Algorithms for structural variant detection

Lin, J.

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

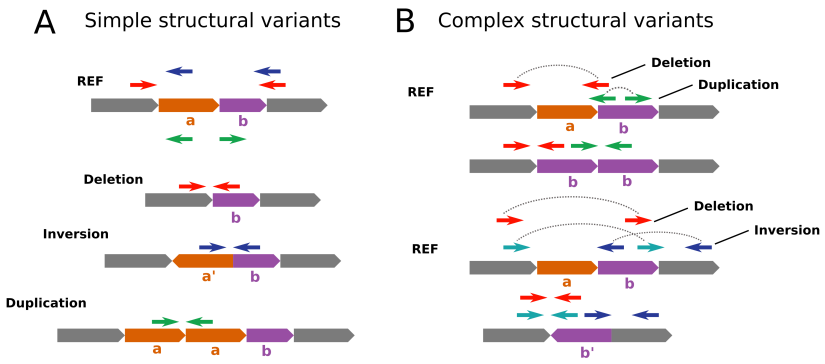

Figure 2.1: Explanation of simple and complex structure variants alignment models derived from abnormal read-pairs. (A) Three common simple SVs and their corresponding abnormal read-pair alignment on the reference genome, representing by red, blue, and green arrows. (B) The alignment signature of two 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 $v = (type, pos, weight)$,

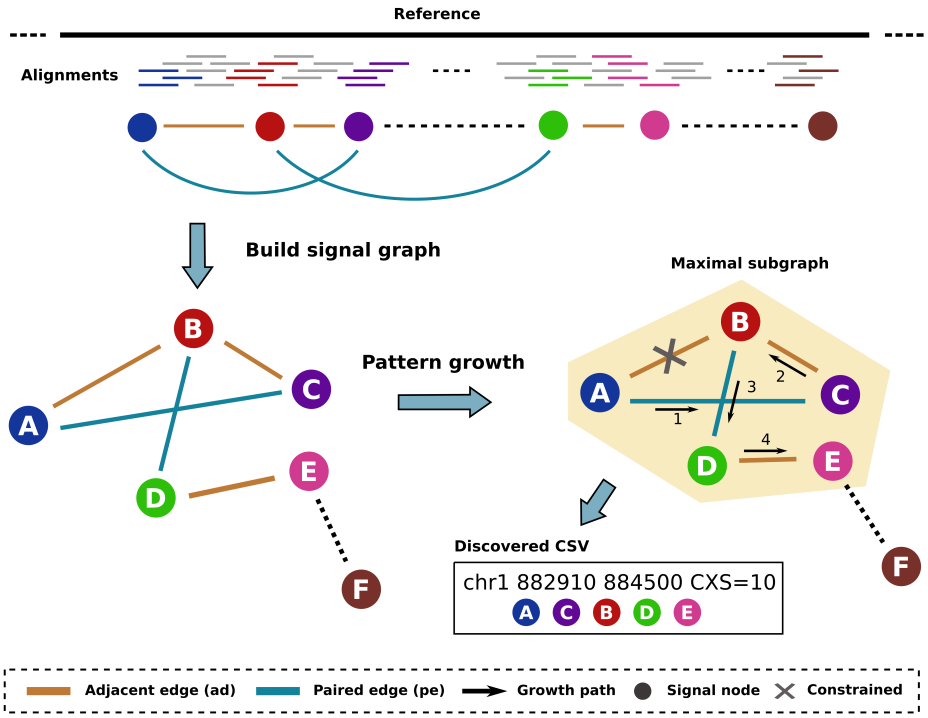

Figure 2.2: Overview of Mako. Mako first builds a signal graph by collecting abnormally aligned reads as nodes, and their edge connections are provided by paired-end alignment and split alignment. Afterward, Mako utilizes the pattern growth approach to find a maximal subgraph as a potential CSV site. 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.

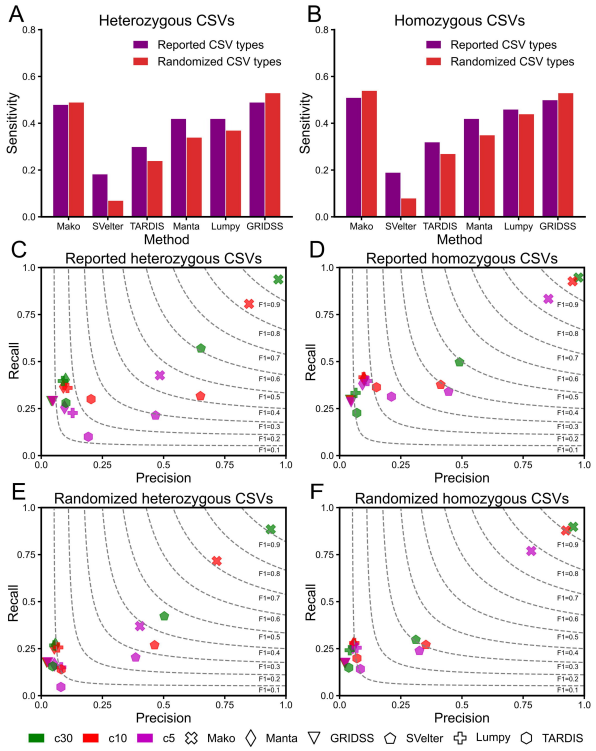

Figure 2.5: Performance comparison on simulated CSVs with different match criteria. All-breakpoint match (A and B) and unique-interval match (C–F) evaluation of selected tools for detecting simulated CSVs. (A) The sensitivity of detecting heterozygous CSVs breakpoints. (B) The sensitivity of detecting 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.

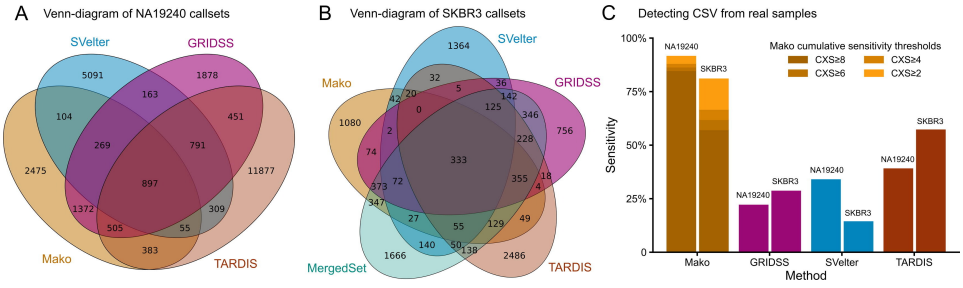

Figure 2.6: Overview of performance on NA19240 and SKBR3 for Mako, GRIDSS, SVelter and TARDIS. (A) Venn diagram of NA19240 callsets. (B) Venn diagram of SKBR3 callsets. The Venn diagrams are created by 50% reciprocal overlap via a publicly available tool Intervene with '`--bedtools-options`' enabled. 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.

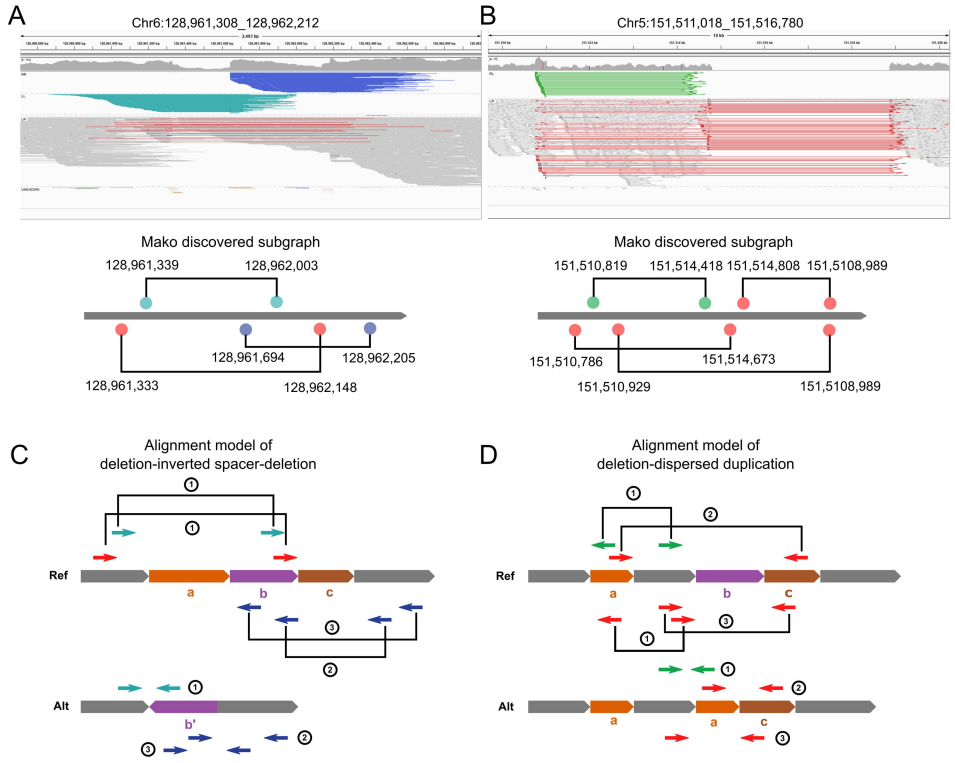

Figure 2.7: Two representative CSV subgraphs identified by Mako. The top panel of (A) and (B) are IGV views of the two events, and the alignments are grouped by read-pair orientation. The dark blue shows reverse-reverse alignments, light blue represents forward-forward alignments, green represents reverse-forward 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).

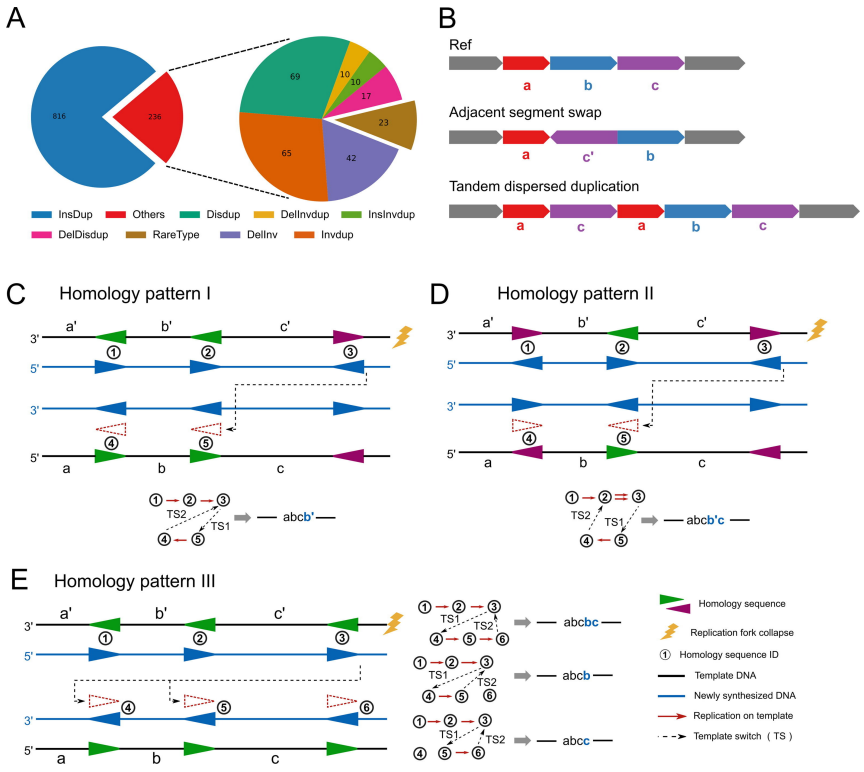

Figure 2.8: Overview of Mako's CSV discoveries from three healthy samples and proposed CSV formation mechanisms. (A) Summary of discovered CSV types, these types are reconstructed by HiFi PacBio reads, where a type with fewer than 10 events was summarized as RareType. (B) Diagrams of two novel and rare 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.