Algorithms for structural variant detection

Lin, J.

Chapter 4

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

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

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

4.1 Introduction

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

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

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

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

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

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

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

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

4.2 Material and methods

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

4.2.1 Overview of SpotSV

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

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

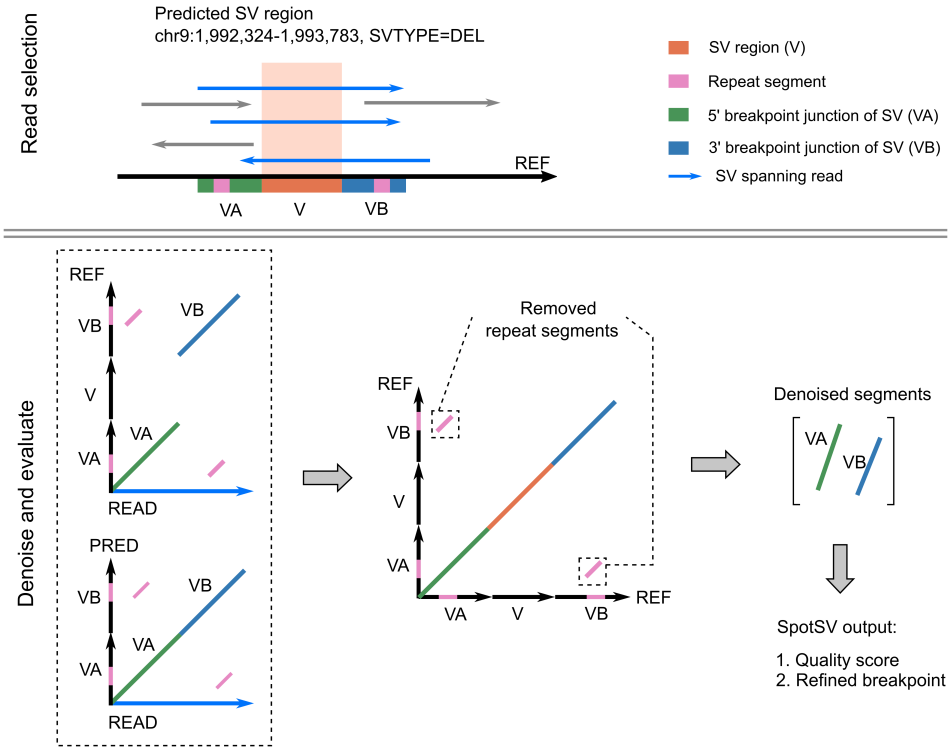

Figure 4.1: Overview of SpotSV. SpotSV consists of two major modules: 1) Read selection and 2) Denoise and evaluate. Module 1) is designed to select variant overlapping reads, containing reads across entire events and those only covering the breakpoint junctions. Module 2) consists of two steps. Firstly,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

4.2.2 Modify reference sequence with predicted structural variants

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

4.2.3 Generating denoised segments based on k-mers

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

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

4.2.4 Calculating structural variant validation score

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

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

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

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

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

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

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

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

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

4.2.5 Data availability

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

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

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

4.3 Results

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

4.3.1 Evaluating SpotSV with simulated data

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

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

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

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

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

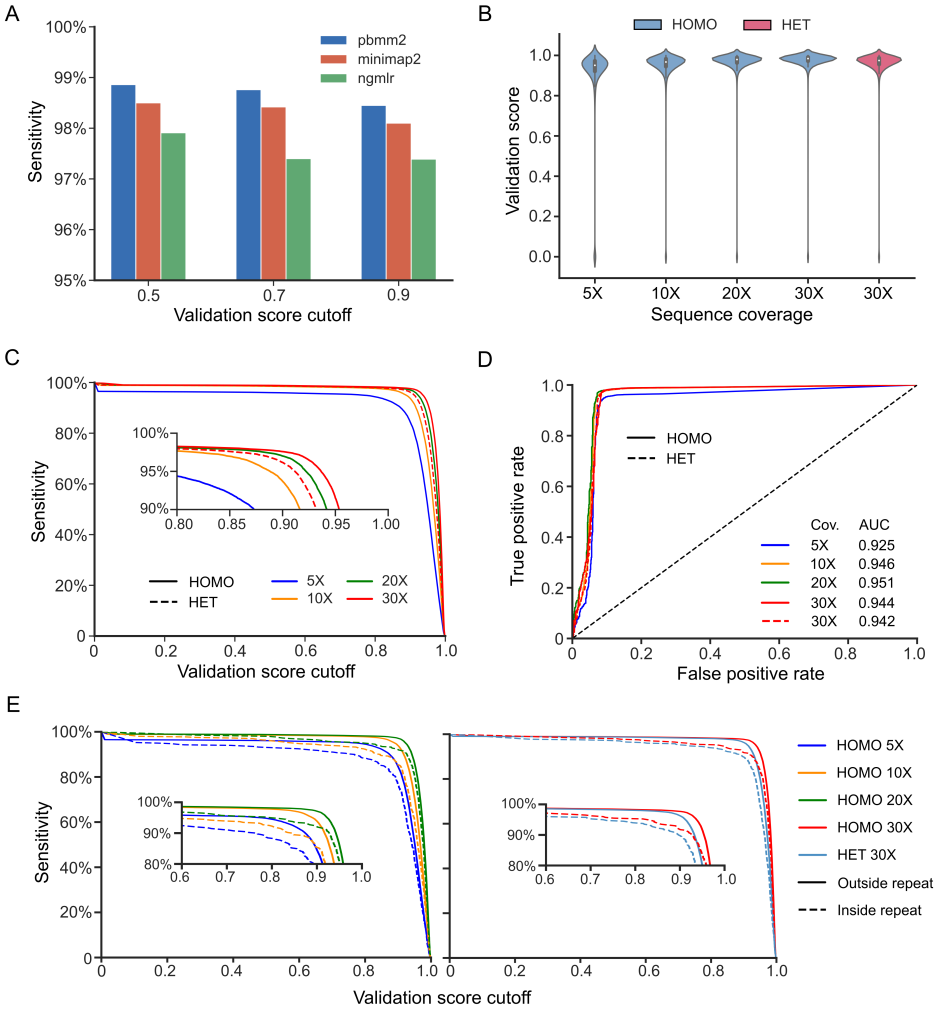

Figure 4.2: Performance of validating simulated structural variants across different coverages. (A) Sensitivity of validating simulated structural variants (SVs) at different validation score cutoffs using long reads mapped with different aligners. (B) The distribution of validation score of homozygous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

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

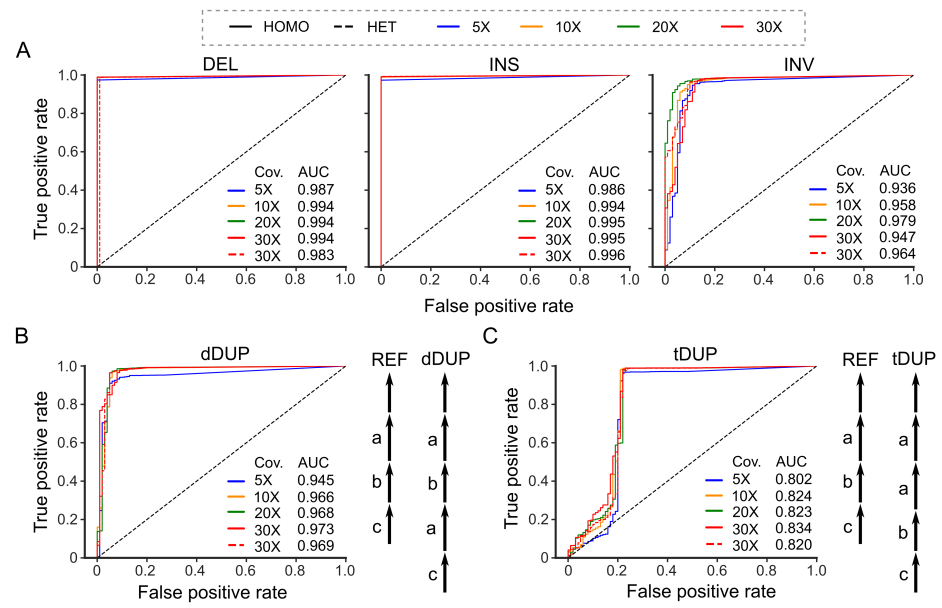

Figure 4.3: Receiver operating characteristic curve of validating five simple structural variant types. (A) The true positive rate and false positive rate of validating deletion (DEL), insertion (INS) and inversion (INV) across different coverages. (B) The true positive rate and false positive rate of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

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

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

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

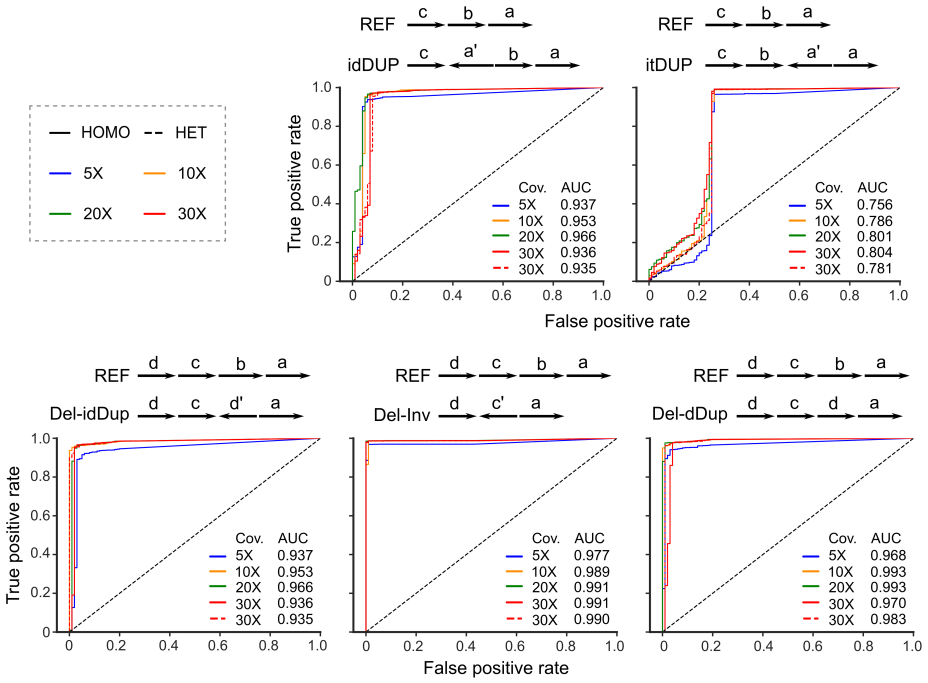

Figure 4.4: Receiver operating characteristic curve of validating five complex structural variant types. idDUP: inverted dispersed duplication, itDUP: inverted tandem duplication, Del-idDup: deletion associated with inverted dispersed duplication, Del-Inv: deletion associated with inversion, Del-dDup: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

4.3.2 Validating structural variants in a well-characterized genome

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

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

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

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

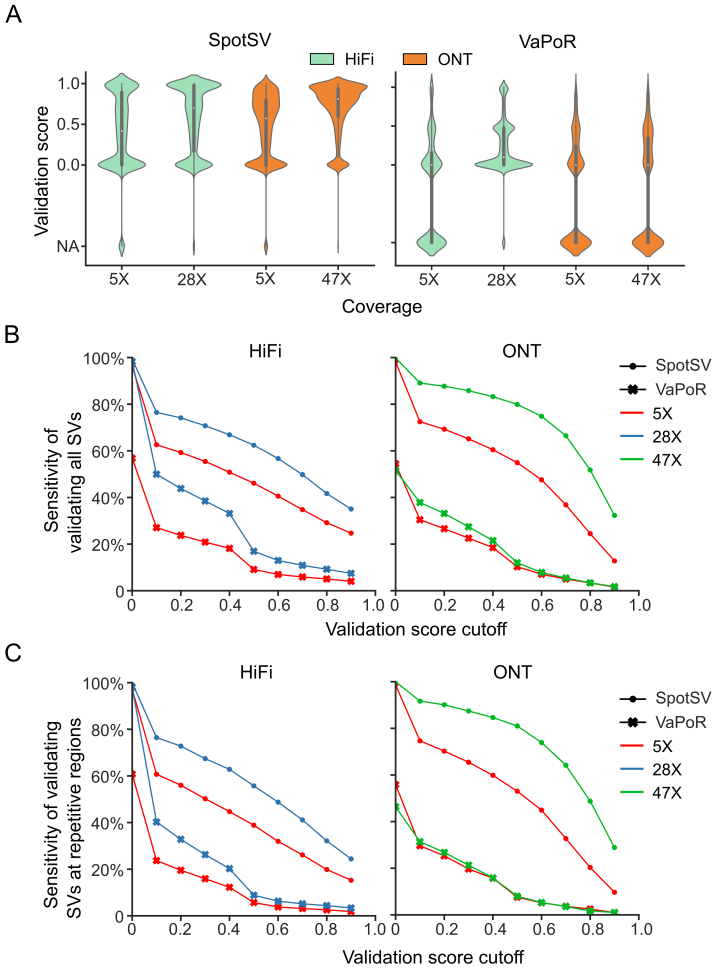

Figure 4.5: Performance of validating structural variants in HG002. (A) The distribution of validation score assessed by SpotSV and VaPoR for all structural variants (SVs) in the HG002 benchmark. NA indicates SVs that coule not be assessed. (B) The sensitivity of validating all SVs in HG002 using different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

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

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

4.3.3 Structural variant breakpoint validation and accuracy

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

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

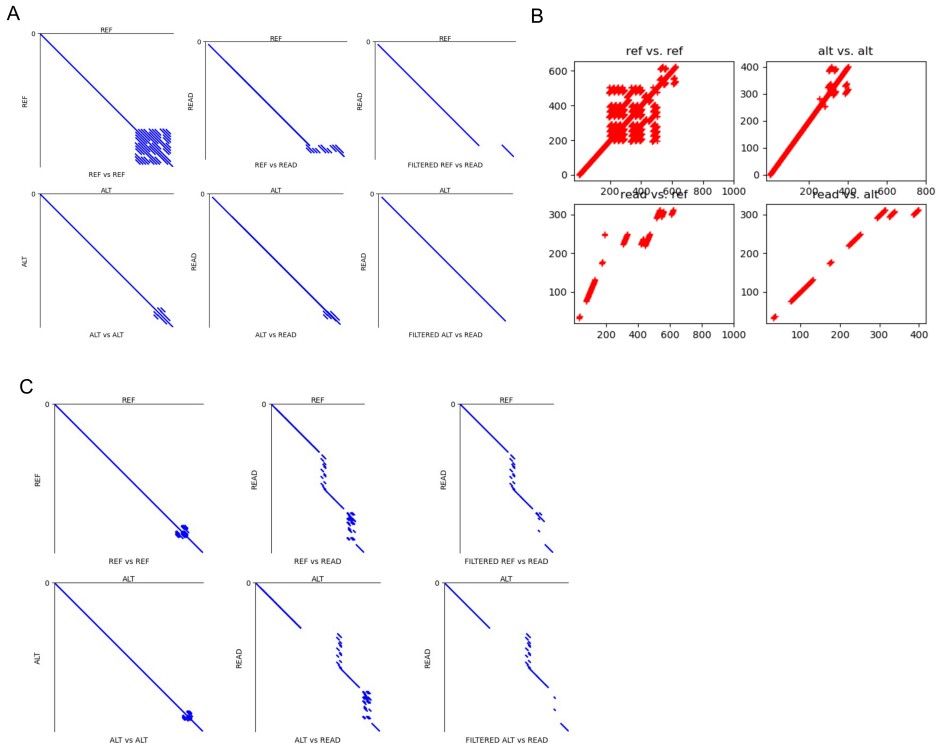

Figure 4.6: Examples of SpotSV validated SVs. (A) SpotSV validates a deletion at a tandem repeat region of validation score 1.0, while VaPoR (B) calculates a validation score of 0.3 for this event. (C) SpotSV identifies a variant locus containing two insertions, but this event was labeled as 'NA' by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

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

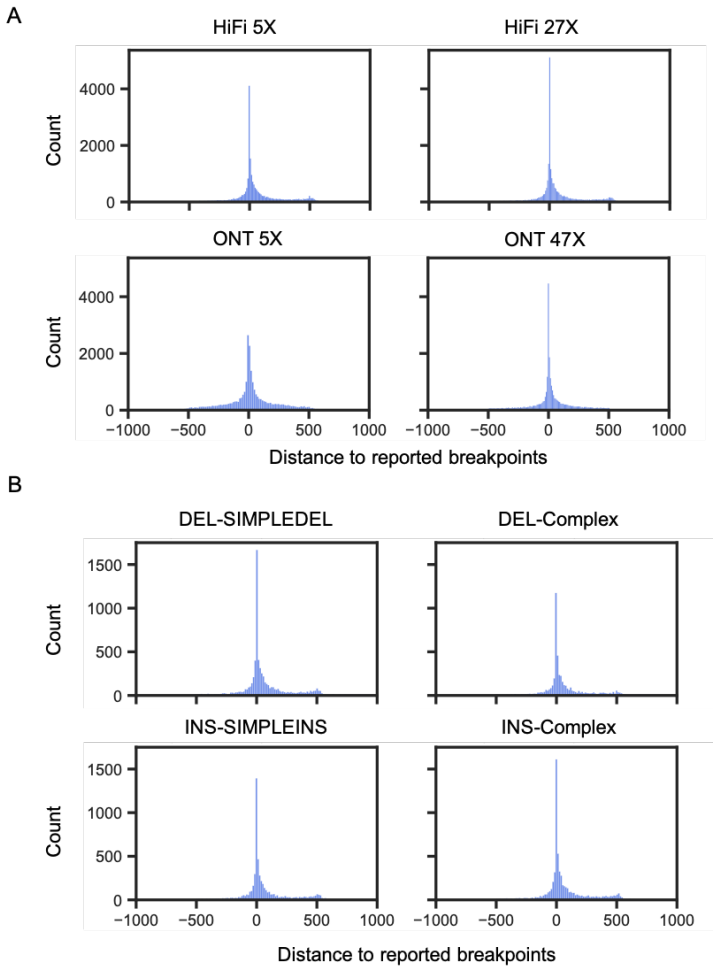

Figure 4.7: Breakpoint accuracy of SpotSV on HG002 calls. (A) Overall breakpoint offsets evaluated on HiFi and ONT data. (B) The distance of benchmark breakpoints to breakpoint identified by SpotSV from 27X HiFi data. The breakpoint comparison is grouped by the complexity of variant loci. Specifically,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

4.4 Conclusion

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

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