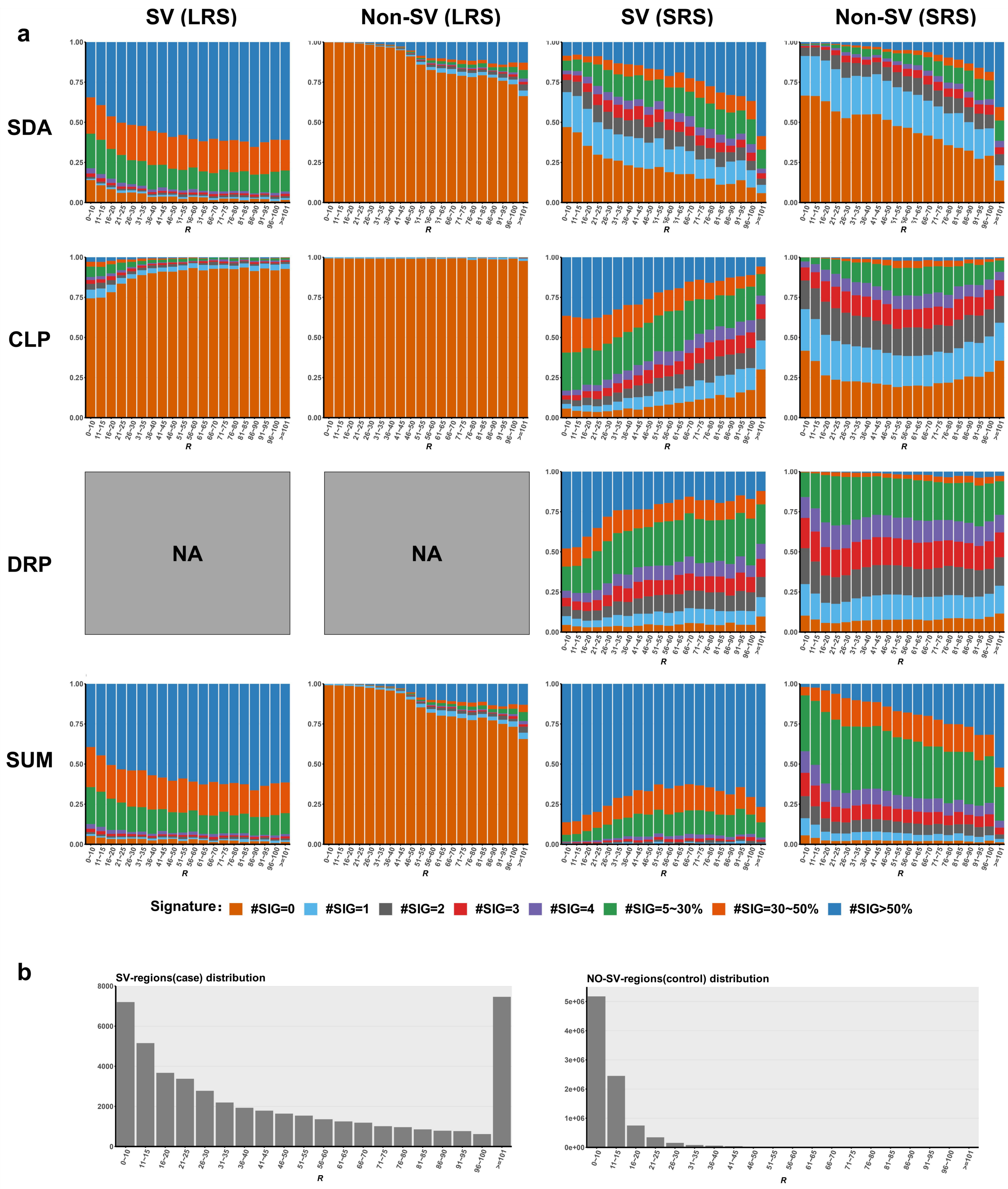
HitSV: Maximizing discovery of structural variants across sequencing technologies

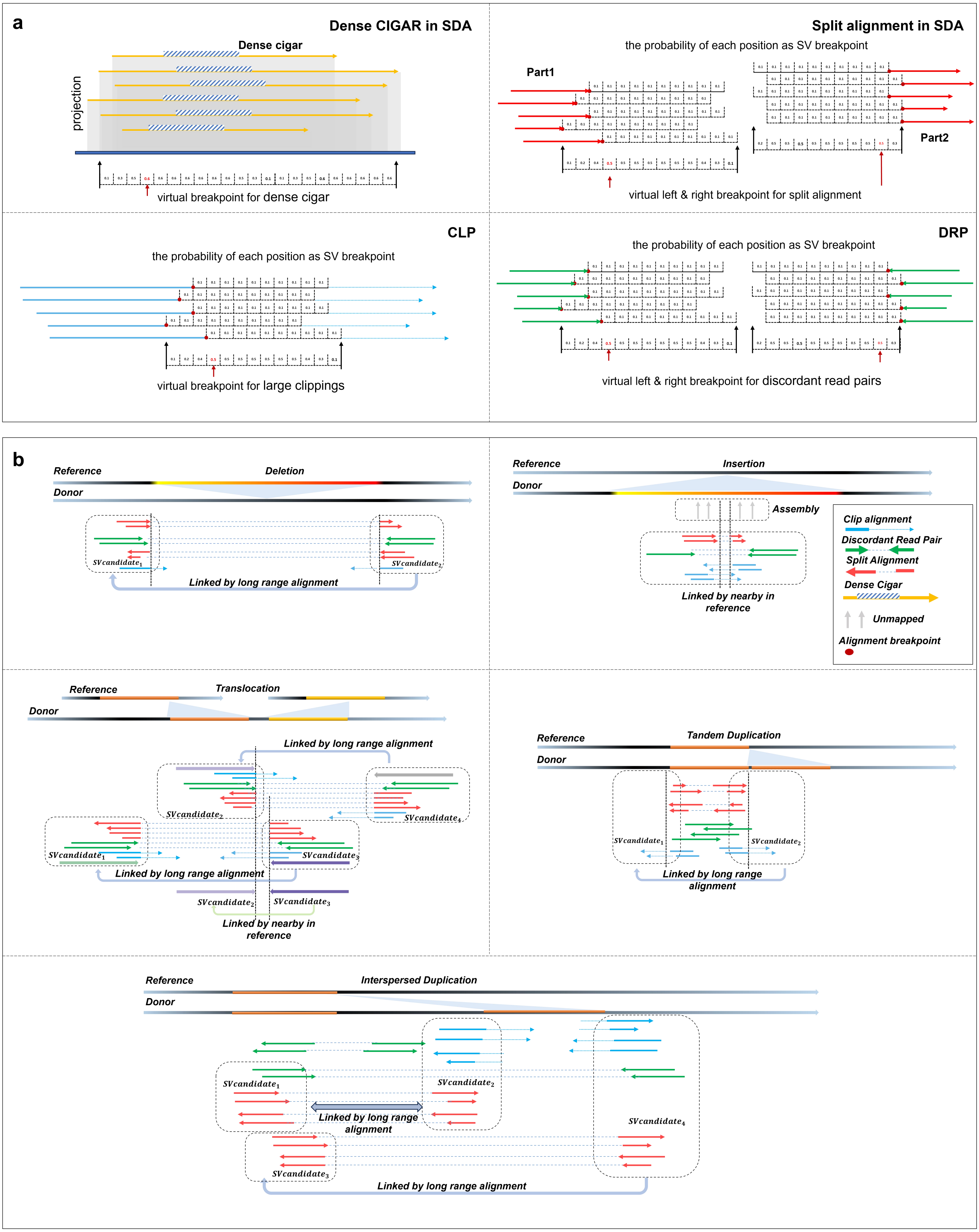
Supplementary Material



Supplementary Figure 1. The distributions of SV signatures in various repetitiveness context

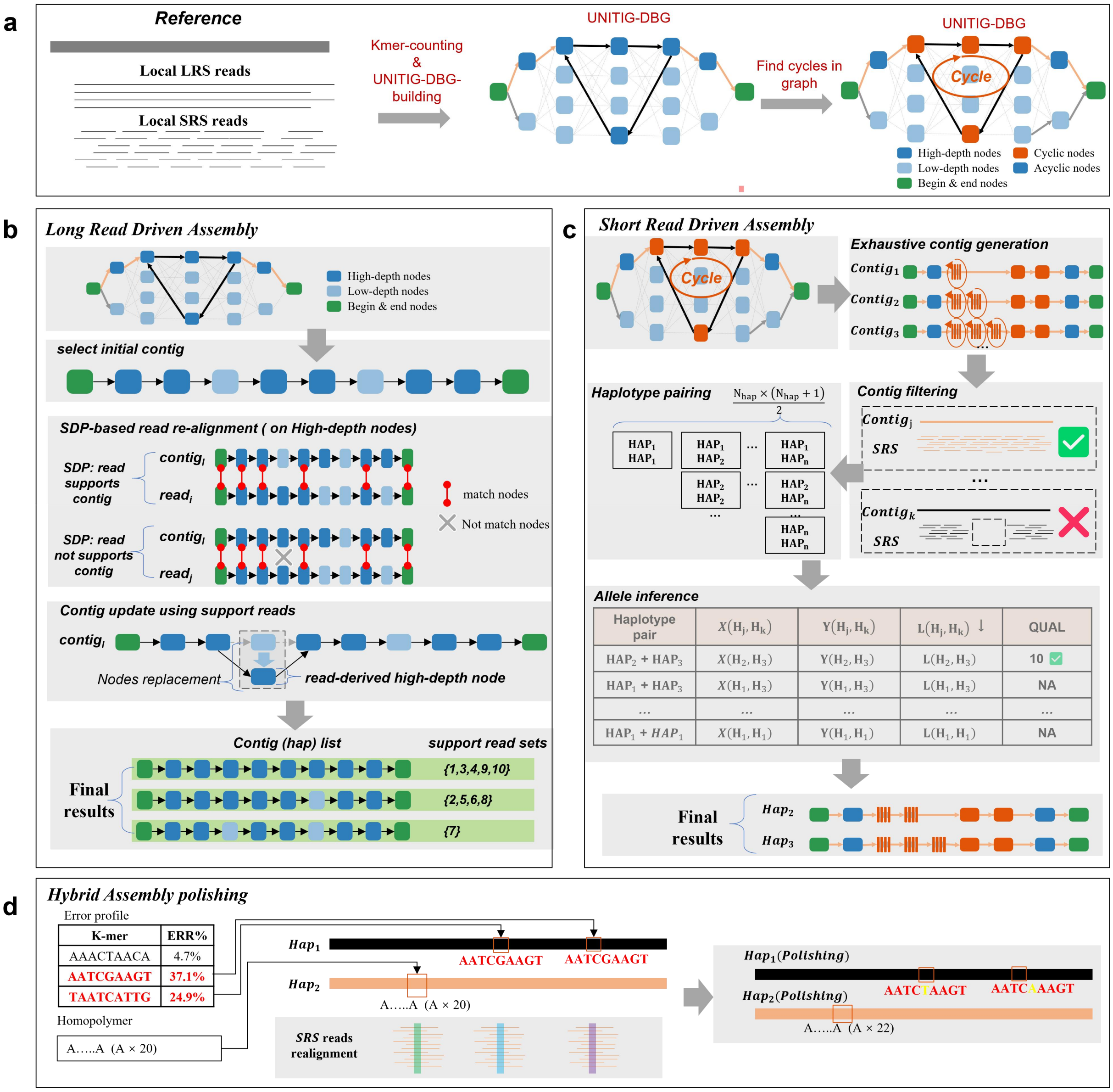
a, The distributions of various kinds of signatures exhibited by SRS and LRS (40 samples shared from HPRC and 1000 genome Project). The four rows indicate the distributions of SDA, CLP, DRP (only available for SRS) signatures and their sums, respectively, and the four columns indicate the LRS and SRS signatures in SV- and non-SV windows, respectively.

Each of the distribution is shown as a histogram (in the form of stacked bar plots). Each bar is corresponding to a specific range of local repetitiveness (R, 10 degrees per bar), and the colored blocks indicate the proportions of the windows in the context of R that have specific numbers of signature reads (i.e., #SIG in figure, indicating the windows having 0, 1, 2, 3, 4, 5 reads in absolute terms and 5-30%, 30-50%, >50% of total number of reads, respectively). **b,** The histogram of the SV- and non-SV windows of human reference genome corresponding various R ranges (shown in absolute terms of window numbers).


a
Dense CIGAR in SDA
Dense cigar
projection
virtual breakpoint for dense cigar
Split alignment in SDA
the probability of each position as SV breakpoint
Part1
Part2
virtual left & right breakpoint for split alignment
CLP
the probability of each position as SV breakpoint
virtual breakpoint for large clippings
DRP
the probability of each position as SV breakpoint
virtual left & right breakpoint for discordant read pairs
b
Reference
Deletion
Donor
$SVcandidate_1$
$SVcandidate_2$
Linked by long range alignment
Reference
Insertion
Donor
Assembly
Linked by nearby in reference
Clip alignment
Discordant Read Pair
Split Alignment
Dense Cigar
Unmapped
Alignment breakpoint
Reference
Translocation
Donor
Linked by long range alignment
$SVcandidate_2$
$SVcandidate_4$
$SVcandidate_1$
$SVcandidate_3$
Linked by long range alignment
$SVcandidate_2$
$SVcandidate_3$
Linked by nearby in reference
Reference
Tandem Duplication
Donor
$SVcandidate_1$
$SVcandidate_2$
Linked by long range alignment
Reference
Interspersed Duplication
Donor
$SVcandidate_1$
$SVcandidate_2$
Linked by long range alignment
$SVcandidate_4$
$SVcandidate_3$
Linked by long range alignment

Supplementary Figure 2. A schematic illustration on refined two-phase SV signature clustering

a, Phase1: virtual breakpoint-based signature clustering and refinement. HitSV estimates the virtual breakpoint implied by SDA, CLIP, DRP signature clusters in three different approaches. For SDA signature in dense cigar form (upper left panel, the dashed blocks indicate the dense cigars), HitSV projects all the bases of the reads of signature to the reference and define the distribution of SV breakpoint on the positions having at least one base being mapped to. For each of the position, a probability is assigned as the number of bases being mapped (normalized by the total number of bases), further, HitSV separately tests each of the signatures where the p-value is determined as the probability at the centroid of the signature in reference. For SDA signature in split alignment form (upper right panel), HitSV separately handles the multiple parts of the alignments. For each part, HitSV extends an L bp (default value: 10 bp) block from the split point, projects the block to reference and assigns a weight (default value: 0.1) to each of the projected positions. Further, the null distribution is defined on all the reference positions having at least one projected base, and for each position the probability is set as the sum weights of all the bases being projected there. Further, HitSV tests each of the signatures using this distribution. For CLP signature (lower left panel), HitSV extends an L bp (default value: 10 bp) block from the clipping point and implements base projection and weighting in a similar way to that of split alignment. Further, the null distribution is also defined on all the reference positions being projected for signature testing. For DRP signature (lower right panel), HitSV also handles the involved read pairs similarly. The difference is that the extended blocks are generated from the downstream and upstream alignment endpoints of the two paired reads, respectively. With the extended blocks, the null distribution is also composed based on the sum weights of the read projection positions. **b,** Phase2: the connection of refined clusters. HitSV greedily connects the signature clusters in nearby regions and/or being linked to combine the signature clusters belonging to the same SV event but having various types of signatures or being interspersed caused by multiple breakpoints and/or genomic repeats. For a deletion event (upper left panel), HitSV directly integrates the nearby clusters of SDA, CLP and DRP. Moreover, the DRP clusters can be further linked by read pairing information. For an insertion event (upper right panel), nearby CLP, SDA and DRP clusters are directly integrated similarly. Since insertion events can shorten the junctions of paired reads, in most cases the clusters can be comprehensively collected in a local region without regarding to the distant linking information. For a translocation event (middle left panel), it typically has two breakpoints and four involving clusters, nearby clusters were linked initially and then the long-range clusters are then clustered. For a tandem duplication event (middle right panel), CLP, SDA and DRP produce two breakpoints located around the boundaries of the duplication, separately. At each boundary, the nearby breakpoints from each category are first combined, and then the long-range clusters are straightforwardly merged. For an interspersed duplication event (lower panel), CLP and SDA produce four involving clusters, the split alignment signature support long-range information to combine all the clusters. Also refer to Fig. 1b for the illustration for inversion events.



Allele inference

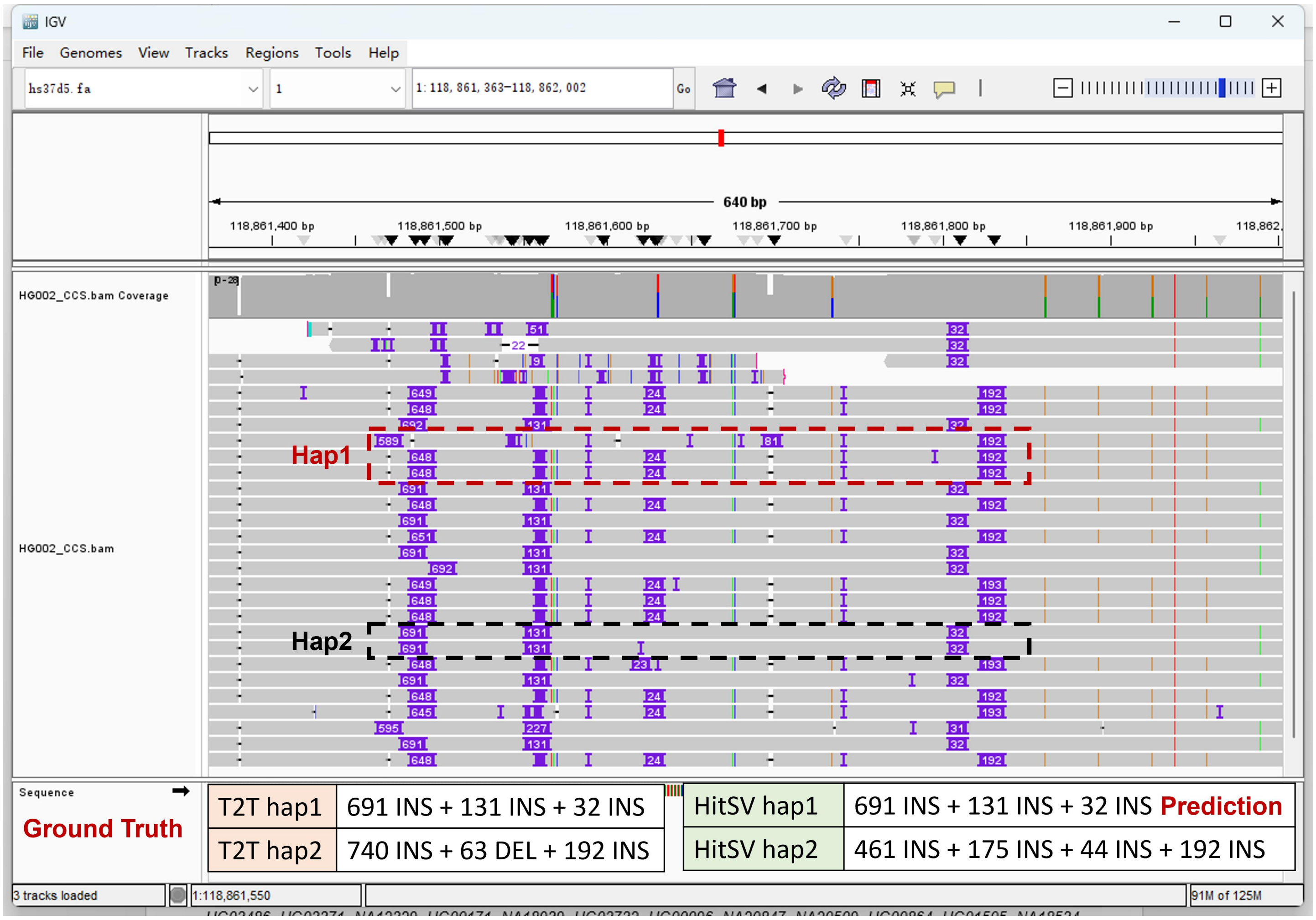
Haplotype pair	$X(H_j, H_k)$	$Y(H_j, H_k)$	$L(H_j, H_k)$ ↓	QUAL
$HAP_2 + HAP_3$	$X(H_2, H_3)$	$Y(H_2, H_3)$	$L(H_2, H_3)$	10 ✓
$HAP_1 + HAP_3$	$X(H_1, H_3)$	$Y(H_1, H_3)$	$L(H_1, H_3)$	NA
…	…	…	…	…
$HAP_1 + HAP_1$	$X(H_1, H_1)$	$Y(H_1, H_1)$	$L(H_1, H_1)$	NA

Error profile

K-mer	ERR%
AAACTAACA	4.7%
AATCGAAGT	37.1%
TAATCATTG	24.9%

Supplementary Figure 3. A schematic illustration of local haplotype sequence reconstruction and polishing

HitSV performs a unitig-DBG–based local haplotype-resolved assembly to reconstruct SV alleles through the following steps. **a,** Read collection and graph construction. All reads including those without explicit SV signatures in the vicinity of candidate SV events are retrieved for assembly. From these reads and the local reference sequence, HitSV builds a unitig de Bruijn graph (unitig-DBG). Then unitigs are annotated by their average coverage, classified as high- or low-depth using an adaptive threshold, and labeled as cyclic or acyclic after cycle detection. **b,** Long-read–based SV detection. Each long read is represented as a path through the graph. HitSV initializes contig from the path spanning the greatest total length of high-depth nodes, and then iteratively aligns other reads to existing paths using dynamic programming on high-depth nodes. A read is considered not to support a contig if high-depth nodes at any corresponding positions diverge in sequence, or if a large gap is observed at any position; otherwise, the read is regarded as supporting the contig. Reads that support a contig refine the corresponding path, whereas reads that do not support any existing contig initiate new ones. **c,** Short-read–based SV calling. To handle cycles that confound conventional assemblers, HitSV enumerates candidate alleles by controlled graph traversal between flanking anchors, limiting node revisits to a fixed number (at most nine times by default) to prevent infinite walks. Generated contigs are filtered by realigning reads and discarding those lacking sufficient support. Surviving contigs are treated as candidate haplotypes; all diploid haplotype pairs are evaluated using a likelihood that penalizes unmapped reads and inconsistencies between read and haplotype k-mer counts. The best pair is realigned to the reference to infer SV calls. **d,** Hybrid calling and polishing. When both data types are available, HitSV prioritizes long-read contigs as backbones, and polishes them with short reads. An empirical 9-mer error profile derived from high-confidence alignments guides identification of low-fidelity regions and homopolymers, which are locally reassembled from short reads and corrected to improve base-level accuracy.

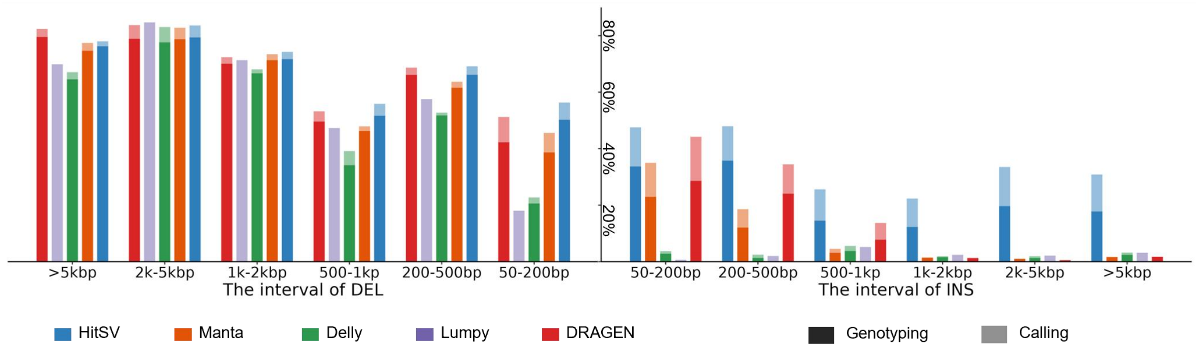


T2T hap1	691 INS + 131 INS + 32 INS
T2T hap2	740 INS + 63 DEL + 192 INS

HitSV hap1	691 INS + 131 INS + 32 INS
HitSV hap2	461 INS + 175 INS + 44 INS + 192 INS

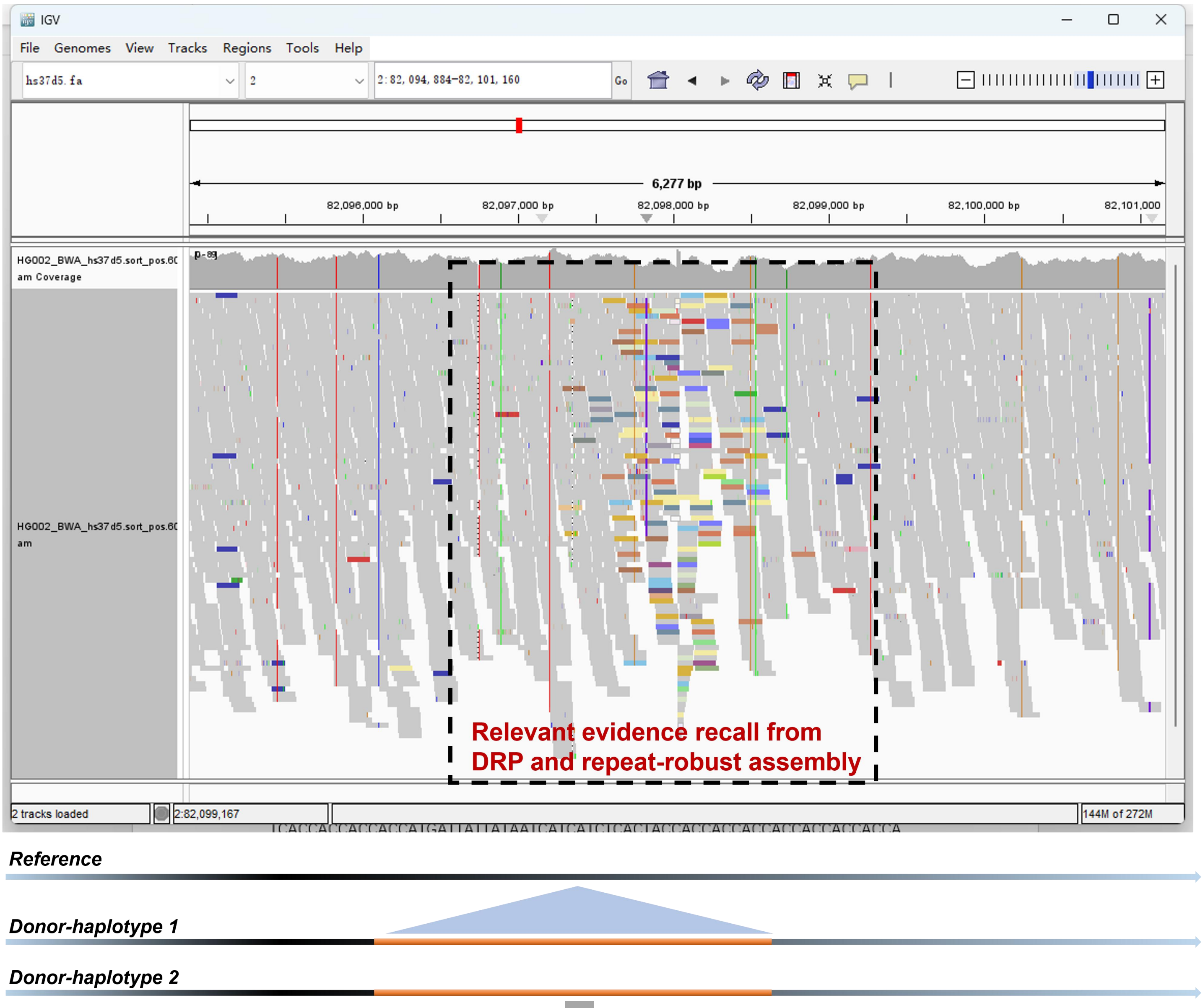
Supplementary Figure 4. Examples of highly matched SV events with divergent representations

The IGV snapshot for several called SVs in a local genome region (hs37d5, Chr1:118,861,163- Chr1:118,862,002). The HG002 ground truth suggests three insertions in one haplotype, i.e., 691 bp, 131 bp and 32 bp, while HitSV detects the same three insertions with identical insertion size (i.e., identical SV representations). However, in the other haplotype, the ground truth suggests a combination of a 740 bp insertion, a 63 bp deletion and a 192 bp insertion. HitSV reports four SV events as 416 bp insertion, 175 bp insertion, 44 bp insertion and 192 bp insertion, i.e., the former three are different from the ground truth linguistically. However, by putting the events back to the reference, the generated alleles of the donor genome by the ground truth and HitSV calls can be exactly matched to each other, indicating that the differences are caused by the various SV representations made by allele realignment, but not real false positive/negative calls in practice.



Supplementary Figure 5. The yields of the callers for various types and sizes of SVs (short read).

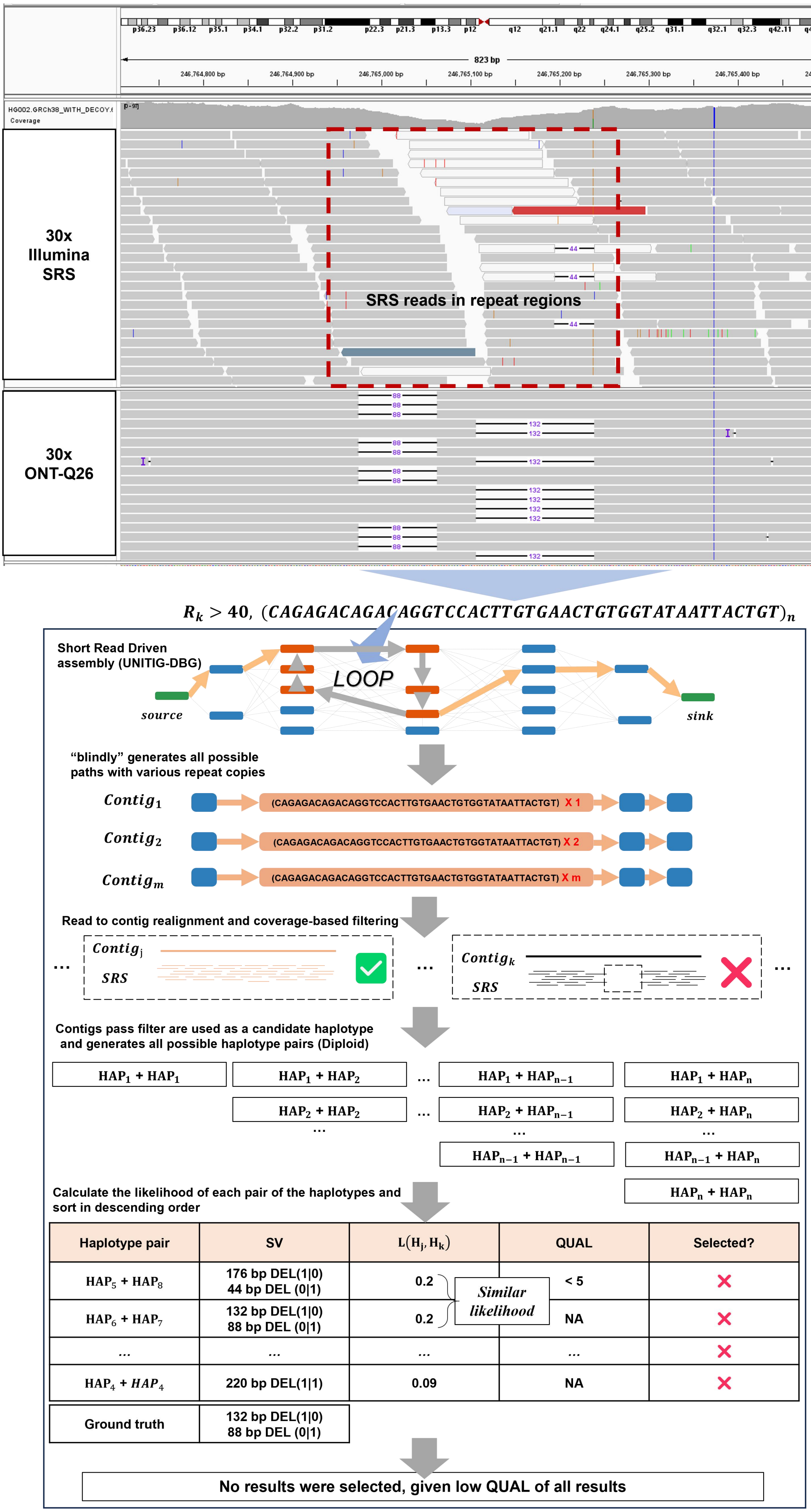
The horizontal axis indicates the types (left: deletions, right: insertions) and sizes (in the ranges of 50-200bp, 200-500bp, 500-1kbp, 1k-2kbp, 2k-5kbp and >5kbp, respectively). The yields of the callers are shown in various colored bars. The dark and light colors indicate the F1-scores with and without genotyping. It is also worthnoting the yield of DRAGEN is on 35x dataset which is directly quoted from its own study. For other callers, their yields on 60x dataset are shown.


IGV
File Genomes View Tracks Regions Tools Help
hs37d5.fa
2
2:82,094,884-82,101,160
Go
6,277 bp
82,096,000 bp
82,097,000 bp
82,098,000 bp
82,099,000 bp
82,100,000 bp
82,101,000
HG002_BWA_hs37d5.sort_pos.60am Coverage
HG002_BWA_hs37d5.sort_pos.60am
Relevant evidence recall from DRP and repeat-robust assembly
2 tracks loaded
2:82,099,167
144M of 272M
TCACCACCACCACCATGATTATTATAATCATCATCTCACTACCACCACCACCACCACCACCACCA
Reference
Donor-haplotype 1
Donor-haplotype 2

Position	2:82098027					
Type	6056-bp large insertion (L1)					
Zygosity	Homozygosity					
Caller	HitSV	DRAGEN	GATK-SV	Manta	Delly	Lumpy
	√	×	×	×	×	×

Supplementary Figure 6. Examples of HitSV to detect large insertions using short reads

The IGV snapshot for a large insertion event at Chr2:82,098,027 which is a typical mobile element (L1) insertion. Using short reads, the event can only be detected by HitSV during the benchmark. HitSV achieves this goal by the refined read clustering which collects plenty of spanning reads by their DRP signatures (the colored ones in the figure). Moreover, the tailored local assembly approach also helps to precisely reconstruct the inserted L1 sequence (6056 bp).



p36.23
p36.12
p35.1
p34.1
p32.2
p31.2
p22.3
p21.3
p13.3
p12
q12
q21.1
q22
q24.1
q25.2
q31.1
q32.1
q32.3
q42.11
q42.3
q44
823 bp
246,764,800 bp
246,764,900 bp
246,765,000 bp
246,765,100 bp
246,765,200 bp
246,765,300 bp
246,765,400 bp
246,765,500 bp
HG002.GRCh38_WITH_DECOY.(
Coverage
[0 - 97]
30x Illumina SRS
SRS reads in repeat regions
44
30x ONT-Q26
88
132
Short Read Driven assembly (UNITIG-DBG)
source
LOOP
sink
"blindly" generates all possible paths with various repeat copies
$Contig_1$
(CAGAGACAGACAGGTCCACTTGTGAACTGTGGTATAATTACTGT) X 1
$Contig_2$
(CAGAGACAGACAGGTCCACTTGTGAACTGTGGTATAATTACTGT) X 2
$Contig_m$
(CAGAGACAGACAGGTCCACTTGTGAACTGTGGTATAATTACTGT) X m
Read to contig realignment and coverage-based filtering
$Contig_j$
SRS
$Contig_k$
SRS
Contigs pass filter are used as a candidate haplotype and generates all possible haplotype pairs (Diploid)
$HAP_1 + HAP_1$
$HAP_1 + HAP_2$
$HAP_1 + HAP_{n-1}$
$HAP_1 + HAP_n$
$HAP_2 + HAP_2$
$HAP_2 + HAP_{n-1}$
$HAP_2 + HAP_n$
$HAP_{n-1} + HAP_{n-1}$
$HAP_{n-1} + HAP_n$
$HAP_n + HAP_n$
Calculate the likelihood of each pair of the haplotypes and sort in descending order
Similar likelihood

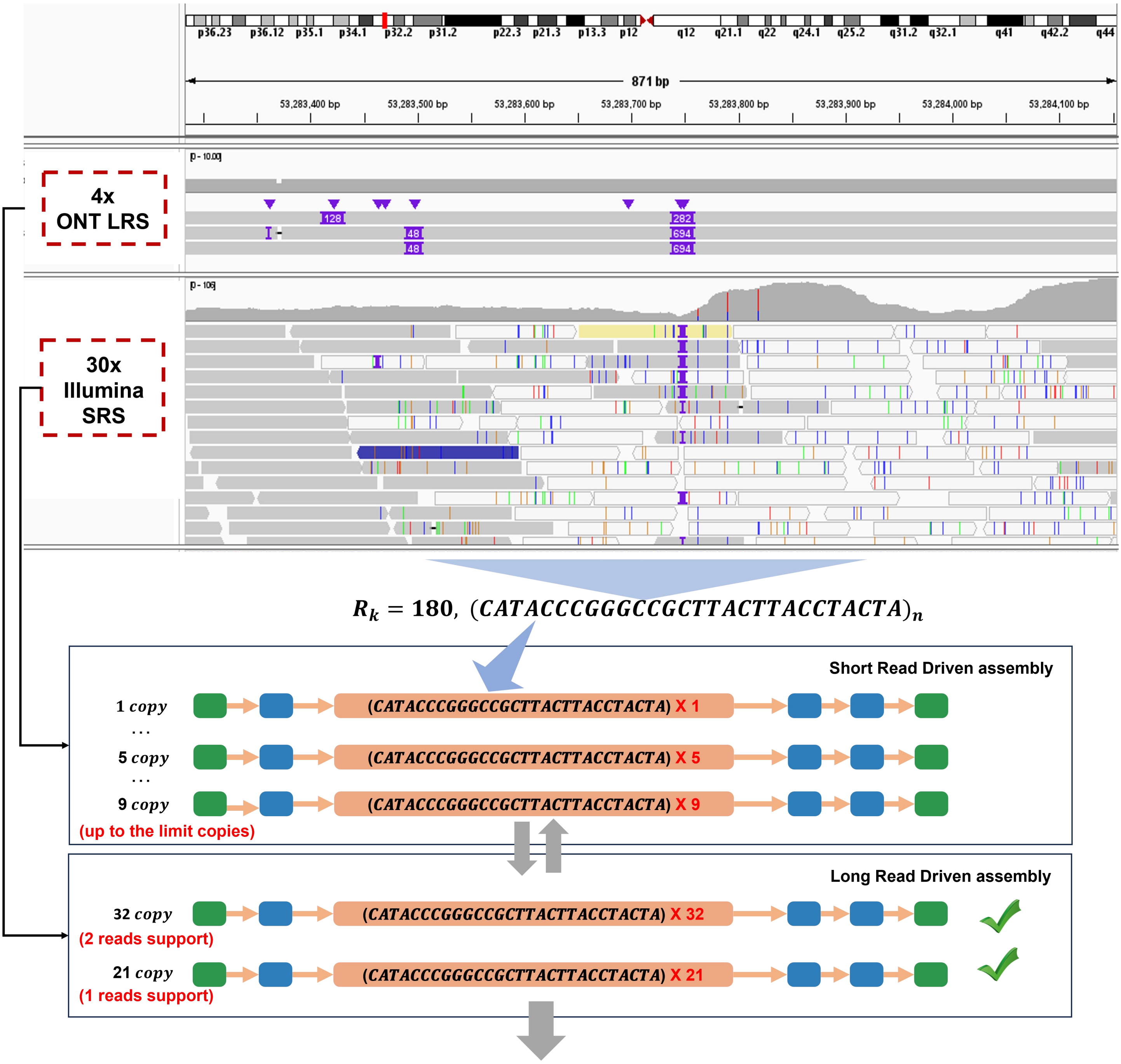
$R_k > 40,\ (CAGAGACAGACAGGTCCACTTGTGAACTGTGGTATAATTACTGT)_n$

Haplotype pair	SV	$L(H_j, H_k)$	QUAL	Selected?
$HAP_5 + HAP_8$	176 bp DEL(1|0) 44 bp DEL (0|1)	0.2	< 5	✗
$HAP_6 + HAP_7$	132 bp DEL(1|0) 88 bp DEL (0|1)	0.2	NA	✗
…	…	…	…	✗
$HAP_4 + HAP_4$	220 bp DEL(1|1)	0.09	NA	✗
Ground truth	132 bp DEL(1|0) 88 bp DEL (0|1)			

No results were selected, given low QUAL of all results

Supplementary Figure 7. A schematic illustration on the failure of SV detection even when the SV alleles themselves were finely reconstructed.

The IGV snapshot shows a tandem-repeat locus at chr1:246,764,708-246,765,529, composed of a 44-bp repeat unit (CAGAGACAGACAGGTCCACTTGTGAACTGTGGTATAATTACTGT). In the truth set, HG002 is heterozygous, carrying two haplotypes with 6 and 7 repeat copies, corresponding to 132-bp and 88-bp deletions relative to the reference, which contains 9 repeat copies. HitSV collects all reads in the local region and constructs a localized unitig-based de Bruijn graph, then enumerates all feasible paths under node-count constraints to generate candidate contigs. Reads are aligned back to these contigs, and only those fully supported by read coverage are retained and added to the haplotype pool. All retained haplotypes are subsequently paired, and for each pair HitSV evaluates the likelihood of being the final solution by jointly considering consistency between read and contig k-mer frequencies as well as the requirement that all reads are explained by the assembly. In this locus, two combinations—5+8 and 6+7 repeat copies—receive identical composite scores, leading to indistinguishable haplotype likelihoods. As a result, HitSV cannot resolve the optimal configuration and fails to report the correct genotype for this VNTR region.

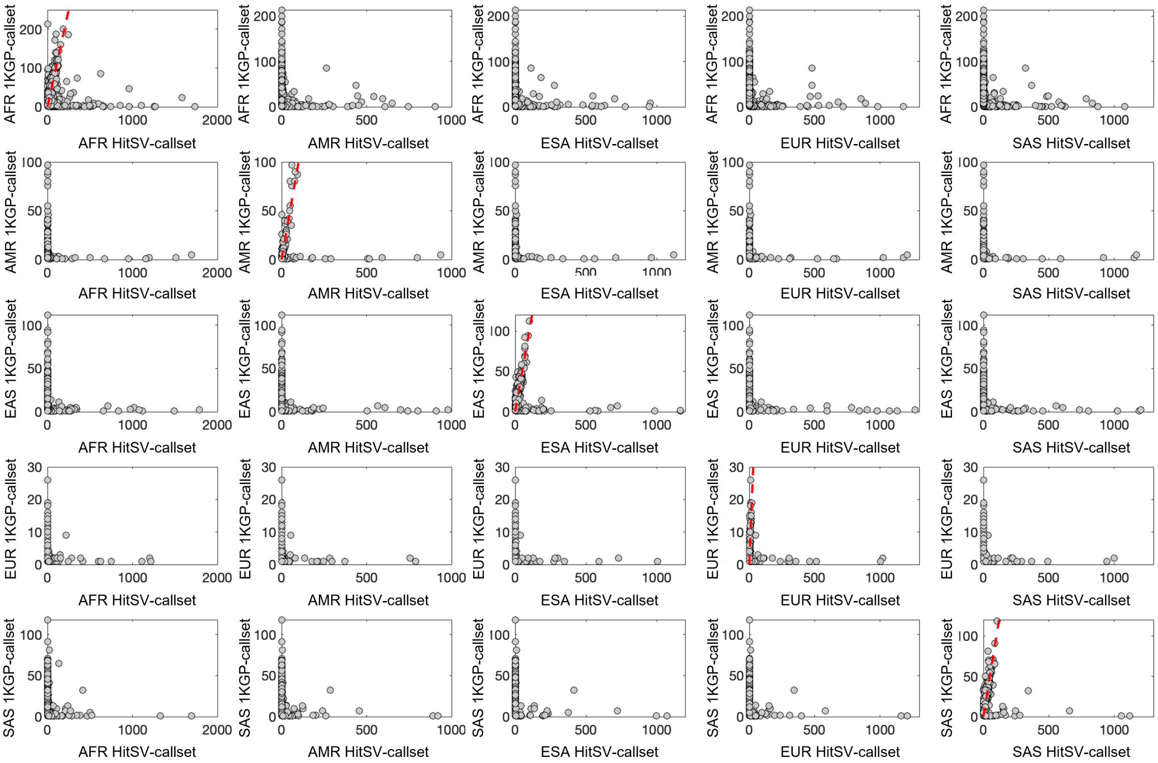

p36.23 p36.12 p35.1 p34.1 p32.2 p31.2 p22.3 p21.3 p13.3 p12 q12 q21.1 q22 q24.1 q25.2 q31.2 q32.1 q41 q42.2 q44
871 bp
53,283,400 bp
53,283,500 bp
53,283,600 bp
53,283,700 bp
53,283,800 bp
53,283,900 bp
53,284,000 bp
53,284,100 bp
4x ONT LRS
[0 - 10.00]
128
48
48
282
694
694
30x Illumina SRS
[0 - 106]
$R_k = 180$, $(CATACCCGGGCCGCTTACTTACCTACTA)_n$
Short Read Driven assembly
1 copy
...
5 copy
...
9 copy
(up to the limit copies)
(CATACCCGGGCCGCTTACTTACCTACTA) X 1
(CATACCCGGGCCGCTTACTTACCTACTA) X 5
(CATACCCGGGCCGCTTACTTACCTACTA) X 9
Long Read Driven assembly
32 copy
(2 reads support)
21 copy
(1 reads support)
(CATACCCGGGCCGCTTACTTACCTACTA) X 32
(CATACCCGGGCCGCTTACTTACCTACTA) X 21

Caller	Alleles in Hap1	Alleles in Hap2	SV allele match	Zygosity match	Sequence match
HitSV	chr1:53283531 410 INS (0|1)	chr1:53283531 694 INS (1|0)	√	√	√

Supplementary Figure 8. A schematic illustration on the successful of SV allele reconstruction and genotyping with evident long reads.

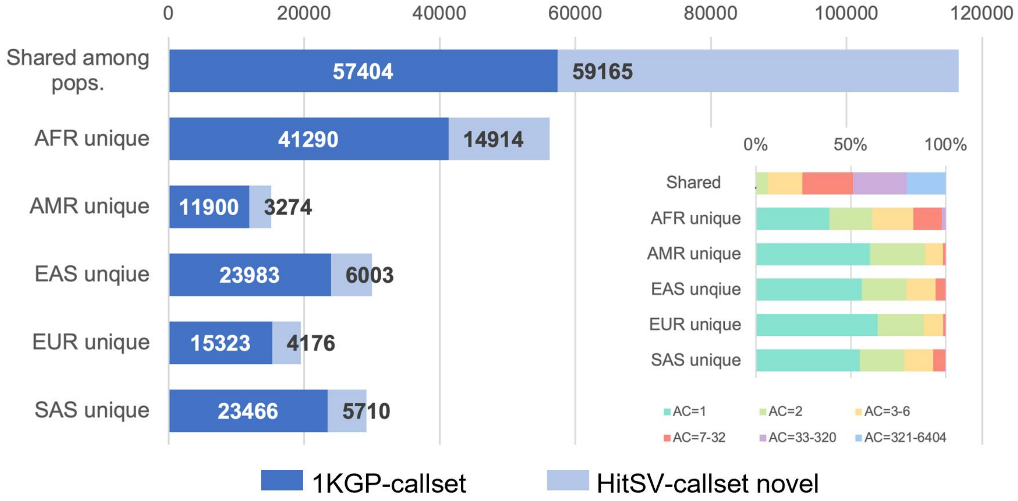
The IGV snapshot for a VNTR in a highly repetitive region (GRCh38, chr1:53,283,531-53,283,934, $R_k = 180$) highlights the improvement of evident long reads during SV allele reconstruction. The reference genome (GRCh38) contains 10 tandem copies of a 28-bp repeat unit. Two target haplotypes harbor 32 and 21 repeat units, corresponding to a 694-bp and a 410-bp insertion, respectively. (Bottom) In the 30x Illumina dataset, the high local repetitiveness prevents short reads from spanning the SV site. Despite heuristic graph walking, the reconstructed alleles are truncated (1–9 copies), failing to recover the full SV architecture. (Top) In the 4x ONT dataset, spanning long reads (one for the 410-bp and two for the

694-bp alleles) provide sufficient structural scaffolds. HitSV leverages these evidential long reads to successfully reconstruct both divergent contigs, enabling accurate SV calling and genotyping in regions recalcitrant to short-read-only methods.



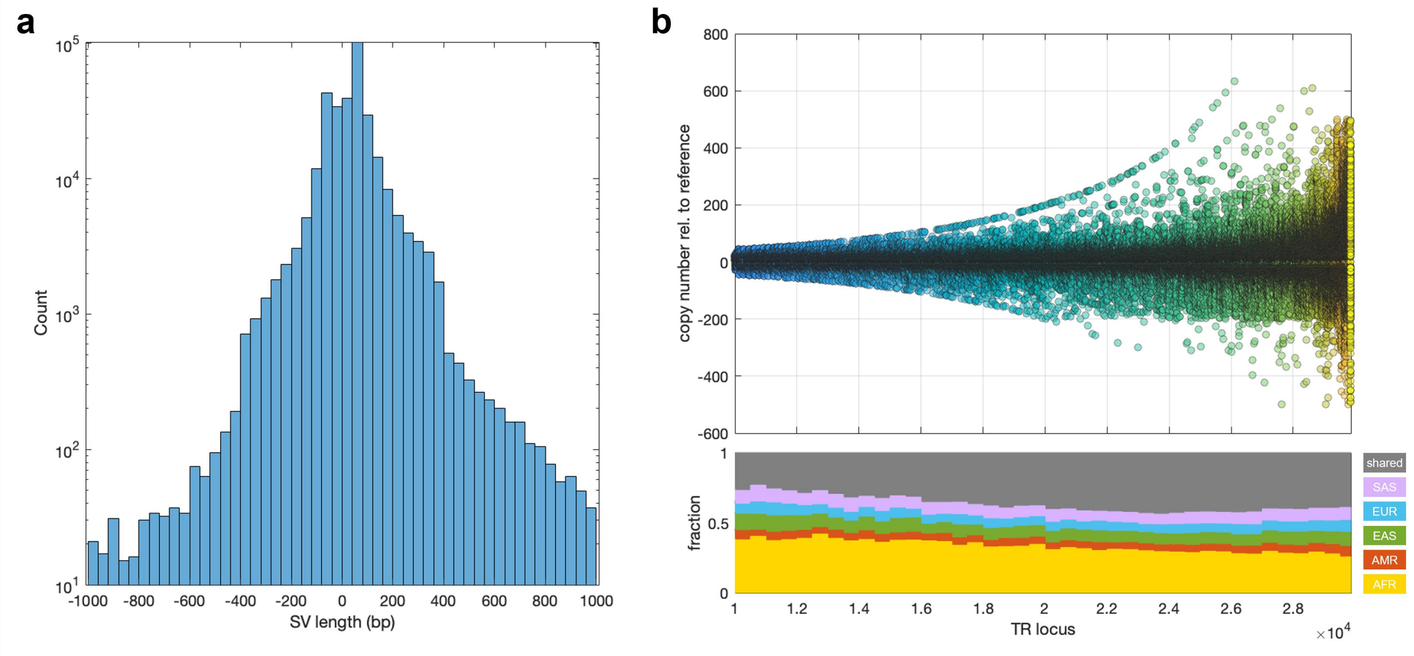
Supplementary Figure 9. Allele counts of population-specific SVs in HitSV-callset and 1KGP-callset

Only SVs regarded unique to the superpopulation indicated on the y-axis in the 1KGP-callset were plotted. The allele count of these SVs in the same superpopulation (diagonal) or in another superpopulation, in the HitSV-callset, is shown on the x-axis. The allele count of these SVs in the 1KGP-callset is shown on the y-axis. In the diagonal subplots, the red dashed lines mark the 1:1 ratio. In the main text, SVs with strongly elevated allele frequencies are those in the diagonal subplots whose allele frequencies in HitSV-callset became at least 2-fold of their allele frequencies in 1KGP-callset, in the same superpopulation.



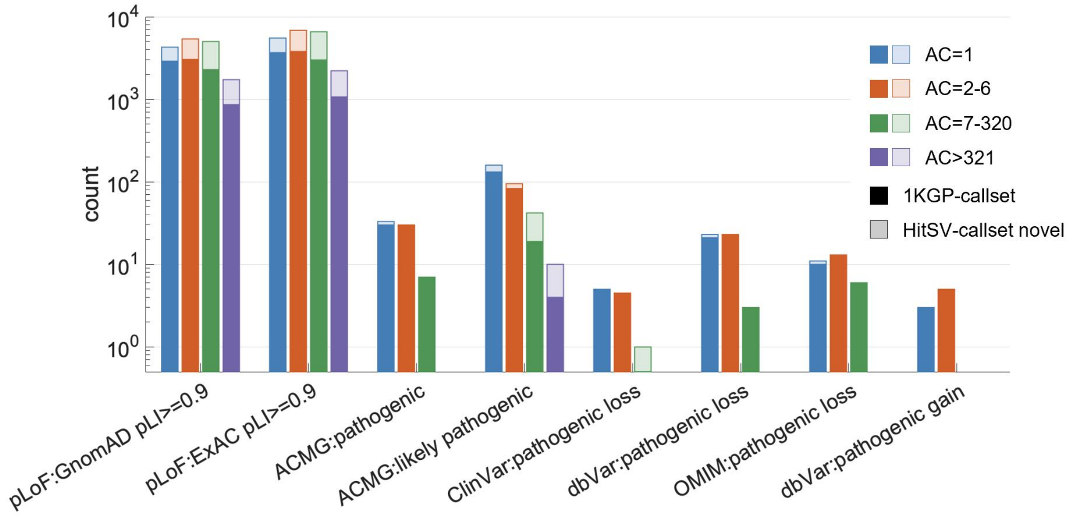
Supplementary Figure 10. Numbers of population unique and shared SVs in 1KGP-callset.

Numbers of population unique and shared SVs in 1KGP-callset (dark blue) and distinctly identified in HitSV-callset (light blue). Inset shows the allele count distribution of SVs distinctly identified in HitSV-callset.



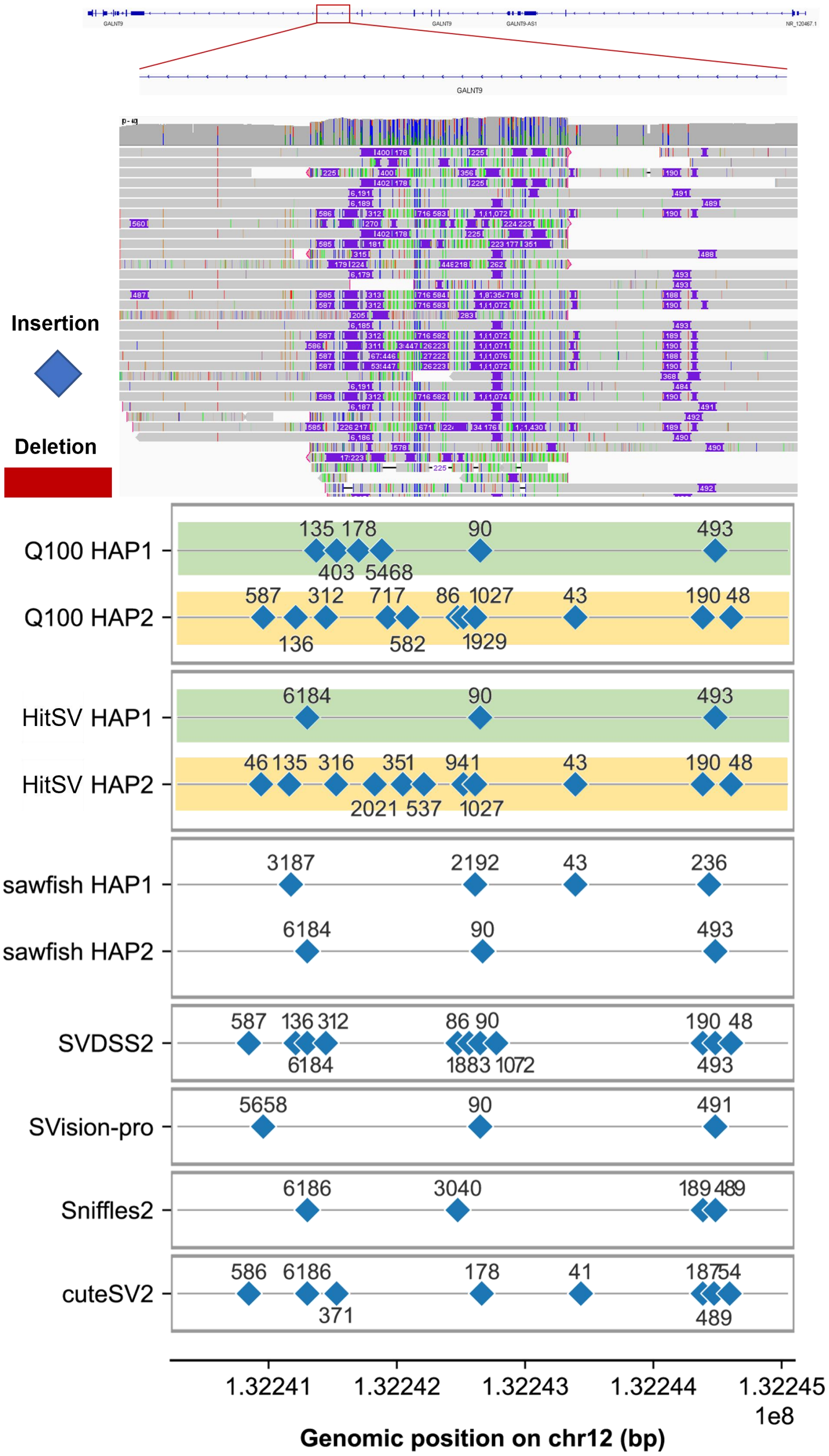
Supplementary Figure 11. Tandem repeat-associated SVs in the HitSV-callset

a, The length distribution of tandem repeat-associated SVs in HitSV-callset, shown on the logarithmic scale. **b,** Copy number variations (upper panel) and the fractions of population unique SVs (lower panel) at each tandem repeat locus. The tandem repeat loci are sorted on the x-axis by increasing variance.



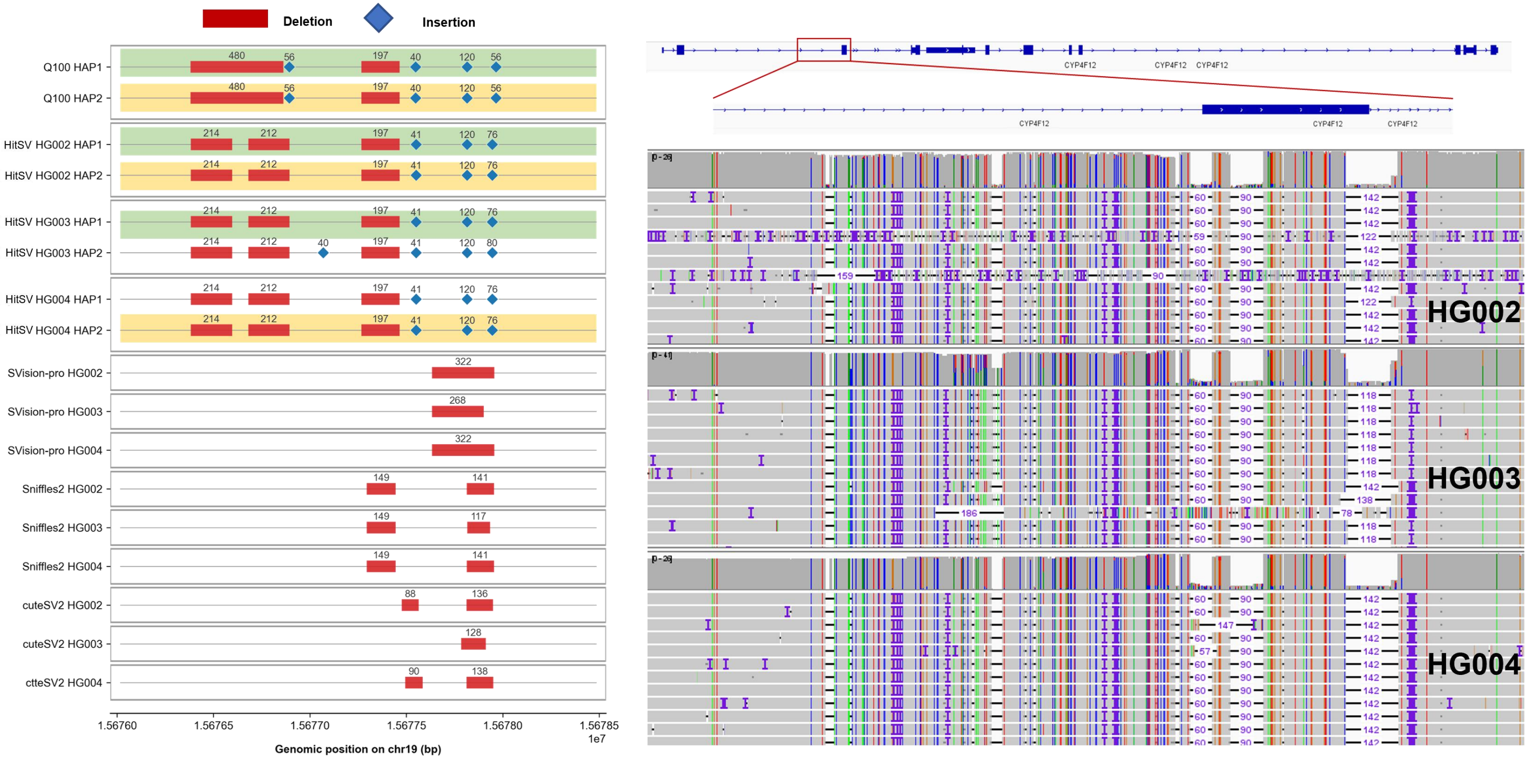
Supplementary Figure 12. Phenotypic annotation of SVs in the 1KGP-callset and HitSV-callset

Phenotypic annotation of SVs in the 1KGP-callset (dark colors) and SVs distinctly identified in HitSV-callset (boxed light colors). For each annotation group, SVs are partitioned into four allele frequency bins.



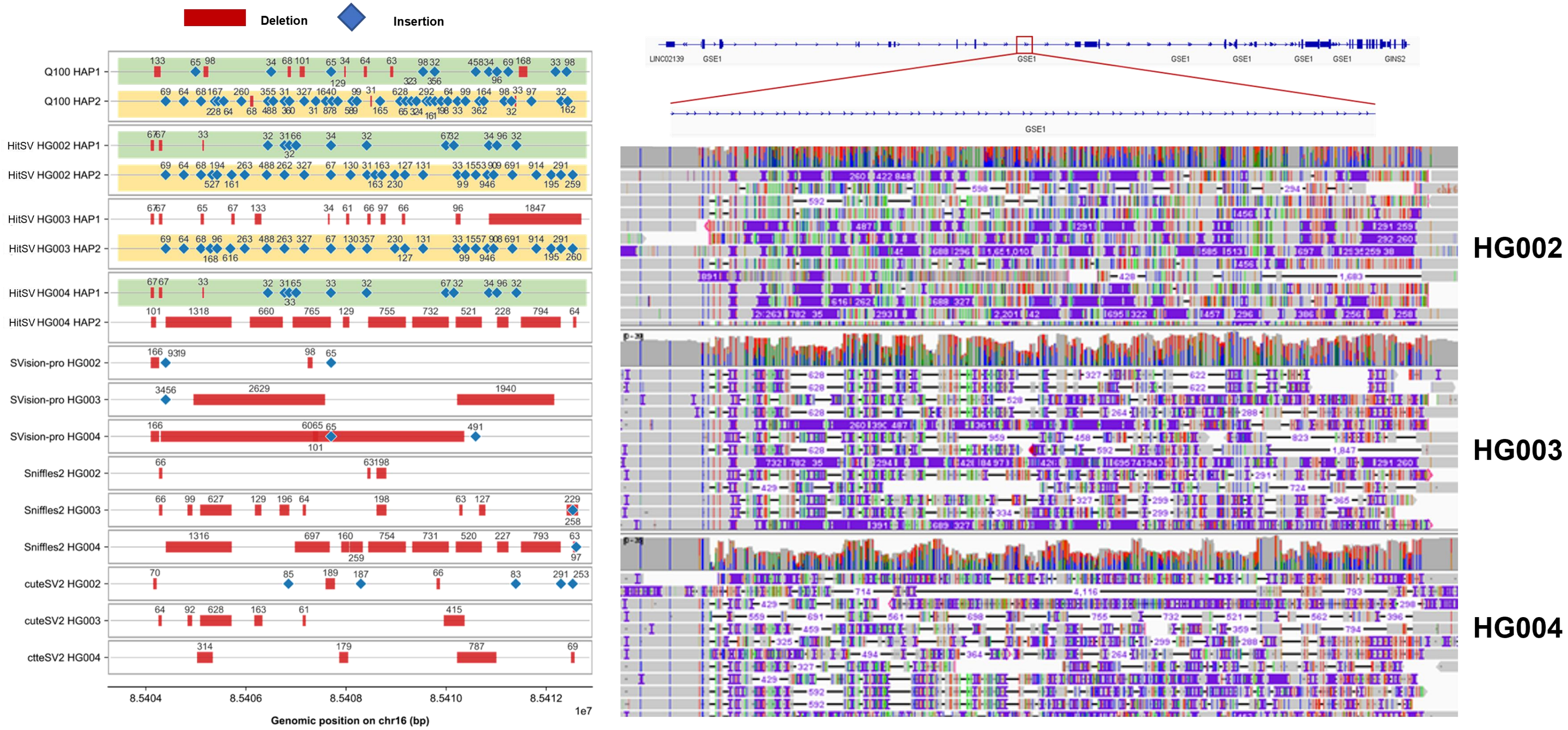
Supplementary Figure 13. An example of CSV detection in a HG002 dataset.

An example of CSV detection in an intronic region of GALNT9 on chromosome 12 (chr12:132,239,289–132,246,050 bp) is shown. The upper panel presents an Integrative Genomics Viewer (IGV) snapshot of this region with heterogeneous SV signatures from read alignment. The lower panel summarizes the SVs detected by different tools in this region, displaying their genomic positions and variant lengths. Q100 HAP1 and Q100 HAP2 represent the ground-truth SVs on the two haplotypes. Among the evaluated methods, HitSV and sawfish provide haplotype-resolved results and can distinguish SVs on HAP1 and HAP2, whereas the remaining tools lack phasing capability and therefore report all detected SVs without haplotype assignment. Notably, HitSV shows high concordance with the ground truth in both the genomic distribution and length of detected variants, whereas other tools exhibit substantial discrepancies.



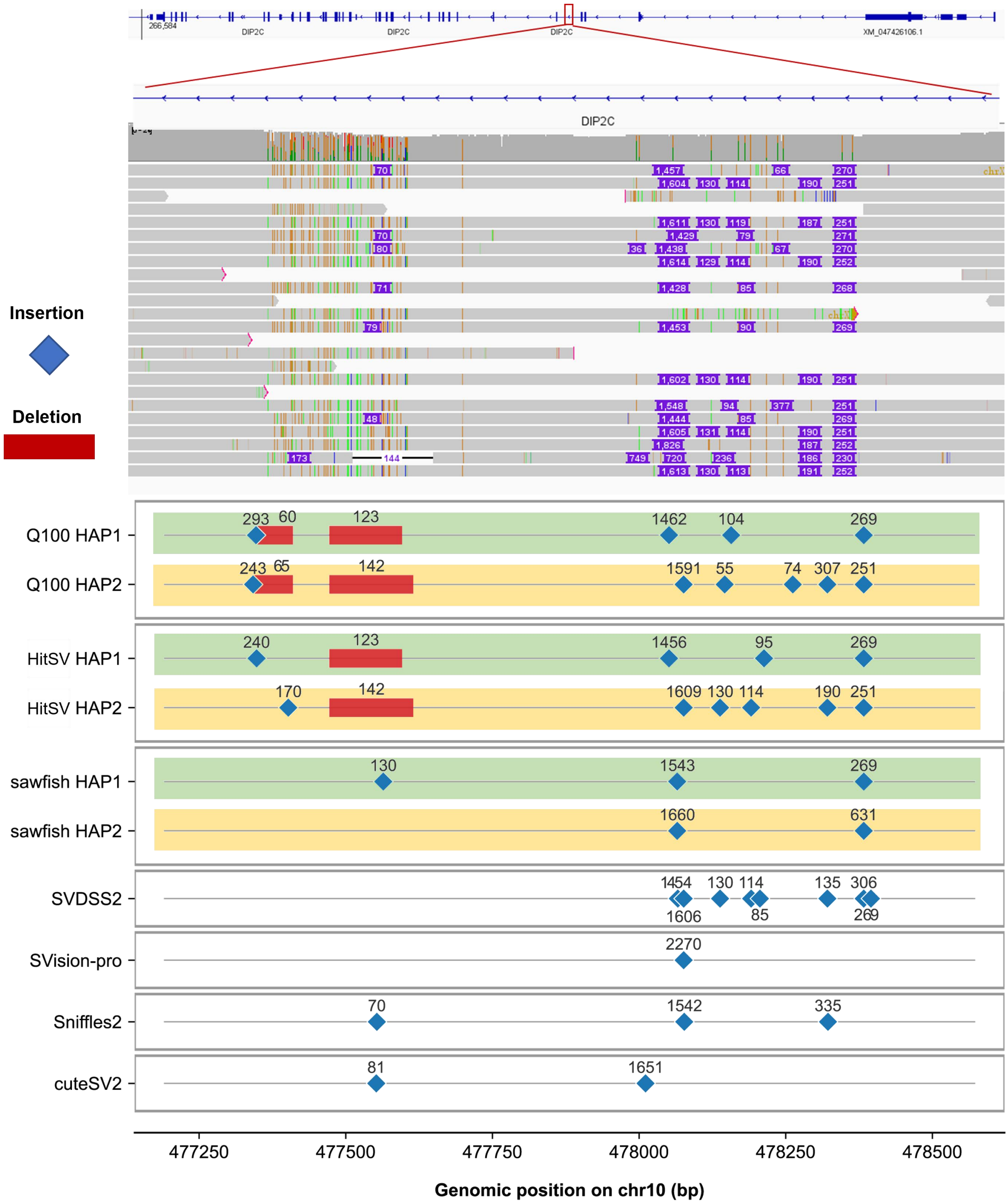
Supplementary Figure 14. An example of CSV detection in a family-based dataset.

The analysis focuses on a local genomic region encompassing the CYP4F12 locus on chromosome 19 (chr19:15,677,010-15,678,477 bp), illustrating CSV inheritance patterns across related individuals. The left panel summarizes the SVs detected by different tools across haplotypes and samples. Among the evaluated methods, HitSV provides haplotype-resolved calls and enables consistent tracking of CSVs across family members, facilitating the identification of shared and inherited variants, whereas other tools either lack phasing capability or fail to recover the complete set of events. This example highlights the advantage of HitSV in accurately resolving complex SV structures and their transmission patterns in pedigree-based analyses. The right panel presents IGV snapshots of the same region for HG002, HG003, and HG004, showing read alignments contain a series of concordantly mapped small divergences.



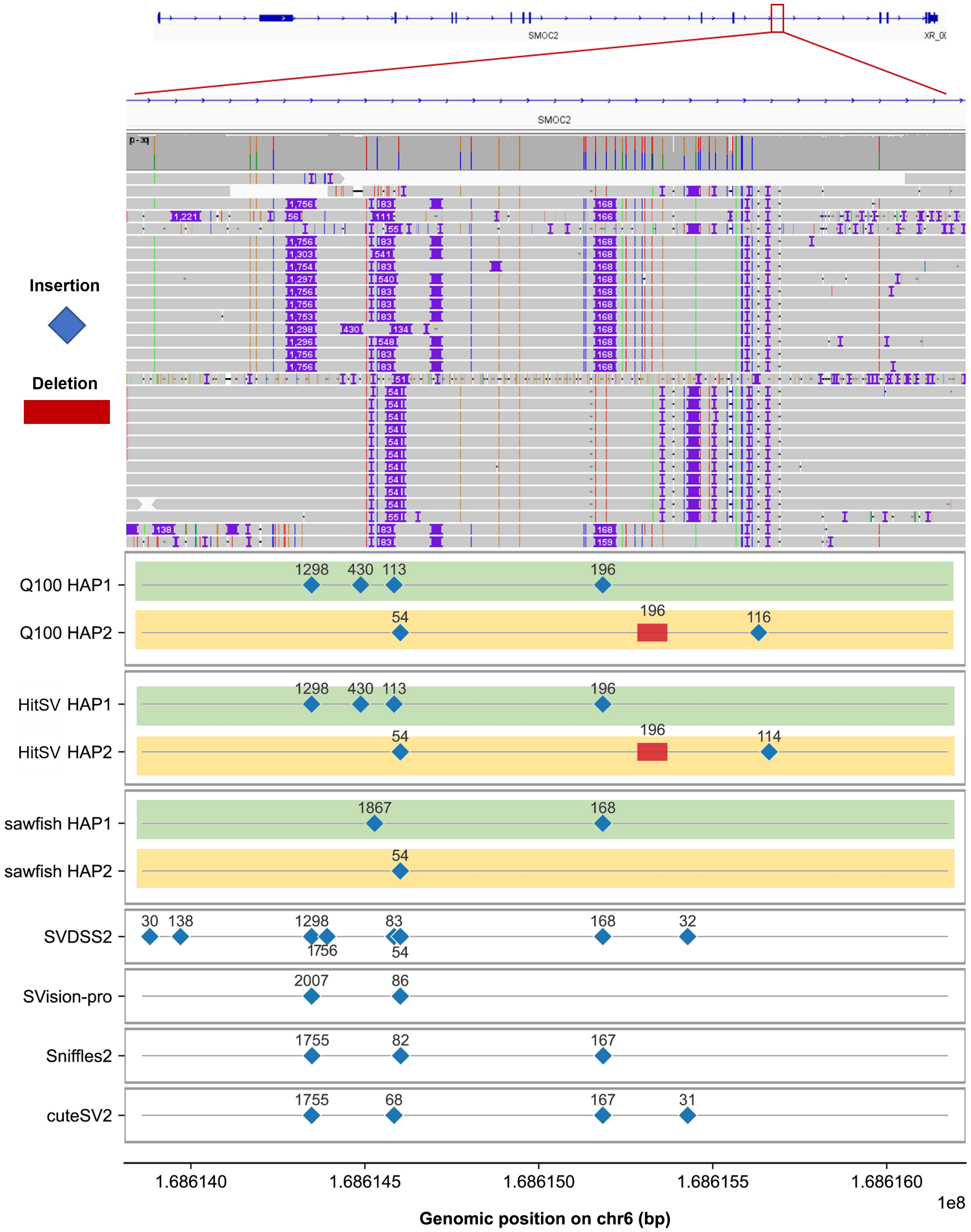
Supplementary Figure 15. An example of CSV detection in a family-based dataset.

The analysis focuses on a local genomic region encompassing the GSE1 locus on chromosome 16 (chr16:85402995-85413170 bp), illustrating CSV inheritance patterns across related individuals. The left panel summarizes the SVs detected by different tools across haplotypes and samples. Among the evaluated methods, HitSV produces haplotype-resolved calls and consistently recovers concordant SV patterns across family members, enabling reliable tracking of inherited variants. In contrast, other tools either lack haplotype resolution or show fragmented, inconsistent, or missing calls. This example highlights the advantage of HitSV in accurately resolving complex SV structures and their transmission patterns in pedigree-based analyses. The right panel presents IGV snapshots of the same region for HG002, HG003, and HG004, showing read alignments and supporting signals consistent with complex SV events.



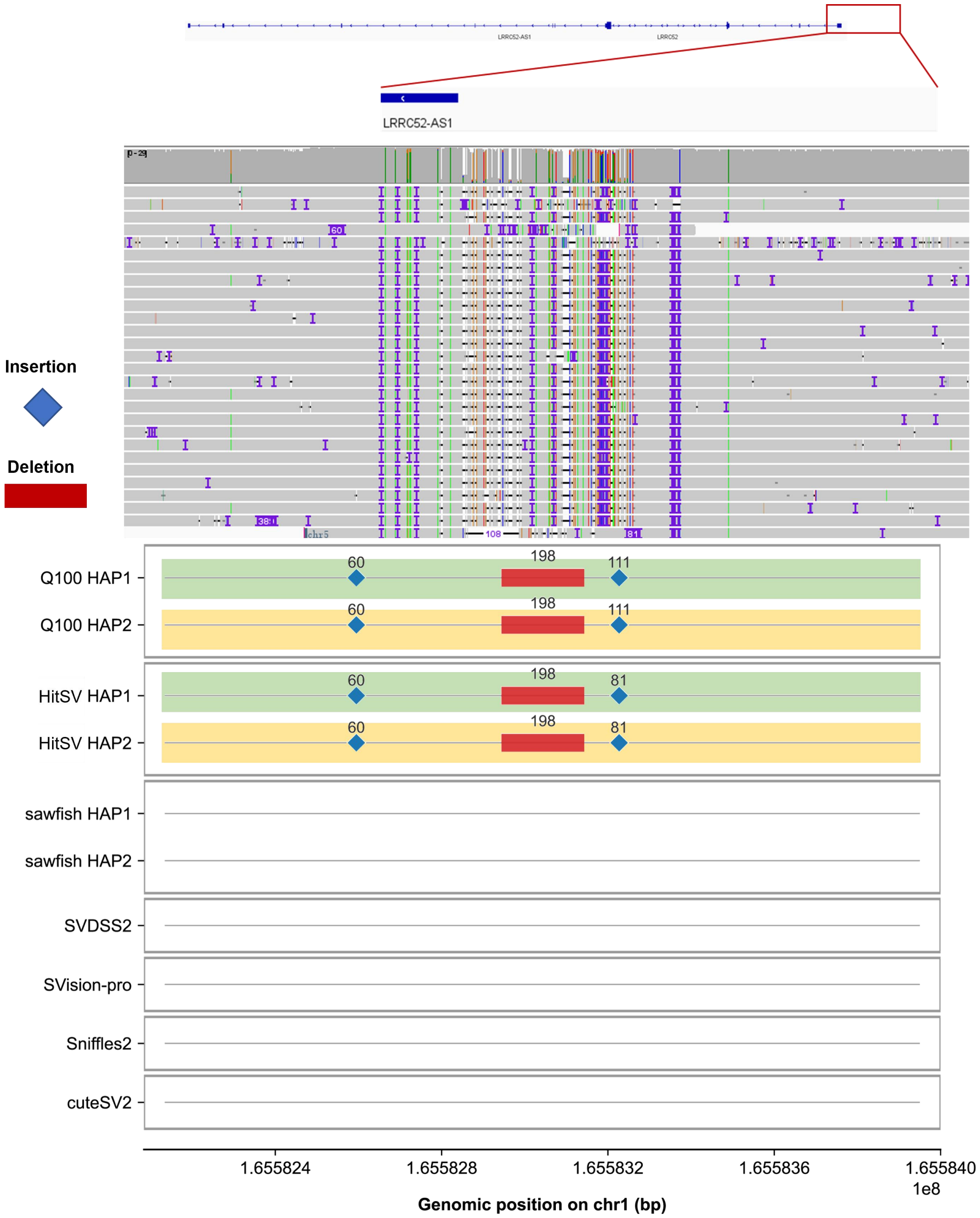
Supplementary Figure 16. An example of CSV detection in a HG002 dataset.

An example of CSV detection in an intronic region of DIP2C on chromosome 10 (chr10:477,090-478,673 bp) is shown. The upper panel presents an IGV snapshot of this region, illustrating the read alignments with heterogeneous SV signatures. The lower panel summarizes the SVs detected by different tools in this region, displaying their genomic positions and variant lengths. Q100 HAP1 and Q100 HAP2 represent the ground-truth SVs on the two haplotypes. Among the evaluated methods, HitSV and sawfish provide haplotype-resolved results and can distinguish SVs on HAP1 and HAP2, whereas the remaining tools lack phasing capability and therefore report all detected SVs without haplotype assignment. Notably, HitSV shows high concordance with the ground truth in both the genomic distribution and length of detected variants, whereas other tools exhibit substantial discrepancies.



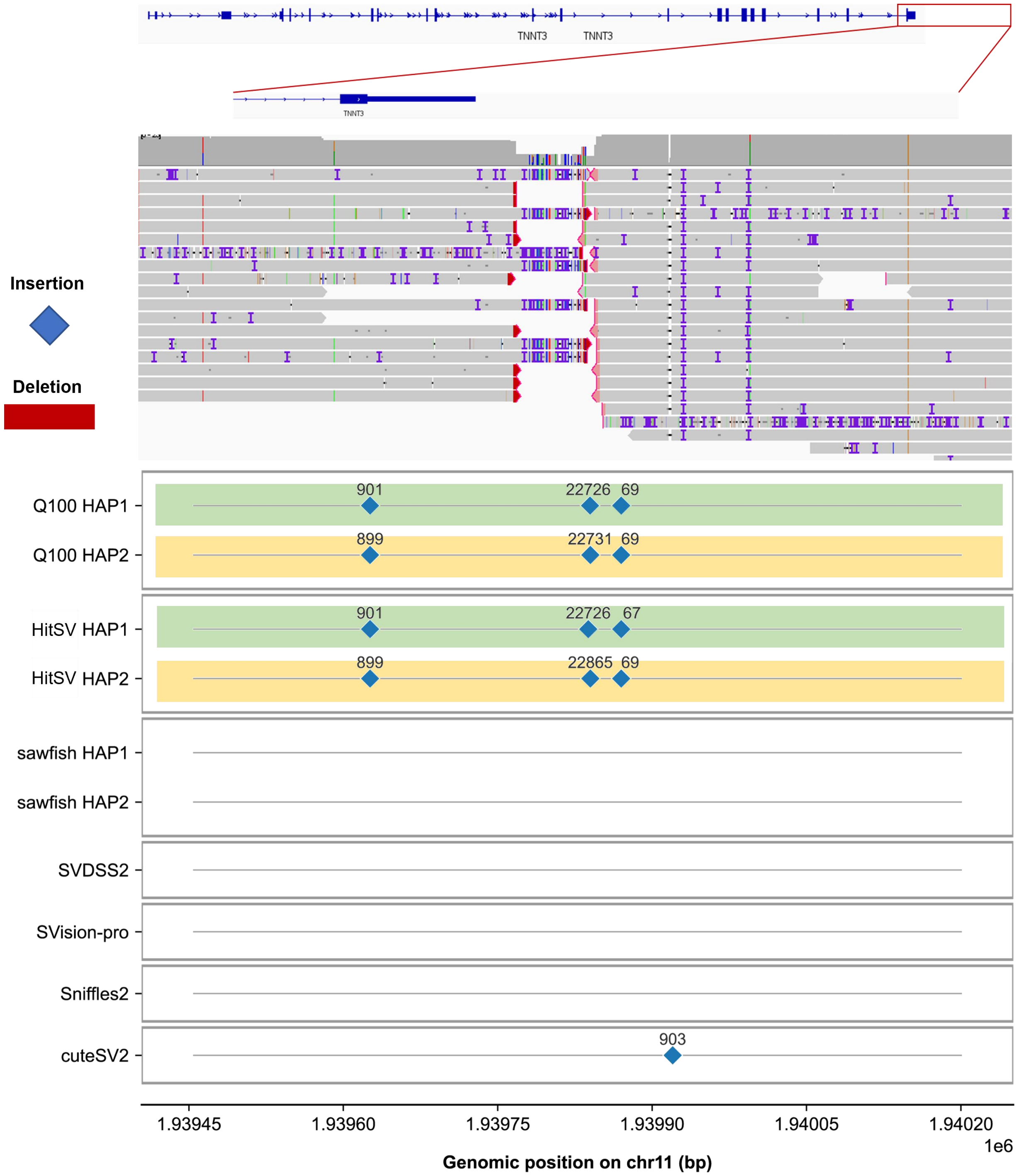
Supplementary Figure 17. An example of CSV detection in a HG002 dataset.

An example of CSV detection in an intronic region of SMOC2 on chromosome 6 (chr6:168,613,859-168,616,177 bp) is shown. The upper panel presents an IGV snapshot of this region, illustrating the read alignments containing a series of concordantly mapped small divergences. The lower panel summarizes the SVs detected by different tools in this region, displaying their genomic positions and variant lengths. Q100 HAP1 and Q100 HAP2 represent the ground-truth SVs on the two haplotypes. Among the evaluated methods, HitSV and sawfish provide haplotype-resolved results and can distinguish SVs on HAP1 and HAP2, whereas the remaining tools lack phasing capability and therefore report all detected SVs without haplotype assignment. Notably, HitSV achieves complete concordance with the ground truth in both variant distribution and length, and the 196-bp deletion is uniquely detected by HitSV, underscoring its superior sensitivity and accuracy in resolving complex SVs.



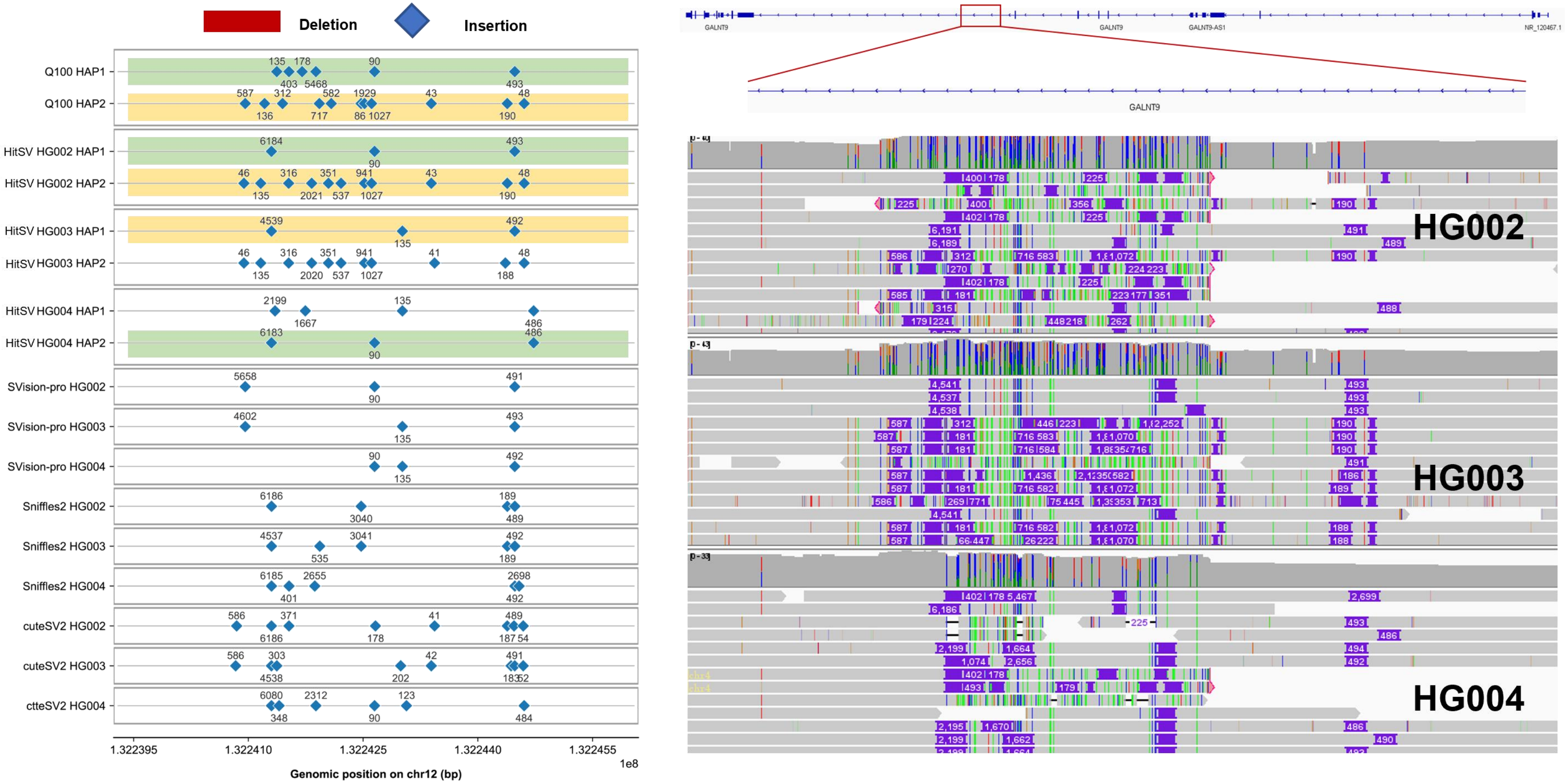
Supplementary Figure 18. An example of CSV detection in a HG002 dataset.

An example of CSV detection in an exonic region of LRRC52 on chromosome 1 (chr1:165,582,134-165,583,950 bp) is shown. The upper panel presents an IGV snapshot of this region, illustrating the read alignments containing a series of concordantly mapped small divergences. The lower panel summarizes the SVs detected by different tools in this region, displaying their genomic positions and variant lengths. Q100 HAP1 and Q100 HAP2 represent the ground-truth SVs on the two haplotypes. Among the evaluated methods, HitSV and sawfish provide haplotype-resolved results and can distinguish SVs on HAP1 and HAP2, whereas the remaining tools lack phasing capability and therefore report all detected SVs without haplotype assignment. Notably, HitSV achieves complete concordance with the ground truth in both variant distribution and length, while no variants are detected by the other methods.



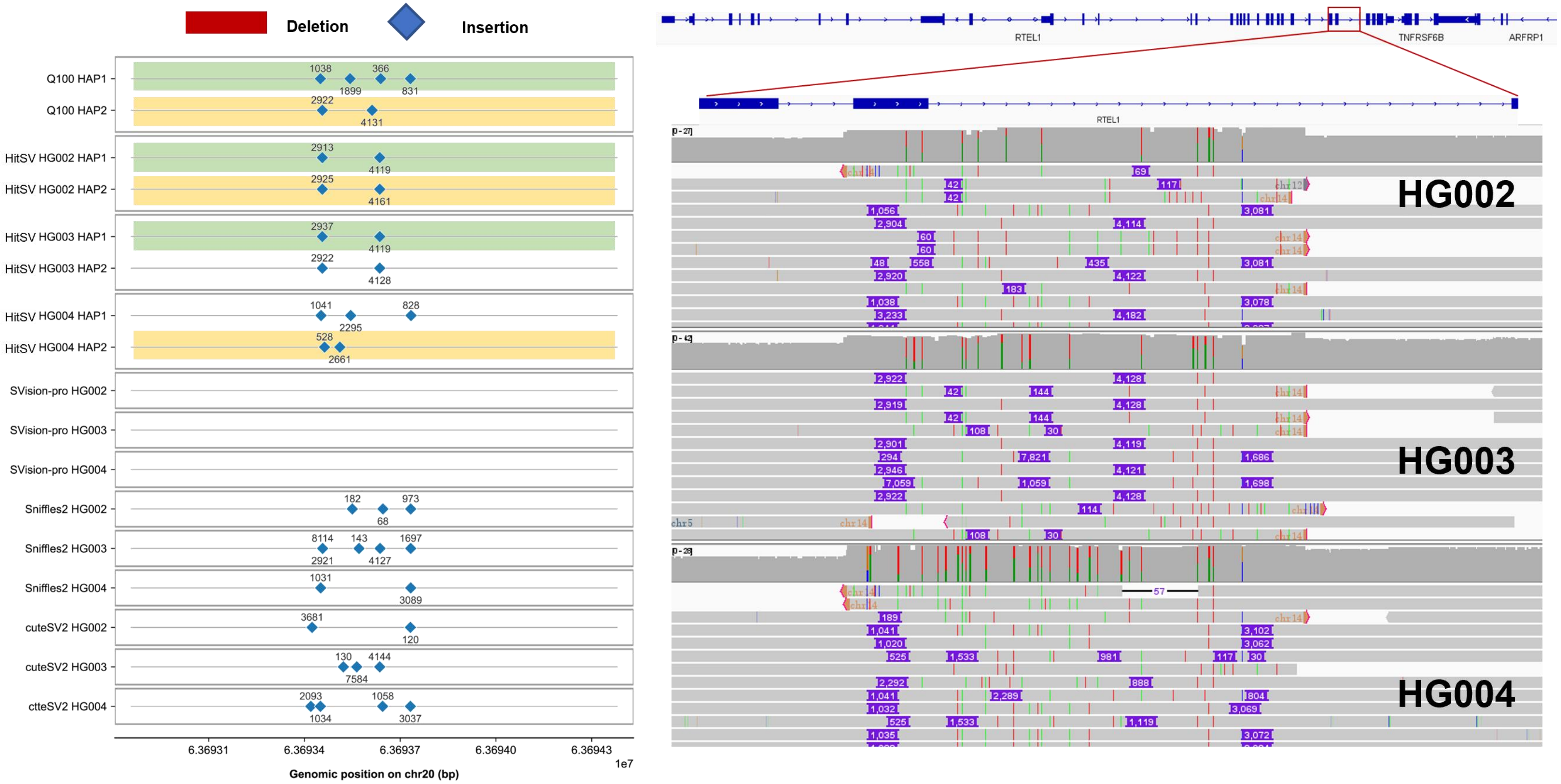
Supplementary Figure 19. An example of CSV detection in a HG002 dataset.

An example of CSV detection in an exonic region of TNNT3 on chromosome 11 (chr11:1,939,454-1,940,201 bp) is shown. The upper panel presents an IGV snapshot of this region, illustrating abnormal read alignments including large soft clipping, and split. The lower panel summarizes the SVs detected by different tools in this region, displaying their genomic positions and variant lengths. Q100 HAP1 and Q100 HAP2 represent the ground-truth SVs on the two haplotypes. Among the evaluated methods, HitSV and sawfish provide haplotype-resolved results and can distinguish SVs on HAP1 and HAP2, whereas the remaining tools lack phasing capability and therefore report all detected SVs without haplotype assignment. Notably, HitSV achieves complete concordance with the ground truth in both variant distribution and length, while no variants are detected by the other methods except cuteSV2 identifies a 903-bp insertion.



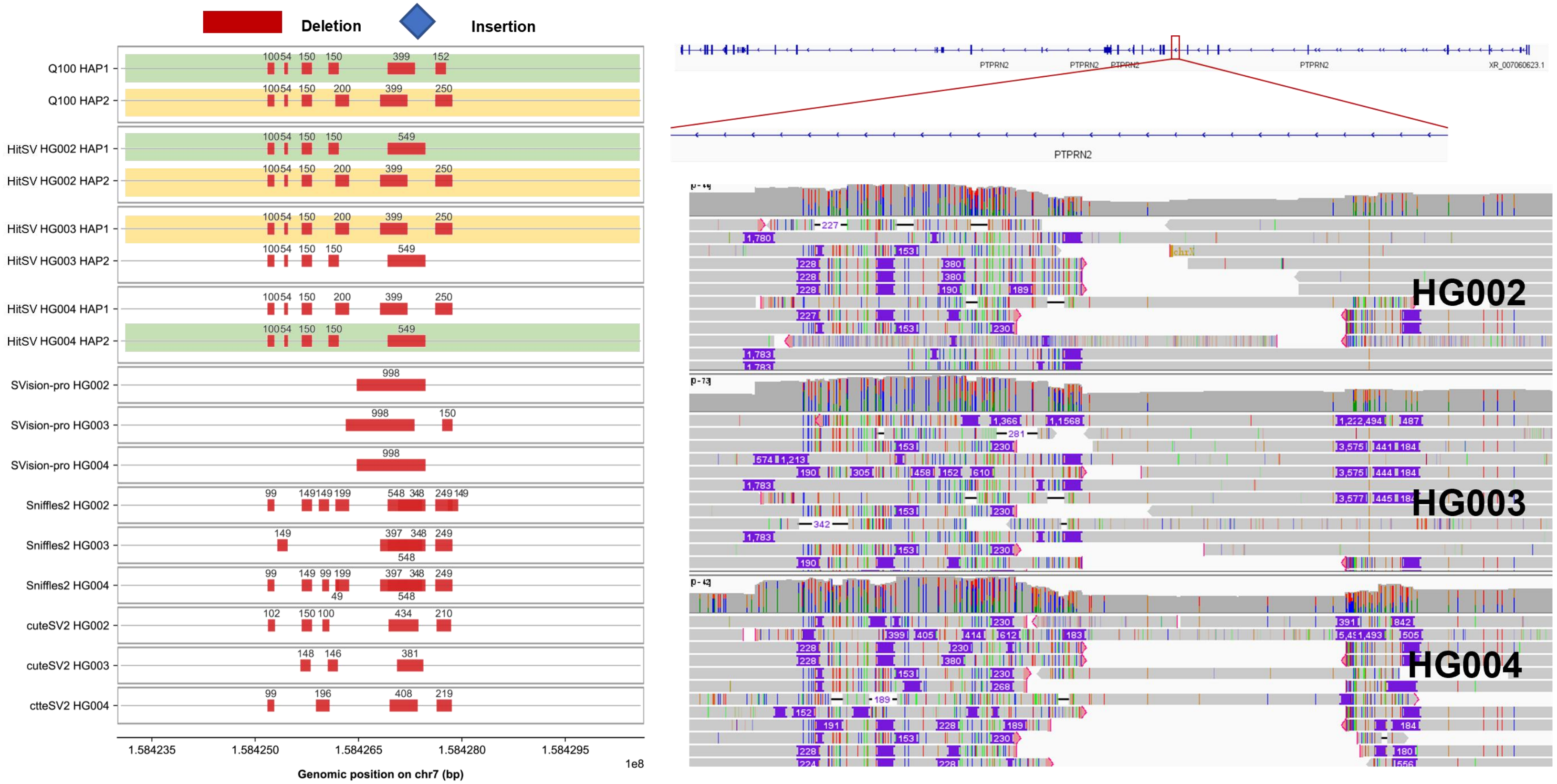
Supplementary Figure 20. An example of CSV detection in a family-based dataset.

The analysis focuses on a local genomic region encompassing the GALNT9 locus on chromosome 12 (chr12:132,239,289-132,246,050 bp), illustrating CSV inheritance patterns across related individuals. The left panel summarizes the SVs detected by different tools across haplotypes and samples. Among the evaluated methods, HitSV produces haplotype-resolved calls and consistently recovers concordant SV patterns across family members, enabling reliable tracking of inherited variants. In contrast, other tools either lack haplotype resolution or show fragmented, inconsistent, or missing calls. This example highlights the advantage of HitSV in accurately resolving complex SV structures and their transmission patterns in pedigree-based analyses. The right panel presents IGV snapshots of the same region for HG002, HG003, and HG004, showing read alignments with heterogeneous SV signatures.



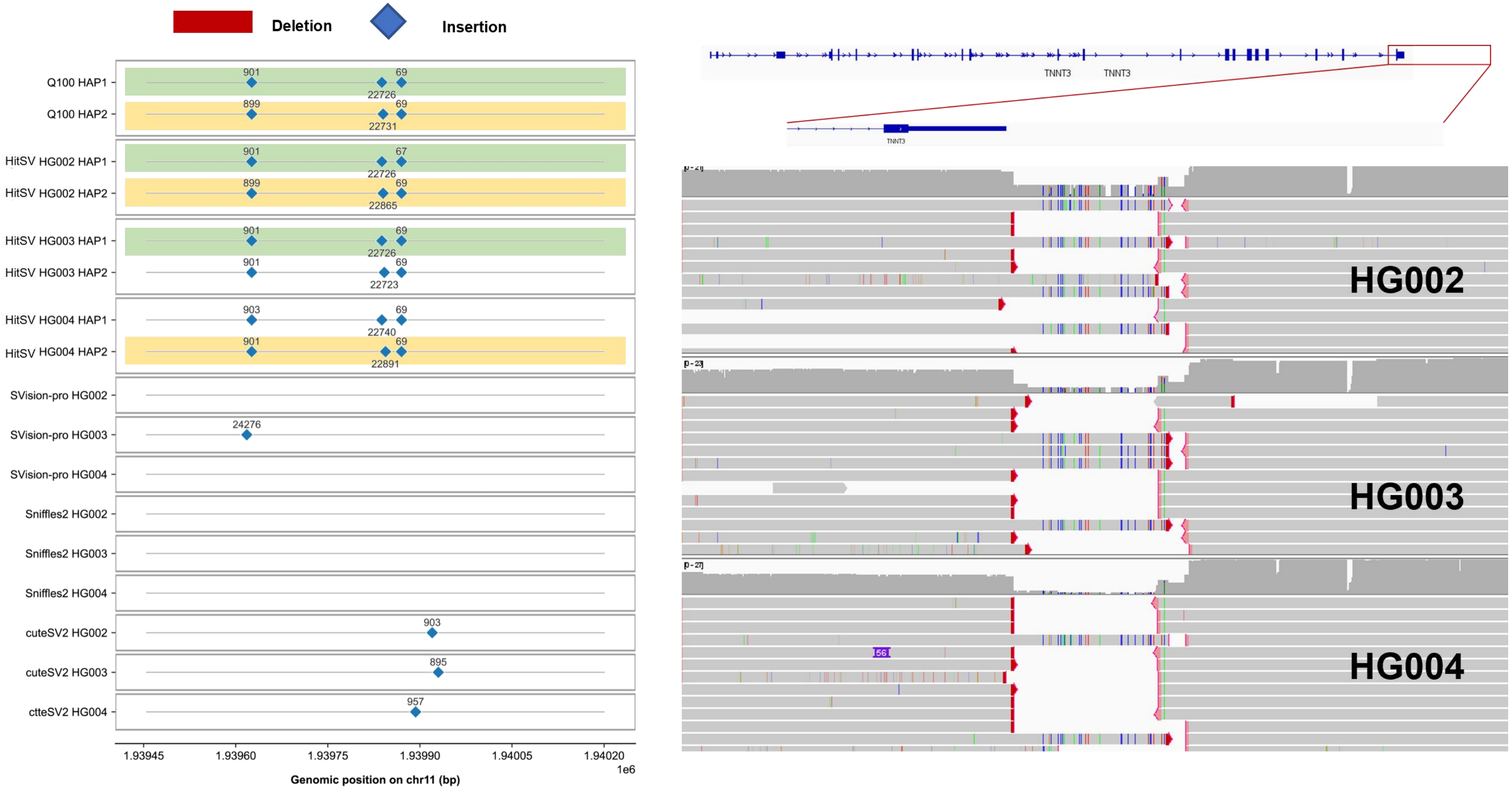
Supplementary Figure 21. An example of CSV detection in a family-based dataset.

The analysis focuses on a local genomic region encompassing the RTEL1 locus on chromosome 20 (chr20:63,692,856-63,694,381 bp), illustrating CSV inheritance patterns across related individuals. The left panel summarizes the SVs detected by different tools across haplotypes and samples. HitSV produces haplotype-resolved calls and consistently recovers SVs that are concordant with the ground truth and shared across family members, enabling reliable tracing of inherited CSVs. The right panel presents IGV snapshots of the same region for HG002, HG003, and HG004, showing abnormal read alignments signatures including large soft clipping, and split.



Supplementary Figure 22. An example of CSV detection in a family-based dataset.

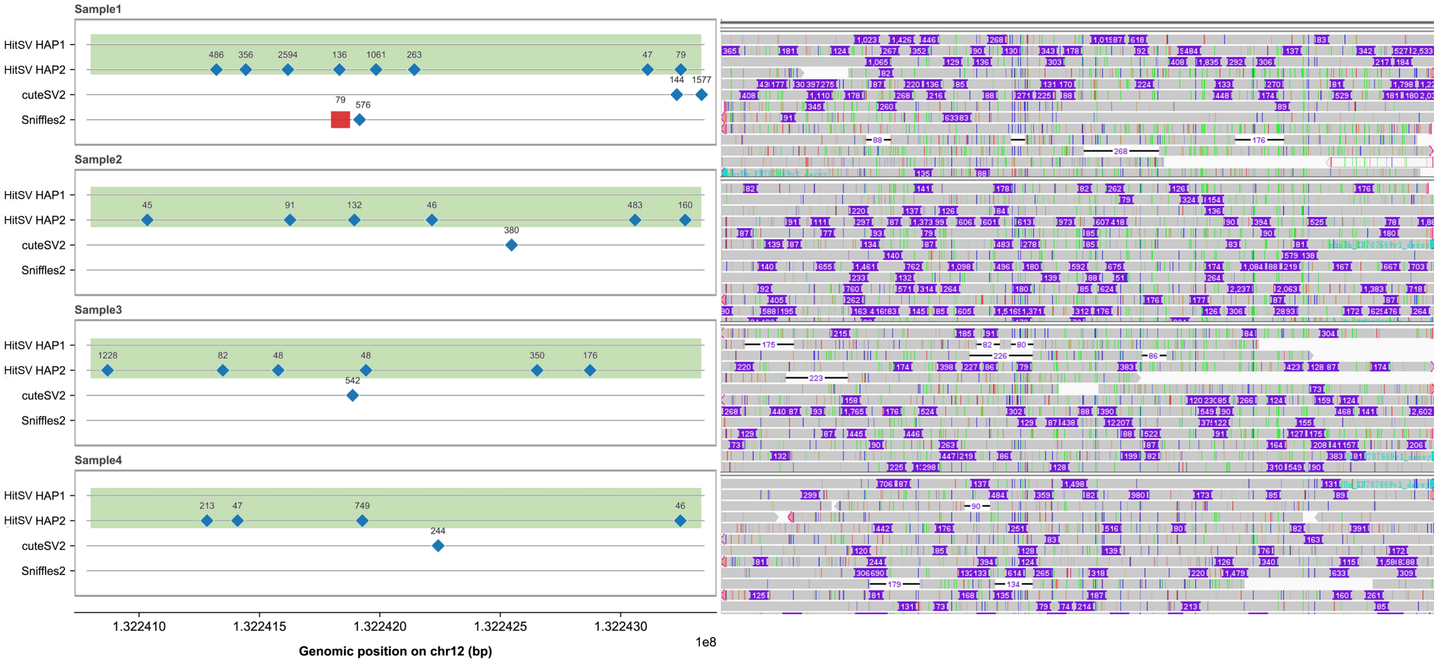
The analysis focuses on a local genomic region encompassing the PTPRN2 locus on chromosome 7 (chr7:158423050-158430593 bp), illustrating CSV inheritance patterns across related individuals. The left panel summarizes the SVs detected by different tools across haplotypes and samples. HitSV produces haplotype-resolved calls and consistently recovers SVs that are concordant with the ground truth and shared across family members, enabling reliable tracing of inherited CSVs. In contrast, other tools lack haplotype resolution and often report inconsistent or missing calls. The right panel presents IGV snapshots of the same region for HG002, HG003, and HG004, showing abnormal read alignments signatures including large soft clipping, and split.



Supplementary Figure 23. An example of CSV detection in a family-based dataset.

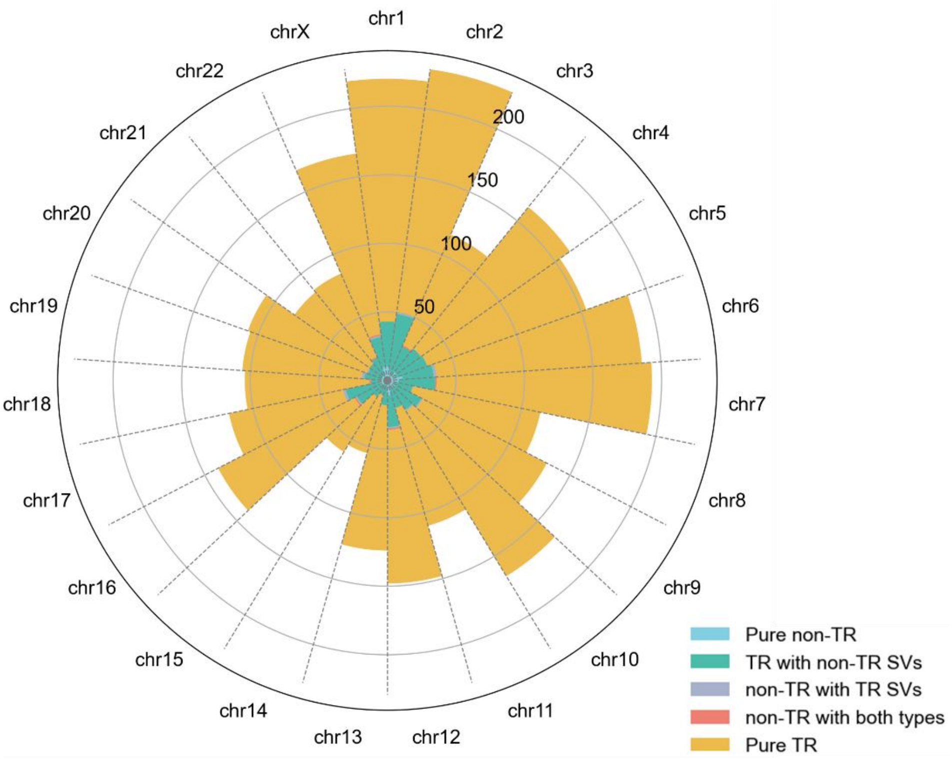
The analysis focuses on a local genomic region encompassing the PTPRN2 locus on chromosome 7 (chr7:158423050-158430593 bp), illustrating CSV inheritance patterns across related individuals. The left panel summarizes the SVs detected by different tools across haplotypes and samples. HitSV produces haplotype-resolved calls and consistently recovers SVs that are concordant with the ground truth and shared across family members, while other methods fail to detect these events. The right panel presents IGV snapshots of the same region for HG002, HG003, and HG004, showing abnormal read alignments signatures including large soft clipping, and split.

`



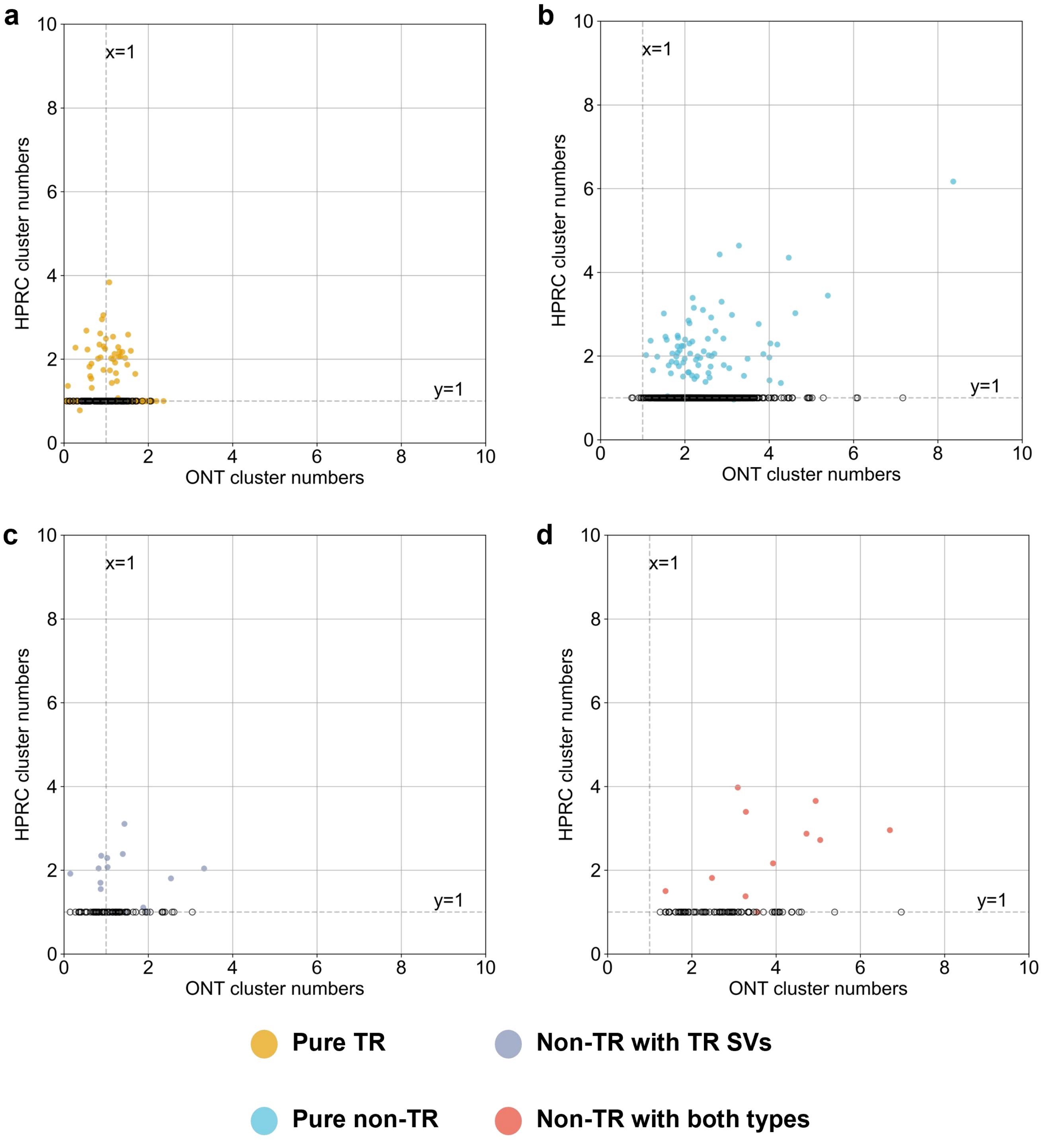
Supplementary Figure 24. Detection of complex structural variants across multiple ONT samples.

Another example focuses on a common CSV region on human chromosome 12 (chr12:132240782-132243348 bp), illustrating CSV detection results across five samples from the 557-ONT dataset. The left panel summarizes the structural variants detected by different tools across samples and haplotypes. Haplotype-resolved calls produced by HitSV enable explicit assignment of CSVs to individual haplotypes, facilitating the tracing and comparison of complex variant patterns across samples. In contrast, other tools lack haplotype resolution and report aggregated SV calls. The right panel presents IGV snapshots of the same region for multiple samples, showing read alignments and supporting signals consistent with TR-MEI–associated CSV events.



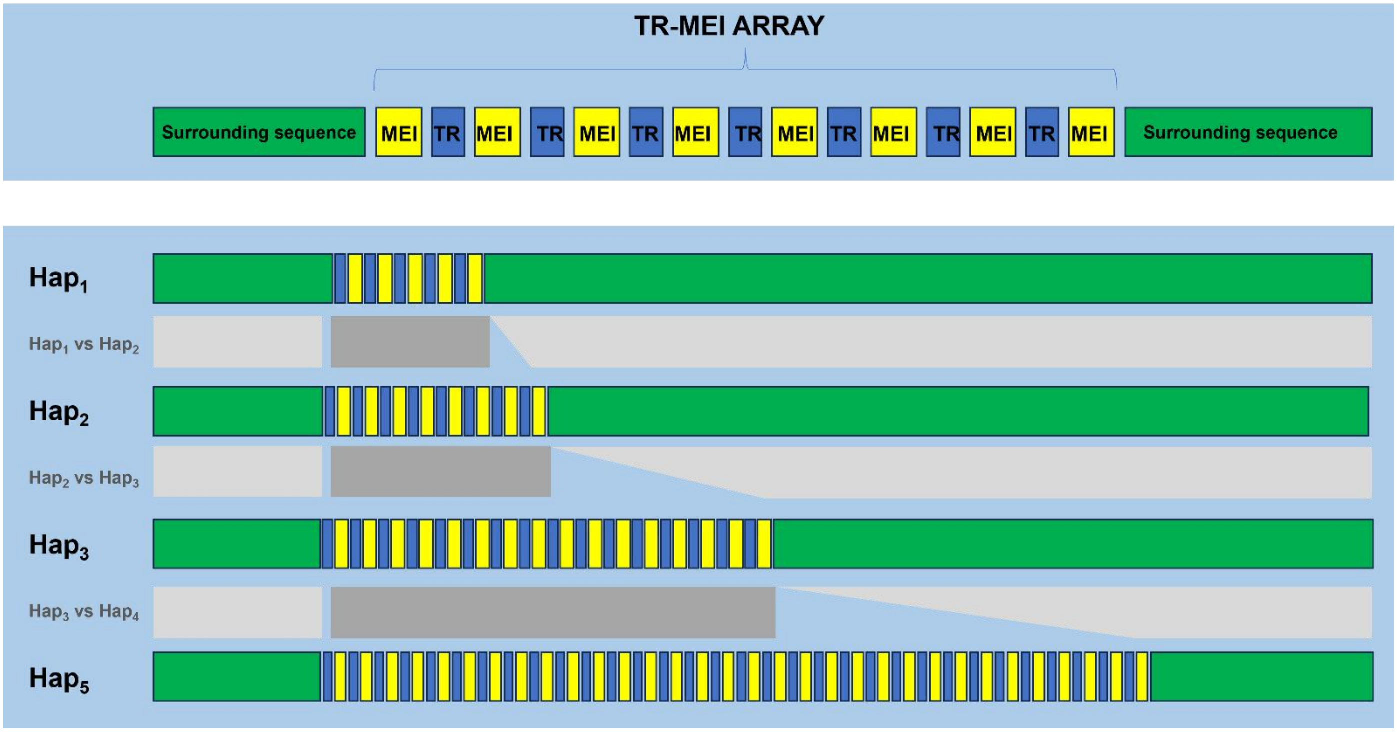
Supplementary Figure 25. Chromosome-wide distribution and classification of CSV regions in HPRC datasets

Chromosome-level classification of CSV regions across 47 Human Pangenome Reference Consortium (HPRC) datasets. Each sector corresponds to one chromosome, and radial bars indicate the number of CSV regions assigned to different structural categories.



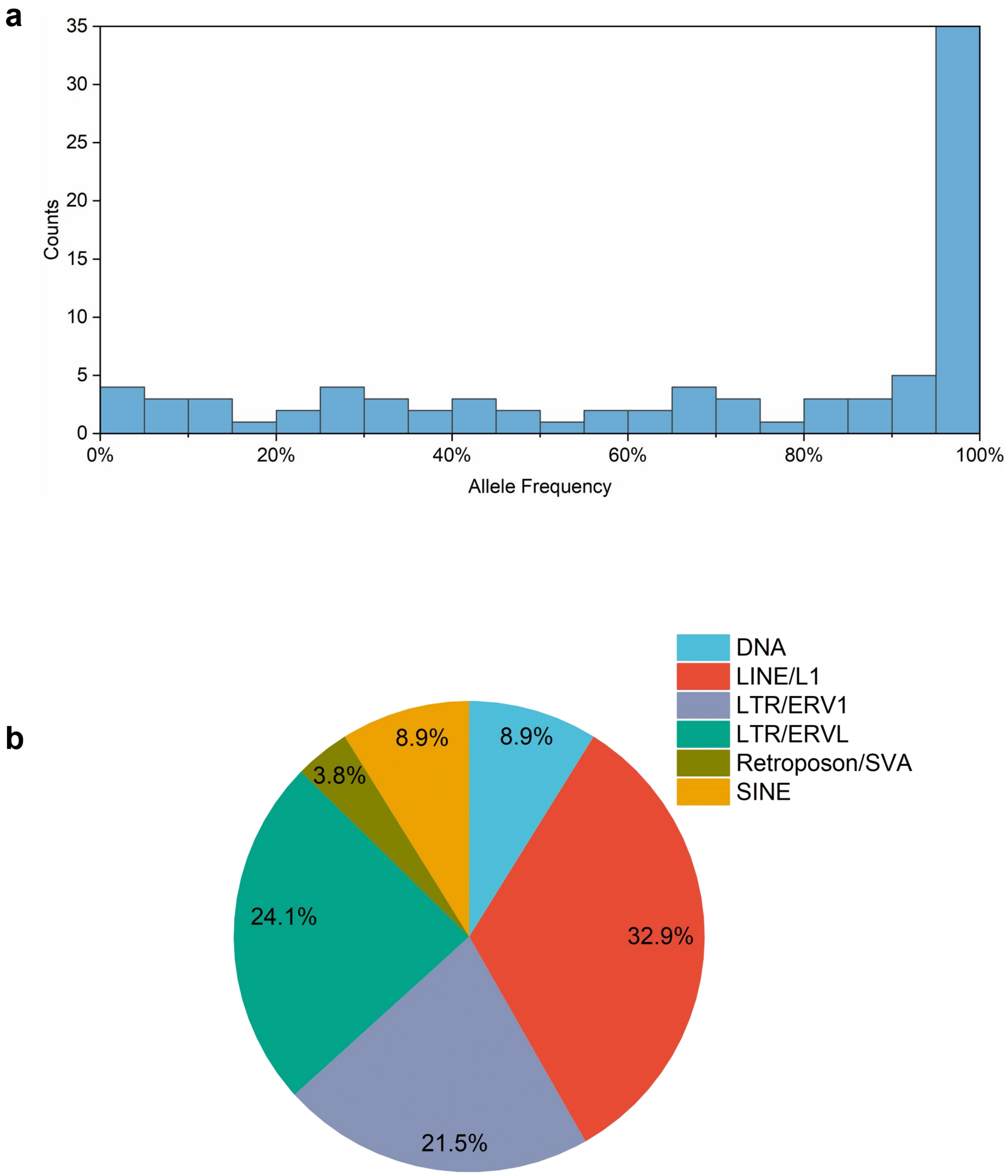
Supplementary Figure 26. Comparison of clustering complexity for CSV regions in the ONT-HPRC datasets.

Scatter plots compare the number of clusters derived from Oxford Nanopore Technologies (ONT) data (x-axis) with the number of clusters inferred from HPRC data (y-axis) across different classes of CSV regions. Each point represents one CSV region. Dashed lines indicate the reference threshold of a single cluster (x = 1 or y = 1).
a, Pure TR regions; **b,** Pure non-TR regions; **c,** Non-TR with TR SVs regions; **d,** Non-TR with both types regions.



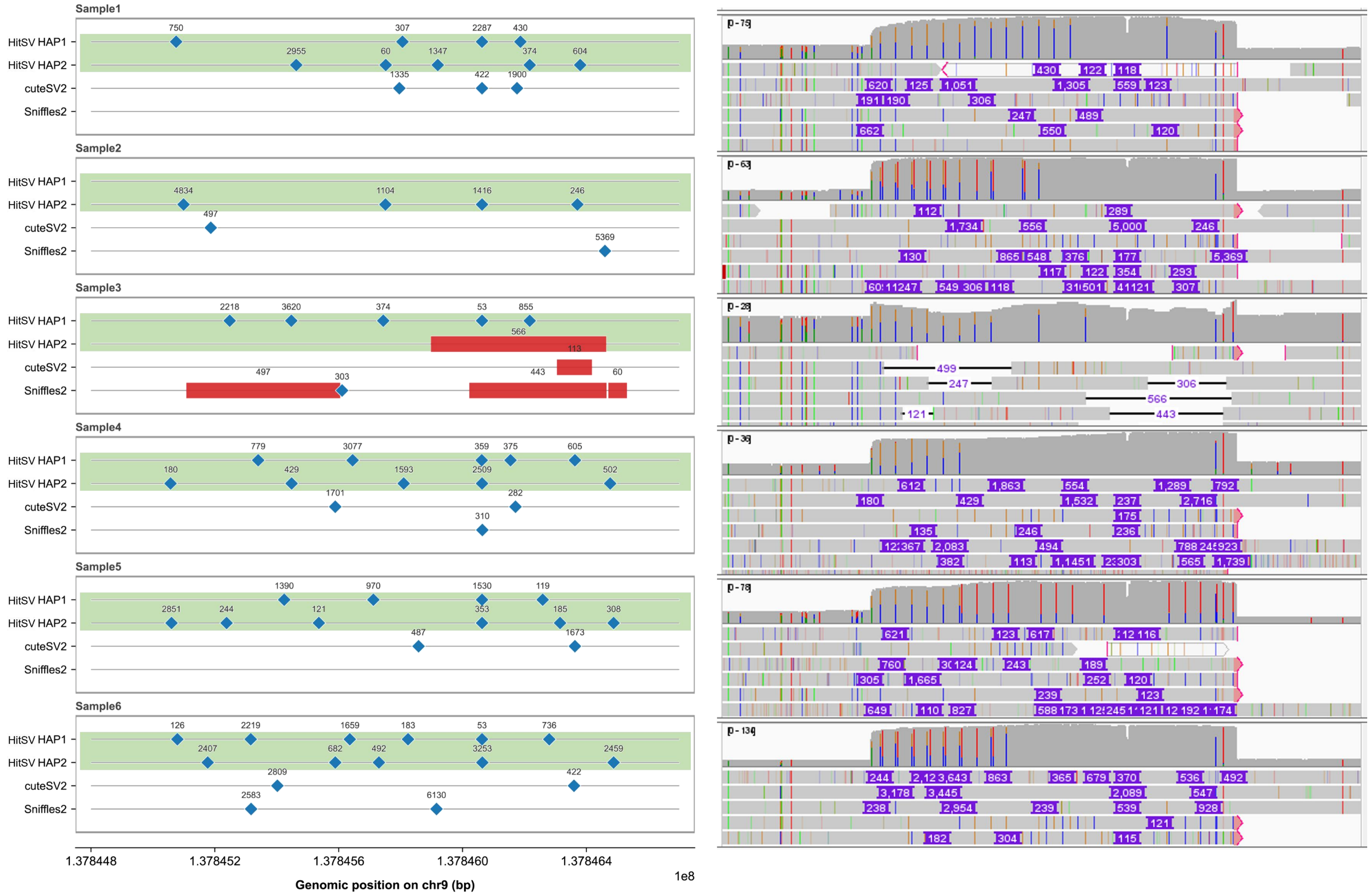
Supplementary Figure 27. Schematic illustration of TR-MEI array formation.

A conceptual diagram illustrating the structure of a mobile element insertion–tandem repeat (TR-MEI) array and its variation across haplotypes. Yellow rectangles represent MEI sequences, blue rectangles denote tandem repeat (TR) units, and green blocks indicate flanking non-repetitive genomic sequences surrounding the TR-MEI region. The upper panel depicts the general architecture of a TR-MEI array composed of alternating MEI and TR units. The lower panels show four haplotypes (Hap1–Hap4) with increasing copy numbers of TR-MEI units. Dark gray trapezoids highlight the genomic region undergoing MEI amplification, illustrating how differential expansion of TR-MEI arrays across haplotypes gives rise to complex structural variation.



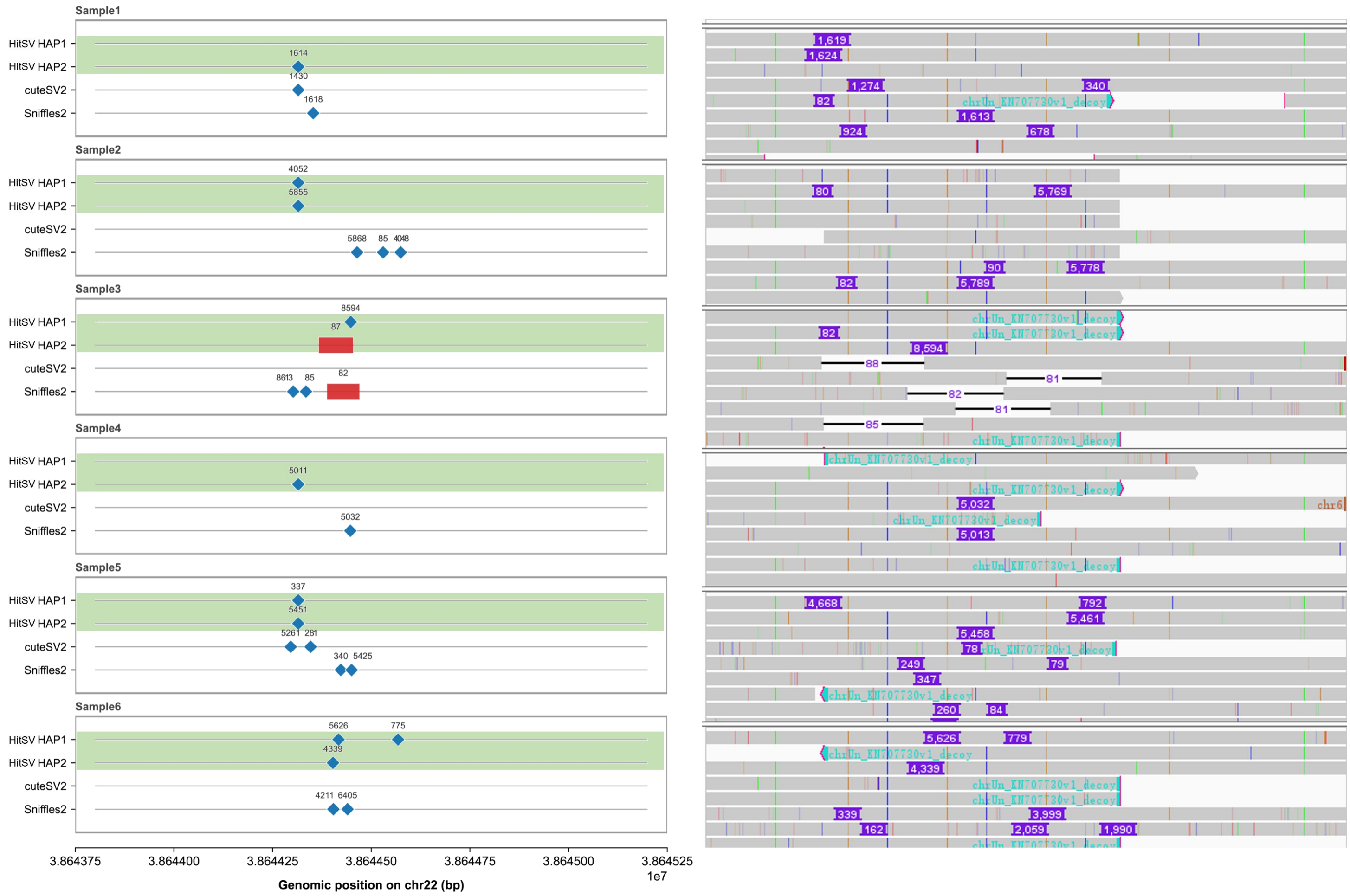
Supplementary Figure 28. Allele frequency distribution and MEI composition of TR-MEI regions.

a, Histogram showing the allele frequency (AF) distribution of samples carrying TR-MEI events across different TR-MEI regions. AF is calculated as the proportion of haplotypes harboring a given TR-MEI event, illustrating the spectrum from rare to near-fixed variants. **b,** Pie chart summarizing the composition of MEI types within TR-MEI intervals, including LINE/L1, SINE/Alu, SINE/MIR, DNA transposons, and Retroposon/SVA elements. Percentages indicate the relative contribution of each MEI class to the total set of TR-MEI regions.



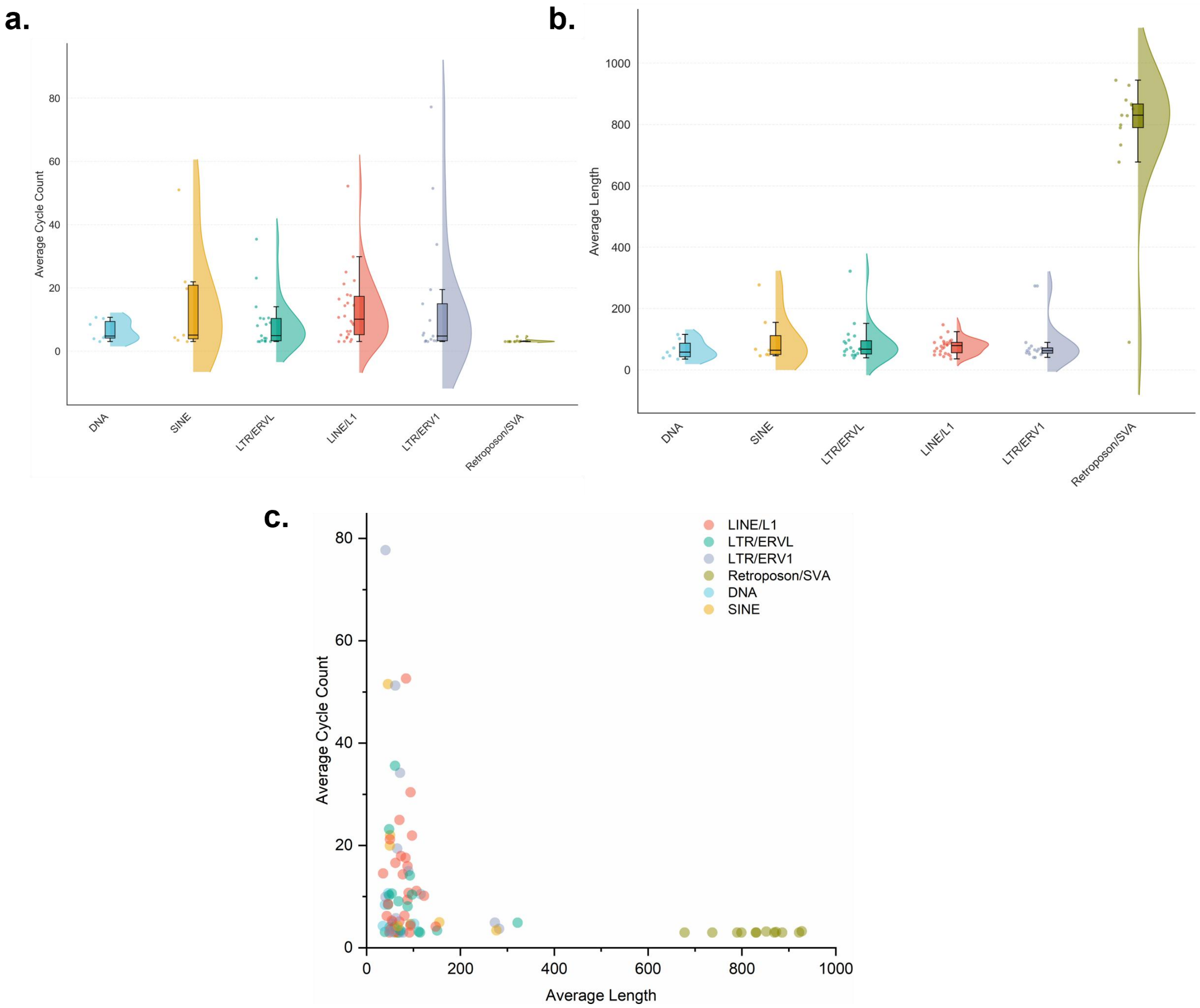
Supplementary Figure 29. Detection of TR-MEI–associated complex structural variants across multiple ONT samples.

Another example focuses on a TR-MEI–related region on human chromosome 9 (chr9:137,844,500-137,847,000 bp), illustrating CSV detection results across six samples from the 557-ONT dataset. The left panel summarizes the structural variants detected by different tools across samples and haplotypes. Haplotype-resolved calls produced by HitSV enable explicit assignment of CSVs to individual haplotypes, facilitating the tracing and comparison of complex variant patterns across samples. In contrast, other tools lack haplotype resolution and report aggregated SV calls. The right panel presents IGV snapshots of the same region for multiple samples, showing read alignments and supporting signals consistent with TR-MEI–associated CSV events.



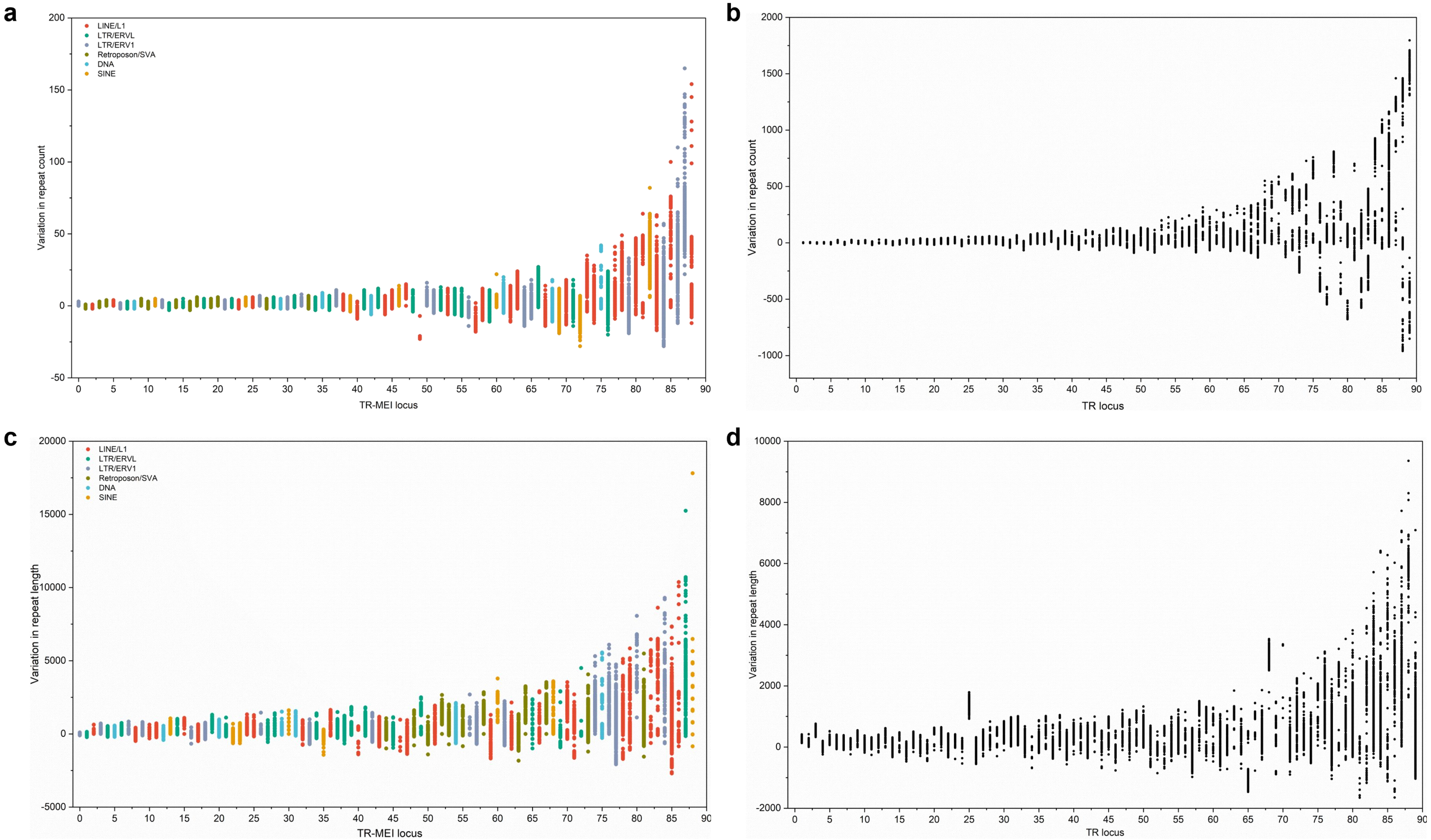
Supplementary Figure 30. Detection of TR-MEI–associated complex structural variants across multiple ONT samples.

Another example focuses on a TR-MEI–related region on human chromosome 22 (chr22:38,643,500–38,645,500 bp), illustrating CSV detection results across six samples from the 557-ONT dataset. The left panel summarizes the structural variants detected by different tools across samples and haplotypes. Haplotype-resolved calls produced by HitSV enable explicit assignment of CSVs to individual haplotypes, facilitating the tracing and comparison of complex variant patterns across samples. In contrast, other tools lack haplotype resolution and report aggregated SV calls. The right panel presents IGV snapshots of the same region for multiple samples, showing read alignments and supporting signals consistent with TR-MEI–associated CSV events.



Supplementary Figure 31. Structural characteristics of TR-MEI arrays across MEI classes.

a, Violin plots showing the distribution of the average number of repeat units per TR-MEI region, stratified by MEI class. Each point represents one TR-MEI region, and boxplots indicate the median and interquartile range.
b, Violin plots of the average MEI length within TR-MEI regions for different MEI classes, illustrating substantial variation in MEI size across element types.
c, Scatter plot depicting the relationship between the average number of repeat units and the average MEI length in TR-MEI regions. Each point corresponds to one TR-MEI region and is colored by MEI class.



Supplementary Figure 32. Repeat unit count and length variability of tandem repeat (TR) regions.

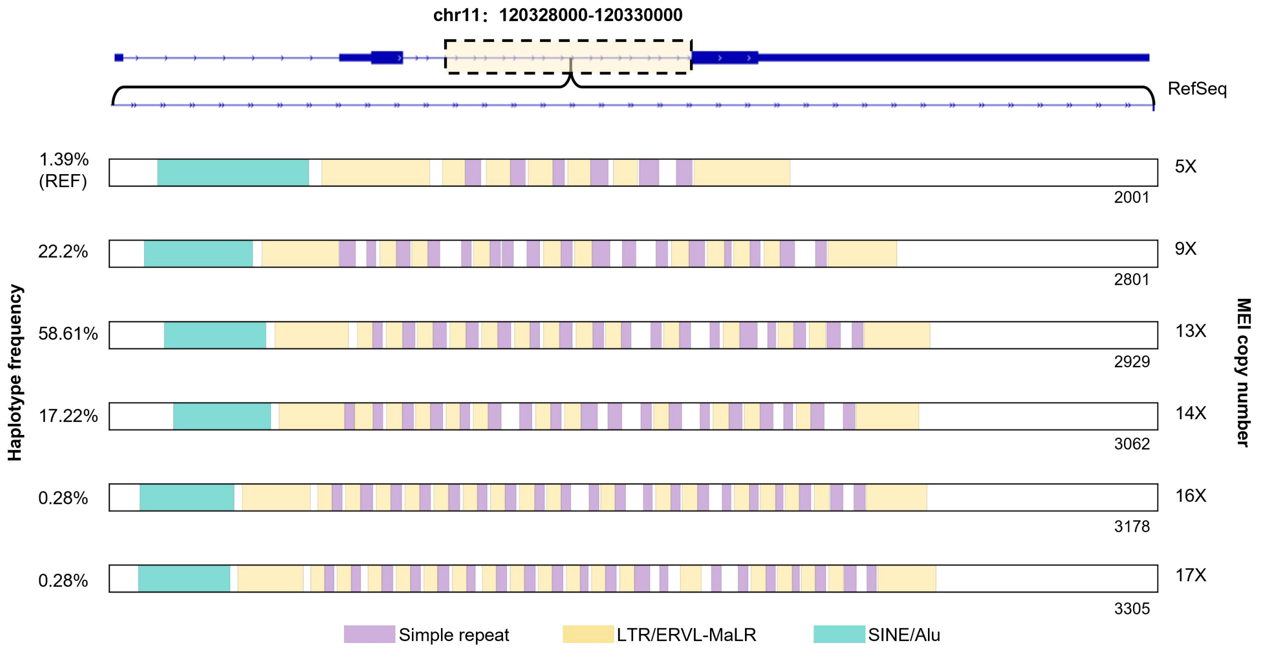
The x-axis denotes individual TR loci, and the y-axis indicates variability relative to the reference genome sequence.

a, Variability in the number of repeat units within TR-MEI arrays. Each point represents one TR-MEI interval, with colors indicating the predominant MEI type associated with the interval.

b, Variability in the number of repeat units across regular TR regions lacking MEI insertions.

c, Length variability of TR-MEI arrays.

d, Length variability of regular TR regions without MEI insertions.



Supplementary Figure 33. Haplotype-resolved characterization of a TR-MEI array at the *TLCD5* locus.

The analysis focuses on a TR-MEI–associated region spanning the TLCD5 locus on human chromosome 11 (chr11:120,328,000–120,330,000 bp), illustrating structural variation across samples from the 557-ONT dataset. The upper panel shows the genomic context of the locus, highlighting the position of the TR-MEI array within an intronic region of TLCD5. The lower panel displays RepeatMasker annotations of representative locally assembled contigs from the reference genome and multiple samples, visualized as colored rectangular bars (contig lengths are indicated beneath each bar). Numbers shown on the left of each contig denote the allele frequency of the corresponding assembly across the full 557-ONT cohort, whereas numbers on the right indicate the repeat counts of MEI arrays within each contig. The TR-MEI arrays at this locus are primarily composed of LTR/ERVL elements interspersed with simple repeat sequences, illustrating haplotype-specific expansion and contraction patterns of TR-MEI structures.

Supplementary Notes 1

Description about the running time and memory footprint.

HitSV showed affordable runtime for various data configurations (30x long reads: 0.83 CPU hours, 4×+30×/60× hybrid sequencing data: 3.35 and 7.94 CPU hours, 30×/60× short reads: 3.14 and 7.39 CPU hours, respectively) as well as relatively small memory footprints (30x long reads: 1.23 GB, 4×+30×/60× hybrid sequencing data: 4.76 and 8.39 GB, 30x/60x short reads: 4.24 and 8.29 GB, respectively).

SV genotyping and quality control

HitSV calculates the posterior probability $P(G|D)$ of a specific genotype G given an observed set of related alignment D for genotyping. Assuming the list of the candidate alleles as $A = \{r, x_1, ..., x_n\}$, where r refers to reference allele and $x_i, i = 1, ..., n$ refers to SV-alleles, there are $|A|^2$ types of candidate genotypes, i.e., $G = \{rr, rx_1, rx_2, rx_3, ..., x_{n-1}x_n\}$ and $P(G|D)$ is formed as following:

$$P(G|D) = \frac{P(D|G) \times P(G)}{P(D)} \tag{1}$$

Further, the equation is reduced as $P(D|G) \times P(G)$ since D has been observed. The prior $P(G)$ is set as:

$$P(G) = \begin{cases} \theta/(n^2), \ \mathrm{G = pq} \\ 0.5\theta/n, \ \mathrm{G = rp} \\ 1 - 1.5\theta, \ \mathrm{G = rr} \end{cases} \tag{2}$$

where θ is a constant (default value: 10^{-5}) and n is the number of non-reference alleles, $\mathrm{p, q} \in A$.

Further, each alignment is considered as an independent event and the conditional probability $P(D|G)$ is written as the product of the probabilities of the events:

$$P(D|G) = \prod\nolimits_{d \in D} P(d|G) \tag{3}$$

where $P(d|G)$ is the probability of observing a specific abnormal alignment d given the genotype G,

$$P(d|G) = \sum\nolimits_{a \in A} P(d|a) \times P(a|G) \tag{4}$$

Similarly, $P(d|a)$ represents the probability of observing a specific alignment d given a particular haplotype a. $P(a|G)$ denotes the probability of observing a specific haplotype a under the condition of a given genotype G , which is taken from $\{0,0.5,1.0\}$. All $P(d|a)$values form a support matrix, an $M \times N$matrix, where M is the number of reads and N is the number of contigs (two under a diploid assumption).

After realigned reads to different candidate haplotypes, HitSV recalculates the alignment scores by setting the penalty scores as 0, 3, 4, 1 for match, mismatch, gap open and gap extension, respectively. The penalty score has an upper limit (default is 5) and the total penalty score for realignment is denoted as S, $P(d|a)$ is set as 10^{-S} consequently.

For long-read data, a binary support matrix is generated by determining whether each long read supports a given contig in the pair (1 indicating support and 0 indicating non-support).

The genotype G maximizing $P(G|D)$ is selected and the quality score is computed by the following equation,

$$QUAL(G) =- 10 \times log_{10}\left(1 - P(G|D)\right) \tag{5}$$

The results corresponding to homozygous reference genotypes are directly discarded. Additionally, results with quality scores below the threshold (default is 20) are labeled as *LOW*, while others are labeled as *PASS.*

Supplementary Notes 2

Command lines.

1. The commands and parameters used for read alignment

BWA-MEM

```
bwa mem -t 10 -K 100000000 -Y ref.fa read1.fq read2.fq > sample.unsorted.sam
```

pbmm2

```
pbmm2 align ref.fa samle.fastq.gz sample.sort.bam -j 10 --sort --rg '@RG\tID:ID_002\tSM:HG002'
```

minimap2

```
minimap2 -x asm5 -t 10 -a ref.fa samle.fastq.gz > sample.unsorted.sam
```

samtools

```
cat sample.unsorted.sam | samtools view -@ 10 -b -o samle.unsorted.bam
samtools sort -@ 10 -m 2000000000 --output-fmt=BAM -o sample.bam sample.unsorted.bam
samtools index sample.bam
```

2. Down sample with samtools.

samtools

```
samtools view -s 0.25/0.5 -b -o output.bam input.bam
```

3. The commands and parameters used for LRS SV calling

HitSV

```
HitSV call -l input.bam -r ref.fa -o output.vcf 2> /dev/null
```

sniffle2 default mode

```
sniffles --input input.bam --vcf output.vcf --reference ref.fa
```

SVDSS2

```
SVDSS_linux_x86-64 index --reference ref.fa --index ref.fmd
SVDSS_linux_x86-64 smooth --reference ref.fa --bam input.bam --threads 10 > SMO.bam
samtools index SMO.bam
SVDSS_linux_x86-64 search --index ref.fmd --bam SMO.bam > SPE.txt
SVDSS_linux_x86-64 call --reference ref.fa --bam SMO.bam --sfs SPE.txt --threads 10 > output.vcf
```

cuteSV2

```
cuteSV --genotype input.bam ref.fa output.vcf work_dir
```

SVision-PRO

```
SVision-pro --detect_mode germline --process_num 10 --target_path sample.bam --genome_path ref.fa --model_path model_liteunet_256_8_16_32_32_32.pth --out_path workdir --sample_name SAMPLE_NAME
```

Sawfish

```
sawfish discover --ref ref.fa --bam sample.bam --output-dir work_dir --threads 10
sawfish joint-call --sample work_dir
```

4. The commands and parameters used for SRS SV calling

HitSV ref stat ***[only run once for all samples]***

```
HitSV srs_fa_stat ref.fa > ref.stat.txt
```

HitSV

```
HitSV srs_trans_reads ref.fa sample.bam TL.bam
samtools sort -@ 10 --output-fmt=BAM -o TL.sort.bam TL.bam
samtools index TL.sort.bam
HitSV call -n input.bam -L TL.sort.bam -r ref.fa -I ref.stat.txt -o output.vcf 2> /dev/null
```

manta

```
configManta.py --bam input.bam --referenceFasta ref.fa --runDir manta_work_dir
python manta_work/runWorkflow.py -m local -j 10
```

delly

```
delly call -g ref.fa input.bam > output.vcf
```

lumpy

```
samtools view -F 1294 input.bam > DIS.bam
samtools sort -@ 16 DIS.bam -o DIS.sort.bam && samtools index DIS.sort.bam
samtools view -h input.bam | ./lumpy-sv/scripts/extractSplitReads_BwaMem -i stdin | samtools view -Sb - > SIP.bam
samtools sort -@ 16 SIP.bam -o SIP.sort.bam && samtools index SIP.sort.bam
lumpyexpress -B input.bam -S SIP.sort.bam -D DIS.sort.bam -o output.vcf
```

5. The commands and parameters used for Hybrid SV calling

HitSV-9mer_profile [only run once for all samples]

```
python error_profile_analysis.py HG002.ONT.sort.bam ref.fa HG002_GRCh38_1_22_v4.2.1_benchmark.vcf.gz 9mer_profile
```

HitSV

```
HitSV srs_trans_reads ref.fa NGS.input.bam TL.bam
samtools sort --output-fmt=BAM -o TL.sort.bam TL.bam
samtools index TL.sort.bam
HitSV call -l lrs.input.bam -n srs.input.bam -L TL.sort.bam -r ref.fa -I ref.stat.txt -k 9mer_profile -o output.vcf 2> /dev/null
```

6 The commands and parameters used for benchmarking

bcftools & bgzip

```
cat tools.vcf | grep -v -E "SVTYPE=BND” | bcftools sort | bgzip -cf > output.vcf.gz
tabix output.vcf.gz
```

truvari bench

```
truvari bench --passonly -p 0 -P 0.7/0.9/1--dup-to-in -c cmp.vcf.gz -b base.vcf.gz -o ./result.dir --reference ref.fa --includebed region.bed
```

truvari refine

```
truvari bench -p 0.7/0.9/1 -P 0.7/0.9/1 --passonly --pick ac --dup-to-ins --reference ref.fa --includebed region.bed -c cmp.vcf.gz -b base.vcf.gz -o ./result.dir
truvari refine --write-phab --use-region-coords --use-original-vcfs --align mafft --reference ref.fa --regions ./result.dir/candidate.refine.bed refine_output_dir
truvari ga4gh --write-phab --input refine_output_dir --output combined_result
```

7 Mendelian consistency evaluation.

Run variant detection for different algorithms on different samples (HG002/3/4 and HG005/6/7), then store the VCF file paths produced by the same algorithm in the file "SUR_l.txt", and execute:

```
SURVIVOR merge SUR_l.txt 10 1 1 1 0 50 merge.vcf
sed -i "s/\.\/\./0\/0/g" merge.vcf
cat merge.vcf | grep -v "SVTYPE=BND" | grep -v "SVTYPE=TRA" | grep -v "SVTYPE=INV" | grep -v "HLA" | bcftools sort | bgzip -cf > merge.vcf.gz
tabix merge.vcf.gz
bcftools +mendelian2 merge.vcf.gz -p 1X:SAMPLE_2,SAMPLE_1,SAMPLE -m c
```

8 The commands and parameters used for 557 ont samples analysis

HitSV single sample call

```
HitSV call -p ERR_PRONE -l ont.sample.bam -r ref -o sample.vcf
```

Single-sample csv region analysis

```
HitSV tools csv_region_500w sample.vcf 500 50 > sample.csv.bed
cat sample.csv.bed | grep "HAP1" | grep "500W" | sort | uniq > sample.csv.hap1.bed
cat sample.csv.bed | grep "HAP2" | grep "500W" | sort | uniq > sample.csv.hap2.bed
```

Merge csv region of all samples and filtering

```
cat */*hap*bed > ALL_single_hap.bed
cat ALL_single_hap.bed | awk '{printf "%s:%s-%s\n",$1,$2,$3;}' | sort | uniq -c > ALL_single_hap_summary.txt
cat ALL_single_hap_summary.txt| awk '{split($2, A, ":"); split(A[2], B, "-"); if($1 >=56) printf "%s\t%s\t%s\t%s\n", A[1],B[1],B[2],$1}' > AF_0.05_CSV_AF.bed
bedtools sort -i AF_0.05_CSV_AF.bed > AF_0.05_CSV_AF.sort.bed
bedtools merge -i AF_0.05_CSV_AF.sort.bed > cohot.csv.bed
```

Recall local contig sequences for each sample and csv region

```
HitSV call -p ERR_PRONE -b cohot.csv.bed -B PURE_STR -l ont.sample.bam -r ref.fa -o sample.FC.txt
```

Store contigs in corresponding csv regions

```
find *.FC.txt > all_sample.FC.file_name.txt
HitSVL tools contig_file_split all_sample.FC.file_name.txt csv_result_dir GRCh38.fa
```

Annotation and contig clustering analysis in single csv region

For each csv region in "*csv_result_dir* ":

```
RepeatMasker contig_in_region.fa
trf contig_in_region.fa 2 7 7 80 10 10 500 -h -d -ngs 1> contig_in_region.fa.trf.out
HitSV tools fc_joint_ana contig_in_region.fa contig_in_region.fa.out contig_in_region.fa.trf.out 0
```

9 The commands and parameters used for 1KGP re-analysis

HitSV single sample SV calling

For each sample in 1KGP P4:

```
HitSV srs_trans_reads ref.fa sample.bam TL.bam
samtools sort --output-fmt=BAM -o TL.sort.bam TL.bam
samtools index TL.sort.bam
HitSV call -n input.bam -L TL.sort.bam -r ref.fa -I ref.stat.txt -o sample.single.vcf
```

SURVIVOR (SV merging)

```
find *single.vcf > vcf_list.txt
SURVIVOR merge vcf_list.txt 500 1 1 1 0 30 output.vcf
```

AnnotSV

```
AnnotSV -SVinputFile inputFile -outputDir output -outputFile sample_name.csv -genomeBuild GRCh38 -SVminSize 30
```