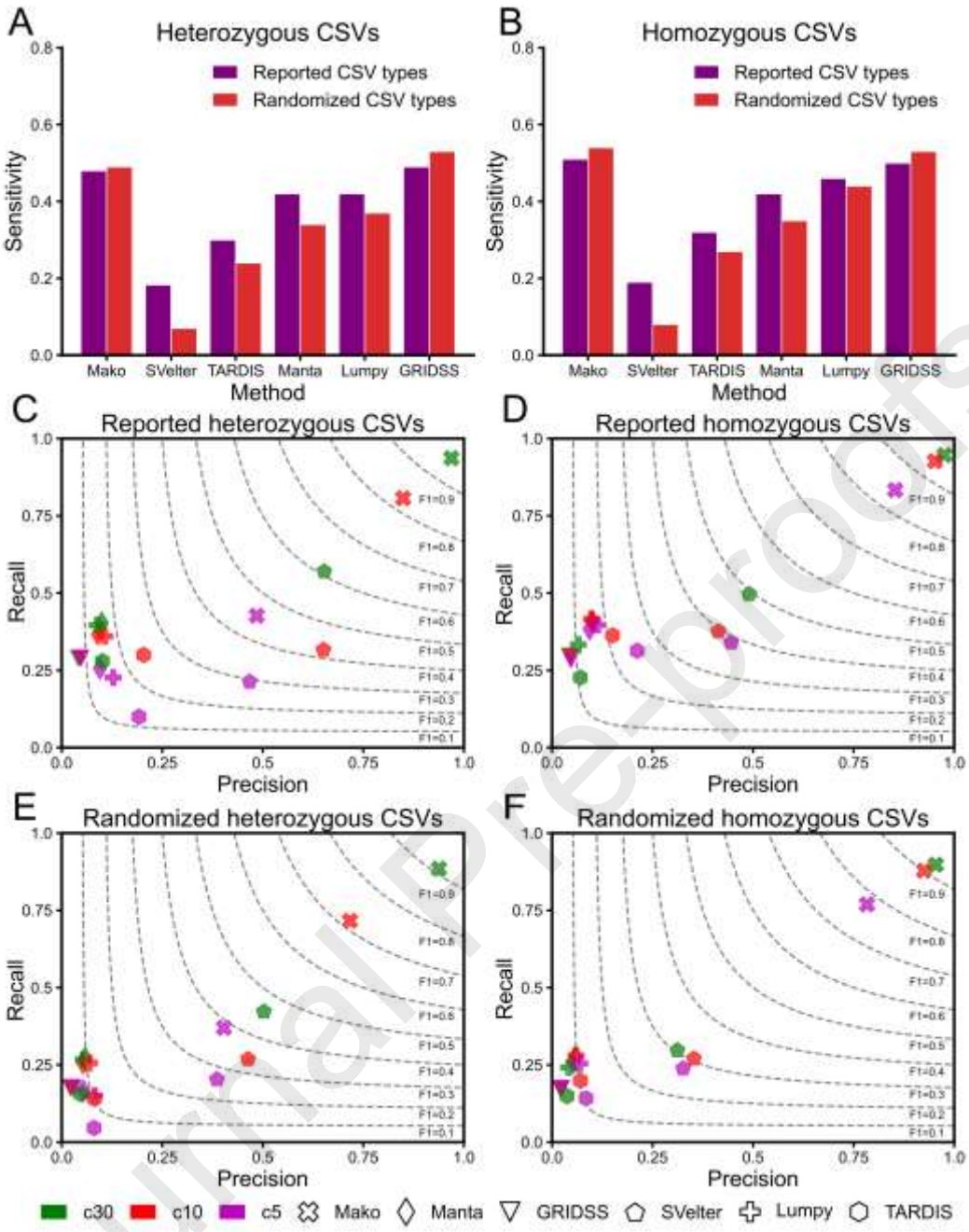
	ONT reads or HiFi contig		544 (87%)	76 (13%)	-
Manual	HiFi reads	609	440 (72%)	169 (28%)	-

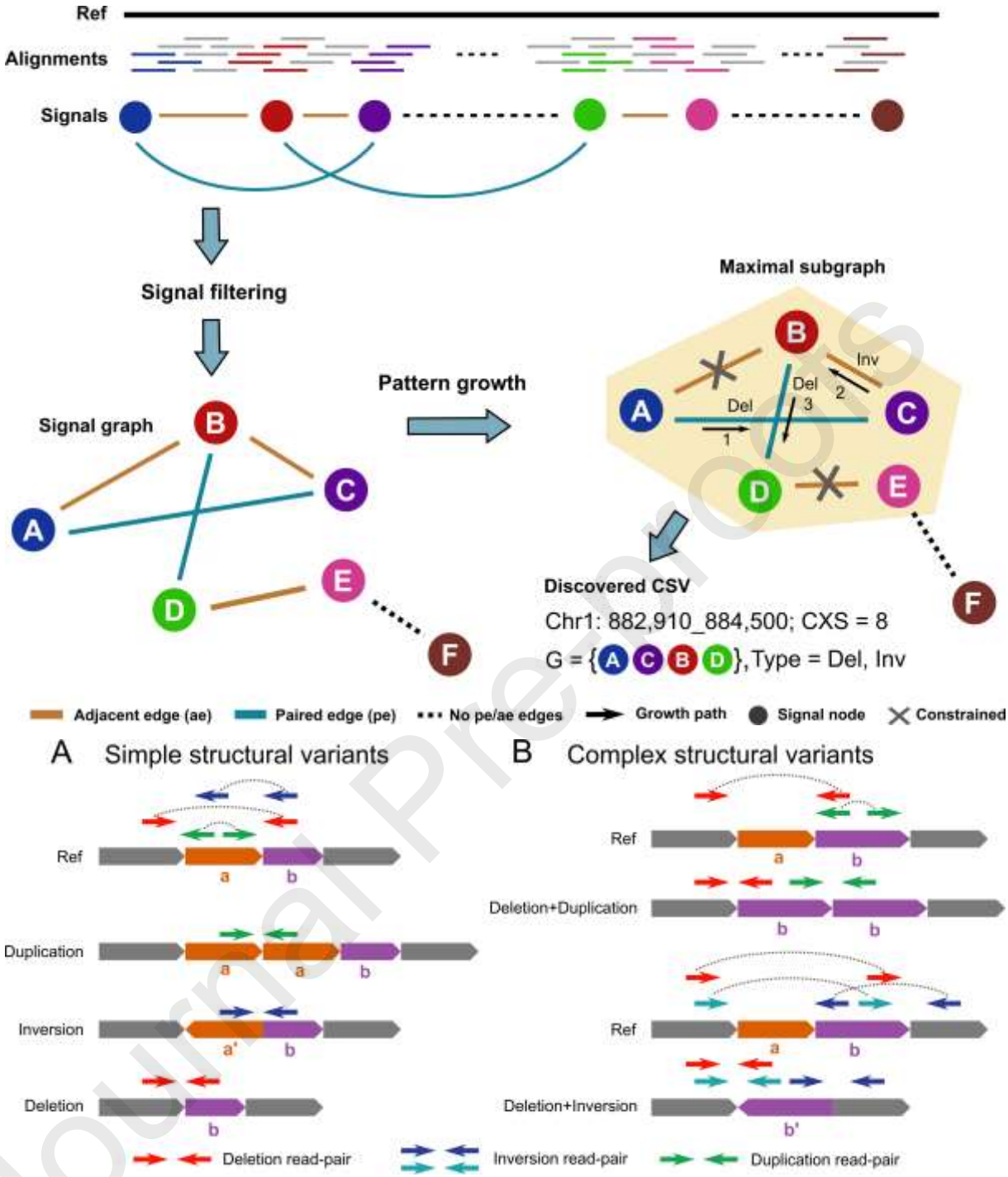
Note: ONT, Oxford Nanopore; HiFi, PacBio HiFi.











Algorithm II: Multi-location subgraph growth

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**
        $g' = g + node$;
        $O.append(g')$;
        $multiLocPatternGrowth(O, g', G|_{g'}, minFreq, minDist)$;
    end if
end for
end procedure
procedure $nodeInRange(g, v)$
    Set the nodes in g with respect increasing order of pos value: $v_0, v_1, ..., v_n$
    Set $v' = v_n$;
    if $freq(v') \geq minFreq$ **then**
        if $dist(v', v) < minDist$ **then**
            return True
        else:
            for $i = n$ **to** 0 **do**
                if $\exists\, v_{pi}$ between v and v_i **then**
                    return True
        end if
    end if
    return False
end procedure

Algorithm I: Detect maximal subgraphs

Input: Signal graph $G = (V, E)$, **parameters** $minFreq$, $minDist$
Output. A set of COV subgraphs $O = \{g_1, g_2, ..., g_m\}$, with $freq(g_i) \geq minFreq$
procedure $findMaximalSubgraph(G, minFreq, minDist)$
Initialize $freq\_types$ equals to $type$ frequency of node in V;
Build index-projection $G|_0$ of G;
for v in $freq\_types$ **do**:
    Build index-projection $G|_{g_i}$;
    $g_i = v$;
    if $freq(g_i) > minFreq$ **then**
        $multiLocPatternGrowth(O, g_i, G|_{g_i}, minFreq, minDist)$;
    end if
end for
end procedure