Chapter 3

SVision: A deep learning approach to resolve complex structural variants

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

3.1 Introduction

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

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

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

3.2 Material and methods

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

3.2.1 Overview of SVision

SVision begins by encoding pairs of sequences, a given read and its counterpart in reference genome, as an image showing sequence similarity and difference adapting variant detection to a multi-object recognition problem amenable to an existing deep learning framework. SVision is composed of three core components: an encoder that represents the differences and similarities between a variant supporting read and its corresponding segment in the reference genome as a denoised image, a targeted multi-object recognition (tMOR) framework that detects and characterizes CSVs via a convolutional neural network (CNN) in the denoised image, and an illustrator that creates and unifies each detected CSV as a graph representation from the denoised image (Figure 3.1A).

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

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

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

3.2.2 Three-channel coding of sequence

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

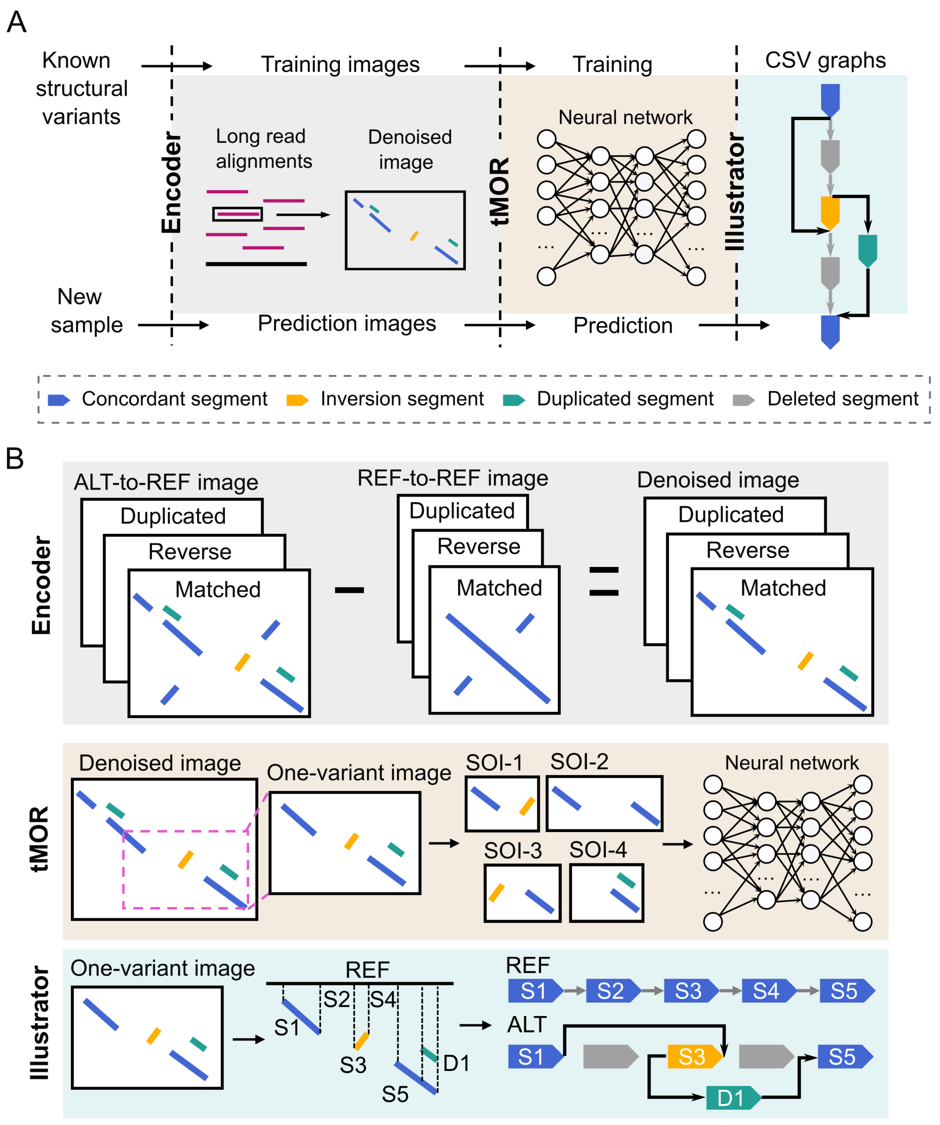


Figure 3.1: Overview of SVision. (A) Overview of the SVision workflow. (B) Details of three major modules implemented in SVision.

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

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

3.2.3 Detecting CSVs from denoised images via tMOR

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

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

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

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

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

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

3.2.4 Creating CSV graphs from denoised images

SVision uses a graph to unify the definition of different CSV types and provides a computational method to compare different CSV graph structures. To create a CSV graph $G = (V, E)$, SVision first collects the node set $V = V_S \cup V_I \cup V_D$ of G. Specifically, $V_S = \{S_1, S_2, \ldots, S_n\}$, $V_I = \{I_1, I_2, \ldots, I_m\}$ and $V_D = \{D_1, D_2, \ldots, D_k\}$, where n, m and k are the number of skeleton nodes, insertion nodes and duplication nodes in the graph, respectively. Skeleton nodes are derived from major segments in a one-variant image and sequence between discontinuous major segments on REF (i.e., concordant segments between VAR and REF). Insertion nodes consist of minor segments in the one-variant image, while insertion nodes with known origins are defined as duplication nodes, representing duplicated segments in the one-variant image. Moreover, each node $v_i \in V$ is represented as a tuple $v_i = (Seq, MathitPos, Strand)$, which represents a segment in the one-variant image. Here *Seq* indicates the segment sequence, *Pos* is the position of the segment on VAR and *Strand* represents the forward or reverse strand of the segment. The edges in G are collected by $E = E_{ad} \cup E_{dp}$. Here E_{ad} represents a set of adjacency edges $e_{ad}^{j} = (v_j, v_{j+1})$, connecting two adjacent nodes v_j and v_{j+1}, and E_{dp} represents a set of duplication edges e_{dp}, connecting the duplicated node with its known origin.

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

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

SVision keeps isomorphic graphs and symmetric graphs in two separate files, enabling search of CSV events of the same structure. For each variant call, SVision keeps all its breakpoints in the "BKPS" column in the INFO field and a type ("SVTYPE" column). Especially for CSVs, their breakpoints are kept with both coordinates and associated graph structure in the "BKPS" and "GraphID" column, respectively. Note that the "GraphID" is used to search events of a specific graph structure in isomorphic and symmetric graph output files. Moreover, SVision involves the graph breakpoints induced from the CSV Reference Graphical Fragment Assembly (rGFA) file in the "GraphBRPKS" column. Note that the "GraphID" and "GraphBRPKS" columns are only reported when the parameter '`--graph`' and '`--qname`' are activated.

3.2.5 Quality score of discoveries

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

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

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

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

  $$Consistency = \frac{Std(\{nonlinear\_score_1, \ldots, nonlinear\_score_M\})}{M}$$

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

- Prediction reliability. This part evaluates the deep learning prediction quality. The last layer in the CNN architecture is a SoftMax layer, which outputs the probability of the prediction results. Therefore, we use the average probability of all SOIs as the CNN reliability:

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

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

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

$$\mathit{qual} = \mathit{Consistency} + (1 - \mathit{Reliability})$$

and

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

where Scores $= \{\mathit{qual}_1, \ldots, \mathit{qual}_M\}$, and M is again the total number of images supporting this variant.

3.2.6 Training data and CNN model training

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

SVision adopts AlexNet, a widely-used CNN model, to recognize sequence differences in similarity images. The AlexNet architecture consists of five convolutional layers and three fully-connected layers. Specifically, the first convolution layer loads images of size $224 \times 224 \times 3$, and it uses the $11 \times 11 \times 3$ convolution kernel with stride 4. The last three layers are fully connected and contains a five-class SoftMax layer with inputs from the five preceding convolution layers. In the end, the input SOIs are detected as either INS, DEL, INV, DUP, tDUP or mixed types for CSVs. We apply the idea of transfer learning to train CNN with 75,000 SOIs. First, the parameters of all layers in the CNN are initialized to the best parameter set that was achieved on the ImageNet competition. Afterwards, we fine-tune the parameters of the last three fully-connected layers on our data using back propagation and gradient descent optimization with a learning rate of 0.001. The loss function is defined as the cross entropy between predicted probability and the true class labels. Moreover, SVision's CNN architecture is lightweight and has far fewer layers than complex CNN models such as ResNet and Inception V3, which results in a highly efficient fine-tuning process with large batch size (default: 128) even on a single CPU machine. To evaluate the trained CNN model, we apply ten-fold cross validation, and the trained model at each round is applied to an independent test set of 7,500 SOIs derived from simulated SVs. Finally, SVision selects the model with the best performance.

3.2.7 Evaluating simple structural variants detection with real data

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

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

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

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

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

3.2.8 Evaluating complex structural variant detection

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

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

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

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

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

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

and

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

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

1. Region-match between predicted CSV and benchmark deletion-inversion event.

2. For each subcomponent, we examine the breakpoint closeness and event size as well as the detected type.

3. The correct detection should pass condition 1) and 2). The subcomponent match is considered as either deletion or inversion correctly detected in 2).

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

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

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

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

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

SVision revealed two special complex structures, i.e., a structure consisting of nodes 'S:2,I:2,D:1' and path 'S1+I1+I1+I2+I2+S2+' as well as another structure consisting of nodes 'S:2,I:1,D:1' and path 'S1+I1+I1+S2+', which were visually confirmed as local targeted site duplication and tandem duplication. Events of these two structures were also filtered because they were considered as simple events from biological perspective. Afterwards, we used RepeatMasker and tandem repeat finder (TRF) annotated files from UCSC genome browser to annotate the CSVs passed the filters through BEDtools [85] intersect option. The repeat type was assigned if the CSV region overlaps with the repeat element, while the size or percentage of overlaps was not required. For CSVs with multiple repeat types, the one with the largest overlapping region with the CSV was chosen. Meanwhile, CSV was annotated as STR if the repeat unit length <7bp; otherwise, it was annotated as VNTR. Finally, we termed all CSVs outside of VNTR/STR regions as high-quality CSVs, which were further validated and used for further analysis. The PAV and short-read data matched CSV loci were obtained through BEDtools without requiring overlap size. For the short-read data, a matched CSV locus was considered as completely reconstructed if both breakpoint positions and types matched what SVision reported, otherwise as partially reconstructed events if either breakpoints or types agreed with SVision's prediction.

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

We validated 80 CSVs detected by SVision in HG00733 via 1) graph-based alignment; 2) contig-based visual confirmation; and 3) PCR and Sanger sequencing:
Graph-based alignment. For each CSV graph in rGFA format, we extracted the CSV locus spanning reads with SAMtools and aligned these reads to each CSV graph via GraphAligner (v1.0.12) with the default setting. A CSV was successfully validated if a single ONT read could be aligned to the corresponding variant path specified in the rGFA file. We then counted the number of long reads covering the entire VAR path as the number of support for this CSV event.
Contig-based visual confirmation. To examine the internal structure of CSVs, the phased-assembly specified in the PAV (v1.1.2, TIG_REGION column) at the reported variant region was used for further analysis. We first extracted the contig sequence harboring variant based on the coordinates provided in the 'PAV_TIG_REGION'. For example, a sequence containing variant was extracted from the h1 assembled genome for '1|1' and '1|0' genotype,

while from h2 assembled genome for '0|1'. In order to validate a CSV structure containing a complex insertion, we extended 5kbp both upstream and downstream the CSV region to extract the reference genome via BEDtools getfasta option, from which the origin of the inserted sequence could be identified. Afterwards, Gepard was used to create the Dotplot of contig sequence (vertical axis in the Dotplot) and reference sequence (horizontal axis in the Dotplot) for each CSV locus. Based on each contig Dotplot, the manual validation contained two tiers of metrics: 1) whether the reported region contains a variant; and 2) whether the SVision reported structure is identical to what was revealed by Dotplot. A CSV was considered completely reconstructed if both 1) and 2) were satisfied, while others were considered as inconclusive events.
PCR and Sanger sequencing. We first determined that about half of the 80 CSVs (39/80) were intractable for PCR due to their location within segmental duplications, the size of the amplicon needed to validate the rearrangement, or the simple repeat nature of the rearrangement. We then randomly selected 20 of the remaining rearrangements, and performed BLAT on the local region from the HG0733 assembly data. We next attempted to PCR each of the 20 CSVs. Briefly, we designed primers flanking the CSV or flanking breakpoints within the CSV for each of the 20 events. Next, we attempted to amplify each region using Takara LA taq. We obtained the predicted band size for 12 of the 20 variant loci; the remaining 8 regions did not amplify in 3 separate attempts with alterations of the PCR conditions and template amounts. All PCR products were sent to Sanger sequencing and validated as on target, and contained the correct amplicon with the breakpoint from the assembly and SVision call.

3.2.10 Data availability

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

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

3.3 Results

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

3.3.1 Evaluating simple SV detection with real data

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

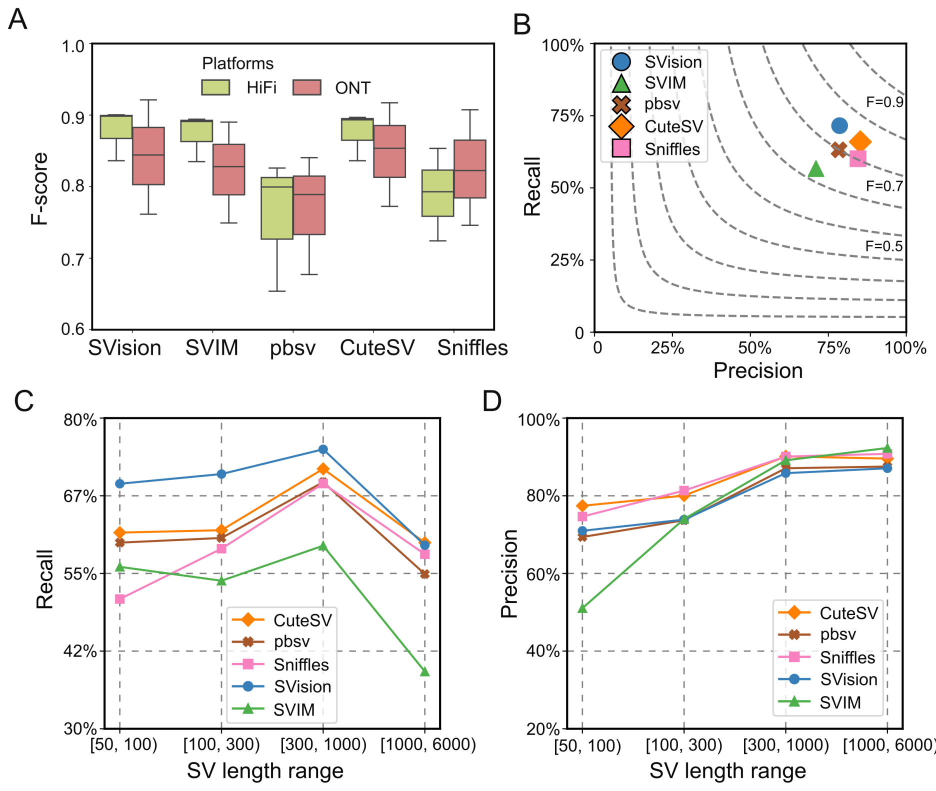


Figure 3.2: Performance of detecting simple structural variants from real data. (A) F-score of detecting variants in HG002 evaluated with Truvari. (B) Recall and precision of detecting NA12878 Phased Assembly Variant (PAV) calls. (C) Recall of detecting NA12878 PAV calls at different size range. (D)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

3.3.2 Performance of detecting complex structural variants

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

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

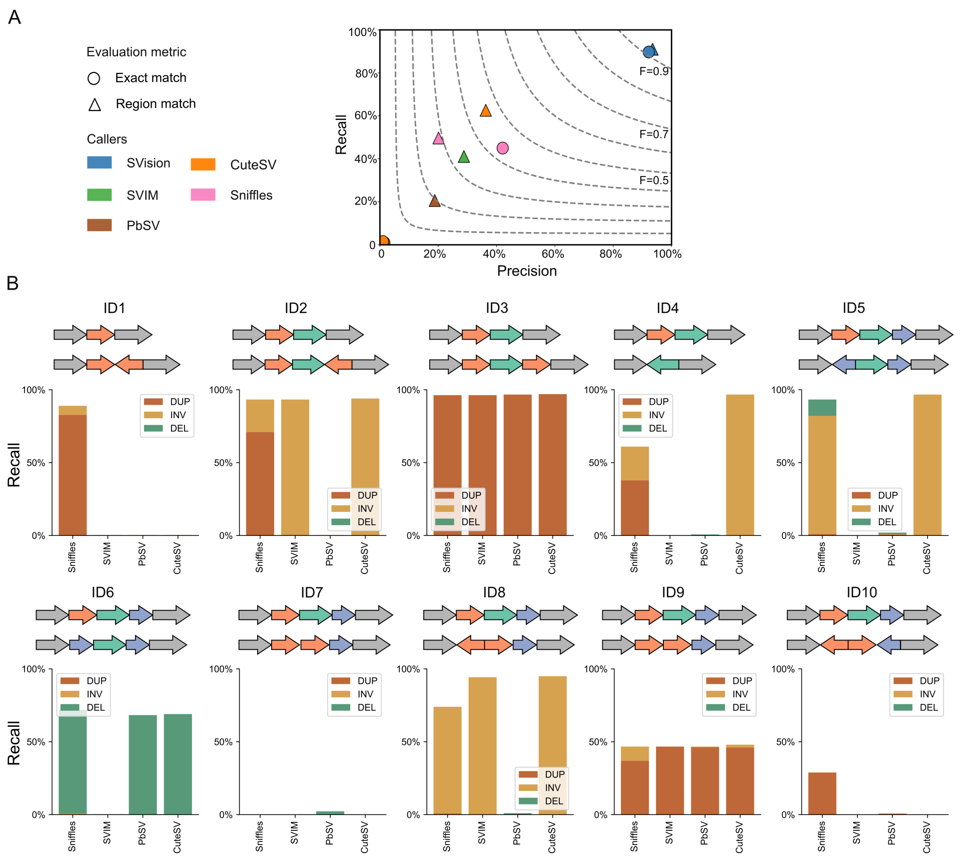


Figure 3.3: Performance of detecting simulated complex structural variants. (A) Performance of detecting simulated complex structural variants (CSVs), which was evaluated with recall (vertical axis), precision (horizontal axis) and F-score (F, dashed line). (B) The recall of model-based callers for detecting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

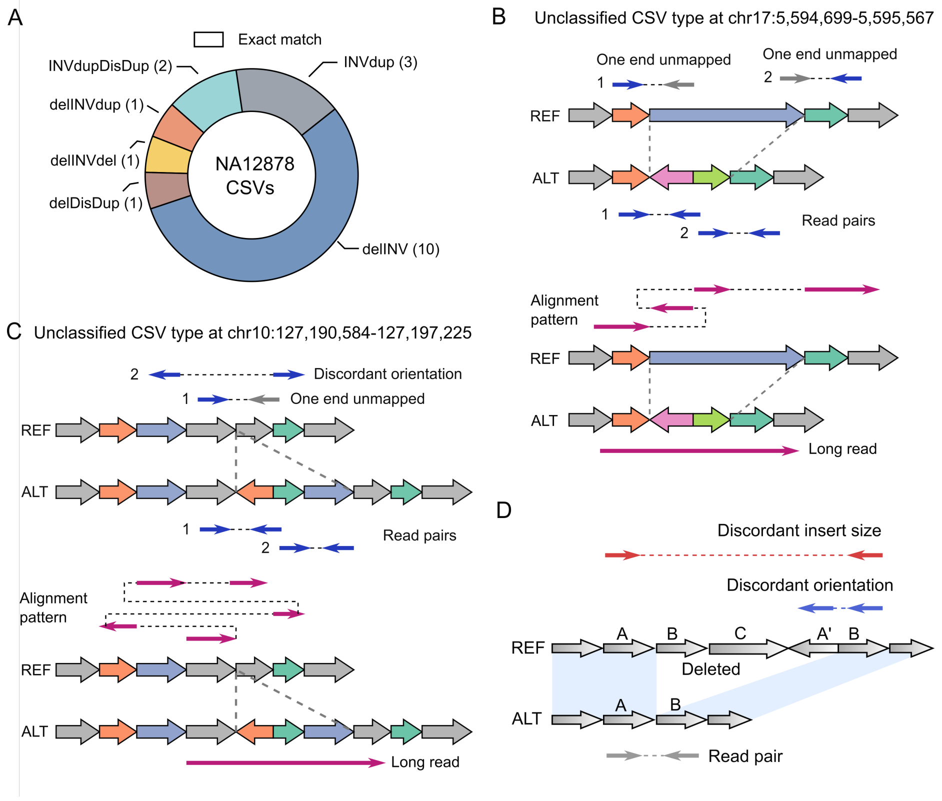


Figure 3.4: Performance of detecting complex structural variants in NA12878. (A) Performance of SVision detecting CSVs from NA12878 evaluated by exact match, where SVision detected all complex events. (B) A deleted sequence replaced with dispersed duplication and inverted duplication, which is correctly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

3.3.3 CSV mediated gene structure change and genome evolution

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

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

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

Additionally, SVision identified an insertion-inversion-insertion event (chr9:74,283,222-74,283,473), which was detected as a 1,737bp insertion by PAV but completely missed in previous long-read call sets [9, 87] (Figure 3.6A). This event was also re-genotyped by PanGenie, and it found 80% allele frequency among 3,202 1KGP cohort [10]. The inserted sequence of this CSV was also identified in primate genomes (Figure 3.6B), such as gorilla, indicating the inserted state was ancestral and the reference was derived via deletion and inversion.

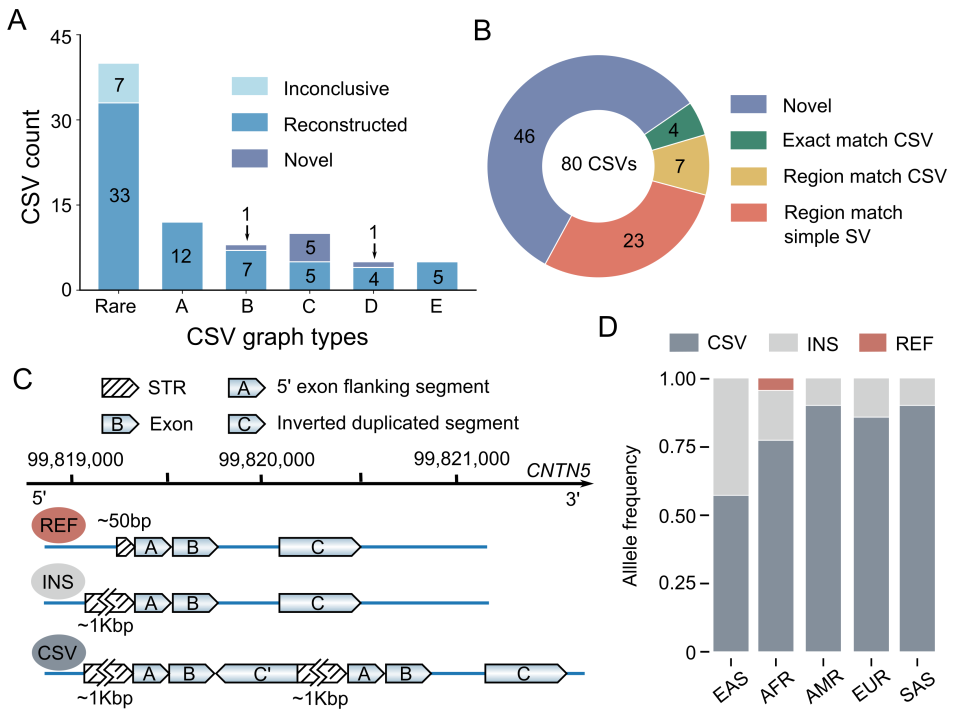
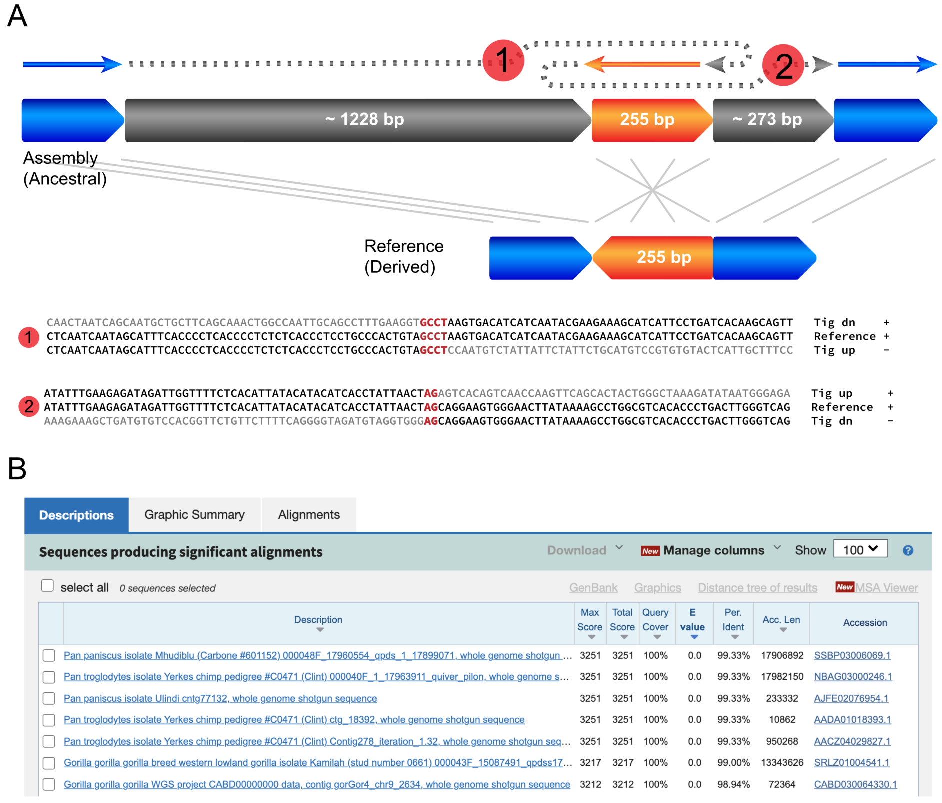


Figure 3.5: Application of SVision on HG00733 HiFi data. (A) SVision detected complex structural variants (CSVs) overlapped with Phased Assembly Variant (PAV) calls and reconstructed with HiFi haplotype contigs. The rare type represented a graph type containing less than five complex events. Graph type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



```
   CAACTAATCAGCAATGCTGCTTCAGCAAACTGGCCAATTGCAGCCTTTGAAGGTGCCTAAGTGACATCATCAATACGAAGAAAGCATCATTCCTGATCACAAGCAGTT   Tig dn    +
1  CTCAATCAATAGCATTTCACCCCTCACCCCTCTCTCACCCTCCTGCCCACTGTAGCCTAAGTGACATCATCAATACGAAGAAAGCATCATTCCTGATCACAAGCAGTT   Reference +
   CTCAATCAATAGCATTTCACCCCTCACCCCTCTCTCACCCTCCTGCCCACTGTAGCCTCCAATGTCTATTATTCTATTCTGCATGTCCGTGTGTACTCATTGCTTTCC   Tig up    -

   ATATTTGAAGAGATAGATTGGTTTTCTCACATTATACATACATCACCTATTAACTAGAGTCACAGTCAACCAAGTTCAGCACTACTGGGCTAAAGATATAATGGGAGA   Tig up    +
2  ATATTTGAAGAGATAGATTGGTTTTCTCACATTATACATACATCACCTATTAACTAGCAGGAAGTGGGAACTTATAAAAGCCTGGCGTCACACCCTGACTTGGGTCAG   Reference +
   AAAGAAAGCTGATGTGTCCACGGTTCTGTTCTTTTCAGGGGTAGATGTAGGTGGGAGCAGGAAGTGGGAACTTATAAAAGCCTGGCGTCACACCCTGACTTGGGTCAG   Tig dn    -
```

Descriptions | Graphic Summary | Alignments

Sequences producing significant alignments Download New Manage columns Show 100

select all *0 sequences selected* GenBank Graphics Distance tree of results New MSA Viewer

	Description	Max Score	Total Score	Query Cover	E value	Per. Ident	Acc. Len	Accession
☐	Pan paniscus isolate Mhudiblu (Carbone #601152) 000048F_17960554_qpds_1_17899071, whole genome shotgun …	3251	3251	100%	0.0	99.33%	17906892	SSBP03006069.1
☐	Pan troglodytes isolate Yerkes chimp pedigree #C0471 (Clint) 000040F_1_17963911_quiver_pilon, whole genome s…	3251	3251	100%	0.0	99.33%	17982150	NBAG03000246.1
☐	Pan paniscus isolate Ulindi cntg77132, whole genome shotgun sequence	3251	3251	100%	0.0	99.33%	233332	AJFE02076954.1
☐	Pan troglodytes isolate Yerkes chimp pedigree #C0471 (Clint) ctg_18392, whole genome shotgun sequence	3251	3251	100%	0.0	99.33%	10862	AADA01018393.1
☐	Pan troglodytes isolate Yerkes chimp pedigree #C0471 (Clint) Contig278_iteration_1.32, whole genome shotgun seq…	3251	3251	100%	0.0	99.33%	950268	AACZ04029827.1
☐	Gorilla gorilla gorilla breed western lowland gorilla isolate Kamilah (stud number 0661) 000043F_15087491_qpdss17…	3217	3217	100%	0.0	99.00%	13343626	SRLZ01004541.1
☐	Gorilla gorilla gorilla WGS project CABD00000000 data, contig gorGor4_chr9_2634, whole genome shotgun sequence	3212	3212	100%	0.0	98.94%	72364	CABD030064330.1

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

3.4 Conclusion

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