nature genetics

Article

https://doi.org/10.1038/s41588-026-02538-6

Population-level structural variant characterization using pangenome graphs

Received: 17 June 2025

Accepted: 10 February 2026

Published online: 10 March 2026

Check for updates

Songbo Wang[1,2], Tun Xu[1,2], Pengyu Zhang[1,2] & Kai Ye[1,2,3,4,5]

Population-level structural variant (SV) profiling is crucial in the era of pangenomes. However, identifying SVs from genome assemblies and pangenome graphs remains a substantial challenge. Here we present Swave, a sequence-to-image, deep learning-based method that accurately resolves both simple and complex SVs, along with their population characteristics, from assembly-derived pangenome graphs. Swave introduces 'projection waves' to summarize the dotplot images that capture mapping patterns between reference and SV-indicating alleles in the pangenome. Then, a recurrent neural network distinguishes true SV signals from background noise introduced by genomic repeats. Swave demonstrates superior performance in both SV-type classification and genotyping compared with existing methods. When applied to healthy cohorts and rare-disease cohorts, Swave reveals complex and polymorphic SV patterns across human populations and identifies potentially pathogenic SVs. These advancements will facilitate the creation of comprehensive population-level SV catalogs, deepening our understanding of SVs in genetic diversity and disease associations.

Structural variants (SVs) are genomic alterations larger than 50 base pairs (bp), categorized into simple SVs[1] (SSVs; for example, insertions, deletions, inversions and duplications) and complex SVs (CSVs) with more than one internal breakpoint or subcomponent[1,2]. Recent advances in whole genome sequencing have underscored the critical roles of SV in development[3], genetic disorders[4] and cancers[5]. The emergence of long-read sequencing (LRS) technologies has markedly improved SV discovery[1], increasing the number of detectable SVs 2–4.25-fold[6], and enhanced resolution of structural complexity, particularly for CSVs. Several LRS-based SV callers have been developed[2,7–11] using either model-matching or deep learning strategies[12]. Concurrently, improvements in genome assembly methods have resulted in high-quality genome assemblies that offer longer, more precise sequences, outperforming read-based SV detection[13,14]. However, existing assembly-based SV callers remain limited by model-matching approaches[15,16], which restrict detection to known SV types and often overlook uncharacterized or CSVs.

With the declining cost of genome sequencing and computing, population-level SV analysis using assemblies is now feasible[17–19]. Examining SVs across large cohorts enables comprehensive characterization of SV landscapes and population-specific features, offering insights into human evolution and clinical interpretation[20,21]. Several gene loci (for example, *AMY1*[22,23] and *MUC5B*[24]) have been examined at population level, revealing contributions of SVs to natural selection and adaptation, highlighting the need for automated, genome-wide screening of evolution-associated SV loci. Clinically, distinguishing pathogenic SVs from benign ones observed in health populations requires a robust SV reference derived from population-scale datasets.

A critical component of population-level SV analysis is cross-sample merging, which integrates individual SV callsets into population profiles reflecting SV genotypes and frequencies[25–27]. However, current merging techniques, primarily based on SV region overlap and sequence similarity, are prone to false positives and missing genotypes[2,28]. They struggle in genome repetitive regions,

[1]School of Automation Science and Engineering, Faculty of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an, China. [2]MOE Key Laboratory for Intelligent Networks and Networks Security, Faculty of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an, China. [3]Center for Mathematical Medical, The First Affiliated Hospital of Xi'an Jiaotong University, Xi'an, China. [4]Genome Institute, The First Affiliated Hospital of Xi'an Jiaotong University, Xi'an, China. [5]Faculty of Science, Leiden University, Leiden, the Netherlands. e-mail: kaiye@xjtu.edu.cn

https://doi.org/10.1038/s41588-026-02538-6

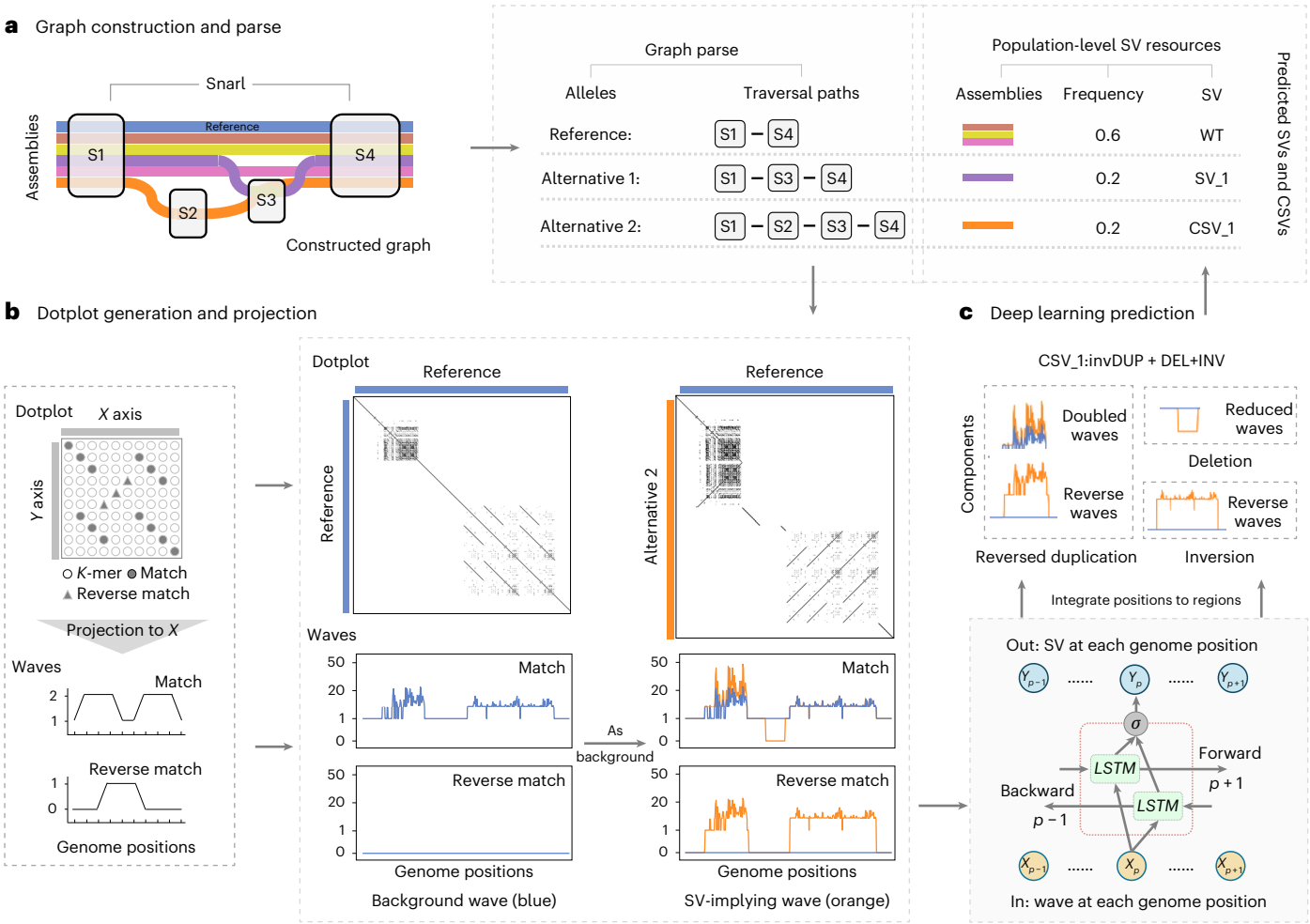


Fig. 1 | Schematic overview of Swave. a, Pangenome graph construction and allele extraction. Swave extracts allele paths for both reference and individual assemblies from a pangenome graph. **b**, Sequence-to-image transformation. Dotplot images depicting structural differences between the REF sequence and ALT allele sequences are projected into waves to extract both background and SV-indicating signals. **c**, Deep learning-based SV classification. A RNN takes in the wave signals and assigns SV types based on learned sequence-context patterns.

where mapping ambiguities cause variable SV lengths, and in CSV regions, where complex structures with nested breakpoints are often misclassified or missed entirely.

A pangenome graph, which compactly represents multiple genomes along with their similarities and differences, is well suited to population-scale SV applications[17–19,29,30]. Through careful alignments[31–34], graph snarls encode SV alleles within nodes and paths[18,35,36]. For any given assembly, pangenome graph construction tools annotate its specific path and SV allele within each snarl, facilitating SV genotyping. Still, classifying SV alleles in the pangenome graph remains a bottleneck. Existing LRS- or assembly-based SV calling methods, whether model based or deep learning based, are not designed for pangenome graphs. Instead, current tools rely on length differences between reference (REF) and alternative (ALT) alleles to identify deletions and insertions[29,30,37], which is insufficient for capturing the full landscape of SV diversity in populations.

Here, to address these limitations, we propose Swave, a method that leverages assemblies and pangenome graphs to enable SV discovery from individual genomes to population-wide datasets. Swave parses SV-indicating alleles from pangenome graphs and realign them against REF alleles to generate base-level dotplot images. These images are then transformed into projection waves that summarize alignment conditions at each genomic location. By comparing against the waves generated from reference genome backgrounds, Swave reduces the disruptions from genome repetitive sequence. A recurrent neural network (RNN) is then applied to classify SV types. Benchmarking against state-of-the-art methods, Swave exhibits superior accuracy in both SV classification and genotyping using assembly-derived pangenomes. Applied to healthy cohorts (334 haplotypes), Swave reveals the population-level complexity and polymorphism of challenging SSVs (inversions) and previously underestimated CSVs, especially rare ones with population allele frequency (AF) below 1%. In a rare disease cohort comprising 287 proband genomes (574 haplotypes), Swave identifies potentially pathogenic SVs characterized by singleton and exon-disrupting events, including 888 SSVs and 34 CSVs. These findings broaden the spectrum of pathogenic variants from small-scale mutations (for example, single nucleotide polymorphisms) to SVs that can induce more extensive genomic damage.

Results

Algorithm overview

Swave comprises three key modules:

(1) Graph module for SV allele deconstruction (Fig. 1a and Extended Data Fig. 1; Methods). Using both reference genome and sample assemblies, Swave constructs a pangenome graph using Minigraph's incremental graph generation[31]. Within these graphs, graph snarls (graph substructures indicating local variation) are considered as candidate SV loci. Paths traversing distinct

https://doi.org/10.1038/s41588-026-02538-6

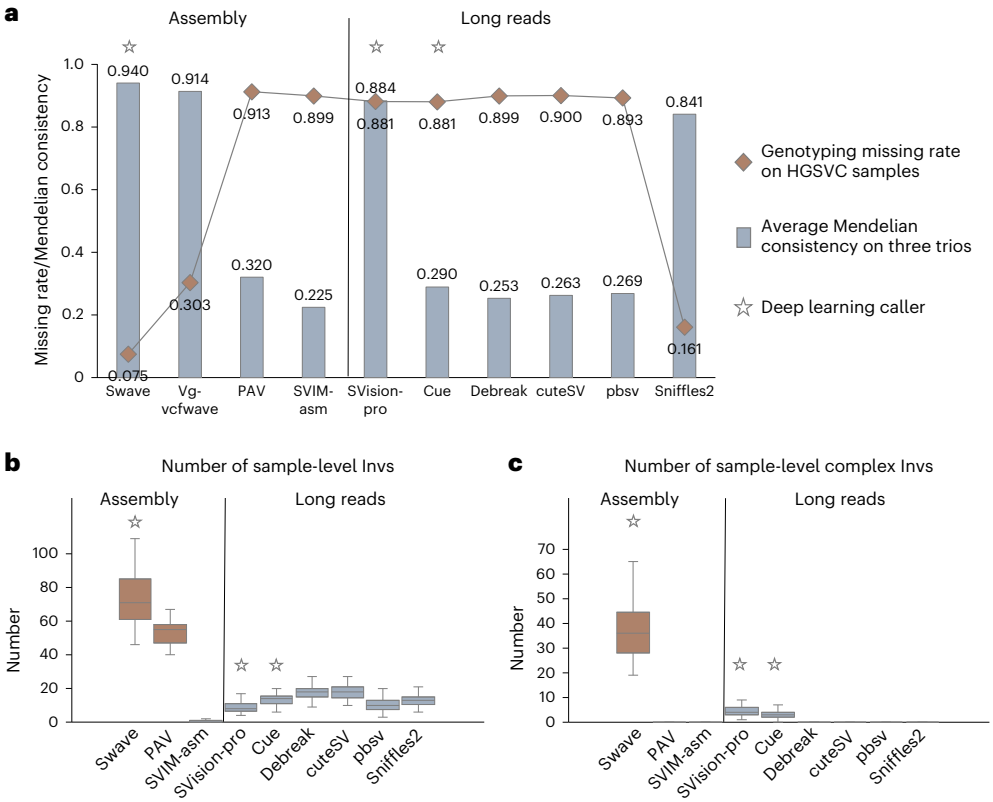


Fig. 2 | Benchmarking the performance of Swave and comparative methods. **a**, Two key metrics, including Mendelian consistency and genotyping missing rates were evaluated among Swave, assembly- and LRS-based callers. Mendelian consistency was averaged across three parent–child trios (CHS, PUR and YRI). Genotyping missing rates were calculated using the HGSVC samples. **b**, Comparison of the detected inversion numbers among HGSVC 65 samples. **c**, Comparison of the detected complex inversion numbers among HGSVC 65 samples. For **b** and **c**, the box plot defines the median (Q2, 50th percentile), first quartile (Q1, 25th percentile) and third quartile (Q3, 75th percentile). The bounds of the box, that is, interquartile range (IQR), of the box plot is between Q1 and Q3. The minima and maxima values are defined as Q1 − 1.5 × IQR and Q3 + 1.5 × IQR, respectively. The whiskers are values between minima and Q1 as well as between Q3 and maxima. Values falling outside the Q1–Q3 range are plotted as outliers of the data. Invs, inversions.

sequence nodes within each snarl are extracted as SV allele sequences. For each SV allele, Swave determines the carrier sample(s) and calculates its frequency against the whole snarl using the outputs of the Minigraph 'call' command. The precise classification of these alleles into SV types is deferred to the next two modules.

(2) Sequence-to-image module for allele realignment and resolution (Fig. 1b and Extended Data Fig. 2; Methods). To classify each SV allele sequence (ALT), Swave performs elaborate realignment against the reference sequence (REF) using a dotplot (REF2ALT dotplot). In this image-based representation, matched and reversely matched *k*-mers are denoted as black dots. Although dotplots sensitively reflect SV-induced sequence differences, they are also cluttered by noise signals from genome repetitive sequences, severely disturbing the identification of SVs. To address this, Swave introduces two dotplot waves by projecting the dotplots to the REF axis that quantify dot density for both matched and reversely matched orientations. Then, we realign the REF sequence against itself to obtain a REF2REF dotplot and use its projected waves as genomic background to reduce the noises from repetitive sequence. In brief, non-SV regions in the REF2ALT dotplot mirror the background REF2REF waves, whereas true SVs produce characteristic wave transformations relative to the REF2REF background waves depending on the SV type. Therefore, the SV types are implied from the comparison between the REF2REF and REF2ALT waves.

(3) Deep learning module for SV type prediction (Fig. 1c and Extended Data Fig. 3; Methods). Swave employs an RNN to automatically classify the wave differences into SV types. A simulated dataset is used to train a bidirectional long short-term memory (Bi-LSTM) network. Every recurrent step receives the wave difference at a reference position and output a predicted SV type among five canonical SV types: insertion, deletion, inversion, duplication and duplicated inversion. Bidirectional recurrences enable the model to incorporate both upstream and downstream sequence contexts for accurate classification. In brief, duplications exhibit increased signal intensity, deletions show diminished signals, inversions present reversed patterns and duplicated inversions combine both features of duplication and inversion. Consecutive positions predicted to share the same SV type are merged into single SV components. When multiple such components shared breakpoints within a dotplot, Swave will sequentially merge them into a CSV type.

Performance evaluation

We assessed the performance of Swave alongside other LRS-[2,8–10,38], assembly[15,16] and pangenome-based[35,37] SV calling approaches and merging approaches[25–28] using both simulated and published assemblies.

Individual-level evaluation. Using the HG002 assembly and tier 1 high-confidence SVs[39] as ground truth (Extended Data Fig. 4a and Supplementary Table 1), Swave achieved the highest F1-score (0.957), outperforming the two assembly-based callers, PAV (0.947) and SVIM-asm (0.951). The pangenome-based Vg-vcfwave pipeline lagged behind with an F1-score of 0.791, probably owing to its reliance on allele length differences alone for SV classification. Compared with

LRS-based callers, Swave performed slightly below our previous method SVision-pro (0.963). For individual-level CSV calling, Swave was compared with SVision-pro[2] on its previously simulated CSV ground truth (Extended Data Fig. 4b and Supplementary Table 1). Swave achieved an F1-score of 0.956, substantially higher than SVision-pro on assemblies (0.655) and nearly matching its performance on long-reads (0.967). This reflects Swave's specialized design for assemblies, while SVision-pro was optimized for read-based input.

Trio- and population-level evaluation. We evaluated performance on three parent–child trio assemblies (CHS, PUR and YRI; Fig. 2a and Supplementary Table 2). Swave achieved the highest Mendelian consistency (mean 0.940), surpassing vg-vcfwave (0.914), SVision-pro (0.884), Sniffles2 (0.841) and external merging-required callers (0.225–0.320). At the population scale, we assessed genotyping completeness using the HGSVC 65 samples[19] (Fig. 2a and Supplementary Table 3). External merging-required callers from assembly and LRS exhibited high genotype missing rates (mean 0.881–0.913). Vg-vcfwave and Sniffles2 reduced this to 0.303 and 0.161, respectively, while Swave further minimized the missing rate to 0.075. Owing to their specific designs for trio- and population-level detection, SVision-pro and Sniffles2 showed improved performance relative to other LRS-based callers but still underperformed compared with Swave. For callers that required external merging tools, no notable performance differences were observed between assembly- and LRS-based approaches, both consistently yielding lower performance (Fig. 2a).

We also scaled to larger cohorts (adding HPRC[17] and CPC[18] assemblies), where Swave and vg-vcfwave maintained stable performance while merging-based approaches exhibited a worsening missing rate (Extended Data Fig. 4c and Supplementary Table 3), restricting their utility for downstream population SV analysis. PanPop PART improved the performance of merging-based approaches slightly (Mendelian consistency of 0.332–0.381; genotyping missing rate of 0.587–0.630; Extended Data Fig. 4d and Supplementary Tables 2 and 3) but still underperformed in comparison with Swave.

Overall, the current pangenome, assembly- and LRS-based tools struggle with either SV classification or multisample genotyping—challenges that Swave addresses jointly across SSVs and CSVs.

Enhanced resolution of large and complex inversions

Large inversion polymorphisms and their complex patterns have been associated with genomic instability and genetic disorders[40,41] yet automated detection of them is still challenging for current methods. As neither the GIAB nor the simulated ground truth callset included large inversions, we benchmarked Swave's inversion-calling performance using published calls from the HGSVC dataset (65 samples). We first quantified the detected inversions number at the individual level (Fig. 2b,c), revealing that Swave and assembly-based PAV identified substantially more inversions than another assembly-based SVIM-asm and all read-based callers. Therefore, only Swave and PAV were included in the downstream inversion-relevant analysis. At the population level, Swave identified 156 inversion snarls encompassing 322 alleles, including 129 balanced inversions and 193 complex inversions (Fig. 3a and Supplementary Table 4). For comparison, the latest PAV calls on the same assemblies reported 189 balanced inversions. Given that Swave reported notably more inversions compared with other approaches, we performed computational validations. First, we mapped the reconstructed inversion-feature sequences back to their carrier assemblies, achieving an average mapping integrity of 99% (Extended Data Fig. 5a and Supplementary Table 5), indicating that these inversion-feature sequences could be forwardly and continuously aligned without clipping. Next, we applied two validation metrics sourced from TT-Mars[42] and Vapor[43], confirming that 99% of Swave's inversion calls were supported by at least one validation metric and 97% were supported by both (Extended Data Fig. 5b and Supplementary Table 5).

Balanced inversion. Balanced inversion refers to a directly inverted sequence without any loss or gain of sequence (Fig. 3b). BEDtools[44] intersection revealed that 117 of 189 balanced inversions from PAV calls overlapped with Swave balanced calls, while 61 overlapped with Swave complex inversions (Supplementary Table 6). Strikingly, among these 117 overlapped inversion calls, Swave consistently reported shorter inversion lengths than PAV (Fig. 3b). We validated this against a high-confidence inversion benchmark compiled from assemblies, Strand-seq, Bionano and manual curation[40]. Swave's length estimates showed tight concordance with this benchmark compared with PAV (Extended Data Fig. 5c and Supplementary Table 7; Wilcoxon rank sum test, $P = 2.3 \times 10^{-19}$), yielding a Pearson correlation of 0.99 ($P = 6 \times 10^{-72}$; Fig. 3c), while PAV's calls were poorly correlated, with a Pearson correlation of 0.10 ($P = 0.44$). During this comparison, we found that 97 out of 117 overlapped balanced inversions were flanked by inverted segmental duplications (SDs; Supplementary Table 8), consistent with previous studies linking large inversions with recurrent SDs. These SDs probably misled PAV's breakpoint resolution, leading to systematic overestimation of inversion lengths (Extended Data Fig. 5d).

Complex and polymorphic inversions. Of the 156 inversion-related snarls identified by Swave, approximately two-thirds ($n = 128$; Fig. 3d and Supplementary Table 4) corresponded to flanked events, where inversions ($n = 43$) or duplicated inversions ($n = 85$) were neighbored by additional SV breakpoints precisely at their boundaries. Notably, Swave uncovered a previously undescribed subclass of complex inversion, termed as scarred inversions ($n = 71$; Fig. 3d,e and Supplementary Table 9), defined by insertion or deletion breakpoints ('scars') occurring inside the inversion bodies. Most scarred inversions (63/71; Supplementary Table 9) contained a single internal scar, while seven had 2 and one included up to 4 (Extended Data Fig. 6a), totaling 81 internal scars. These scars were typically small (Extended Data Fig. 6a): 75 of 81 were under 5,000 bp, with the largest spanning 18,451 bp and removing 24% of the original inverted sequence. Existing tools failed to capture the full structure of these scarred inversions (Fig. 3f). Specifically, PAV misclassified 72% as simple balanced inversions, reported the internal scars in 5% and entirely missed the remaining 23%. SVIM-asm exhibited the opposite tendency, reporting the scars for 70% of scarred inversions while missing the inversion itself. Compared with assembly-based methods, all LRS-based callers recovered a notably lower proportion of the sub-breakpoints, with the proportion of entirely missed events ranging from 75% to 96%.

Among the 156 inversion-related snarls, most of them (102, 65%) were bi-allelic (one wild-type allele and one inversion allele), whereas 54 (35%; Supplementary Table 4) were multi-allelic, harboring an average of 4 distinct alleles. This high allelic diversity reflects the extensive polymorphism of complex inversions across samples. One form of polymorphism in complex inversions was driven by the diversity in types and lengths of additional breakpoints. In scarred inversions, much of the diversity was attributable to variation in the types and lengths of internal scars (Extended Data Fig. 6). For instance, a single graph snarl (>s142244>s142248) harbored five co-existing scarred inversions formed by different combinations of four scar regions (Extended Data Fig. 6a). Repeat elements further contributed to this diversity: 40 of 81 scars occurred within repetitive regions (Extended Data Fig. 6c and Supplementary Table 9), where expansions or contractions probably gave rise to insertion or deletion scars, respectively (Extended Data Fig. 6d). Another form of polymorphism involved the co-occurrence of the three inversion categories (balanced, flanked and scarred inversions; Fig. 3g) at individual loci. Of the 54 multi-allelic snarls, 30 featured two inversion classes and three contained all three (Fig. 3g). For example, the snarl >s21053>s21059, spanning chr1: 247,955,488–247,975,948, included balanced (AF of 0.25), scarred (AF of 0.039) and flanked (AF of 0.0078) forms (Fig. 3h and Supplementary Table 10). The balanced inversion represented

https://doi.org/10.1038/s41588-026-02538-6

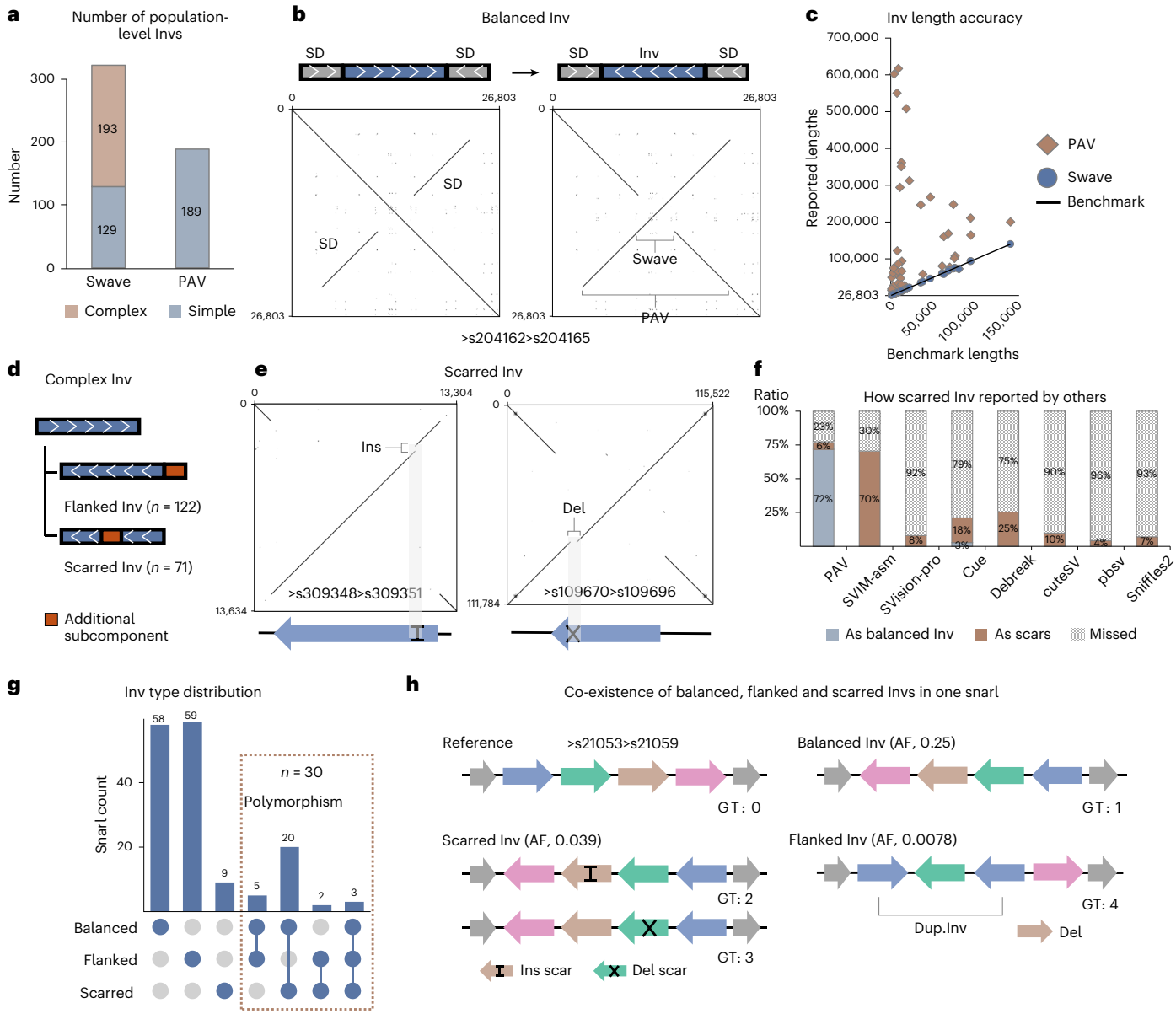


Fig. 3 | Characterization and benchmarking of inversions. a, Total numbers of detected inversions from Swave, PAV and SVIM-asm. Only Swave identified complex inversion types. **b**, Example of a balanced inversion (snarl >s204162>s204165) located between inverted SDs, potentially leading to overestimation of event length. **c**, Comparison of length discrepancies in balanced inversions called by PAV and Swave relative to the benchmark lengths. **d**, Classification of complex inversions detected by Swave into flanked and scarred subtypes. **e**, Representative scarred inversions containing internal insertion scar (left, >s309348>s309351) and deletion scar (right, >s109670>s109696), respectively. **f**, Detection performance for scarred inversions by PAV and SVIM-asm. Neither method successfully resolved these variants. **g**, Distribution of inversion types across the 156 inversion-associated snarls, with 30 snarls containing multiple inversion types. **h**, Example of a multicategory snarl (>s21053>s21059), harboring balanced (AF of 0.25), scarred (AF of 0.039) and flanked (0.0078) inversion alleles. Ins, insertion; Del, deletion; Dup.Inv, duplicated inversion.

the predominant allele, while scarred and flanked alleles introduced unique breakpoint complexities at the same genome locus. A similar pattern was observed in 28 out of the 30 the multicategory inversion snarls featuring a balanced inversion allele, where the balanced form was the most common in 18 cases (Supplementary Table 4).

Resolved SVs in healthy population

Having demonstrated Swave's superior performance in detecting SSVs and CSVs using pangenomes, we next applied it to the pangenomes from three of the largest healthy cohorts so far: HGSVC (130 haplotypes), HPRC (88 haplotypes) and CPC (116 haplotypes), exploring the population-level profiles of challenging SSVs and previously underestimated CSVs (Fig. 4a).

As a result, 134,944 SSV snarls comprising in total of 316,808 alleles were identified (Fig. 4b, left). Repetitive genomic regions were a major source of multi-allelic SSV snarls. Annotating snarls of which over 80% of their variant sequences were repetitive reduced the callset to 92,777 snarls and 112,483 alleles, bringing the average allele number per snarl down from 2.3 to 1.2. Swave also detected 1,649 CSV alleles from 1,097 snarls, of which only 230 were annotated as repetitive. Repetitive sequences had a notably minor impact on CSVs compared with SSVs: annotation decreased the average allele count modestly from 1.5 to 1.4 and only 25% of CSVs were annotated, substantially fewer than the 64% observed for SSV (Fig. 4b, right). The remaining 867 nonrepetitive snarls comprised 1,232 complex alleles (Fig. 4c). Among these, rare CSVs (AF <1%, $n = 755$, 61%) were more prevalent than infrequent (1–5%,

https://doi.org/10.1038/s41588-026-02538-6

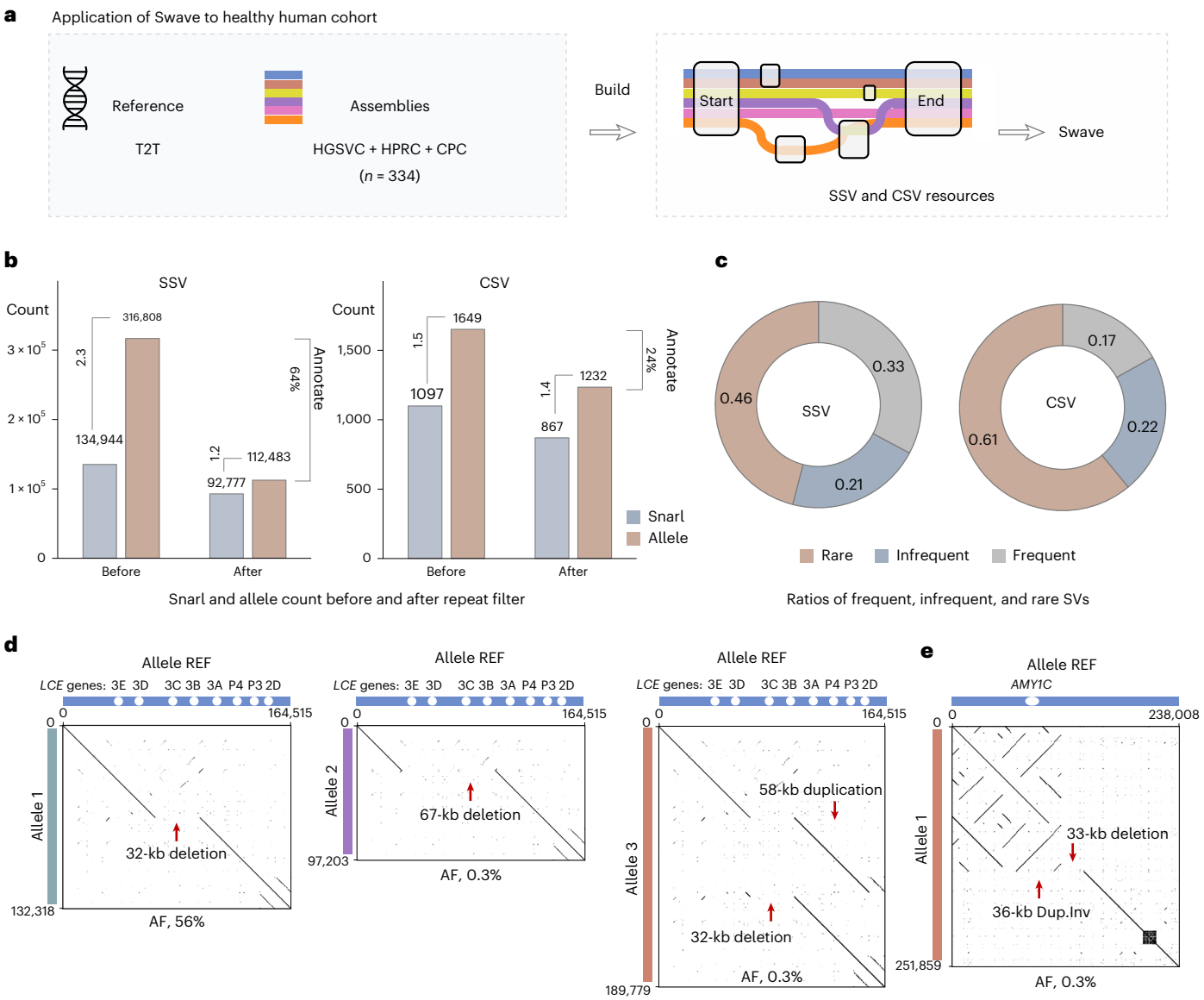


Fig. 4 | Population-scale discovery of SSVs and CSVs using Swave. a, Schematic of Swave's application to healthy genome cohorts (HGSVC, HPRC and CPC). Three samples (HG02818, HG00733 and NA19240) from HPRC (47 in total) overlapped with HGSVC and were therefore excluded, resulting in (47 − 3) × 2 = 88 haplotypes. No samples were excluded from the HGSVC or CPC cohorts. **b**, Total counts of SSVs and CSVs in the combined HGSVC, HPRC and CPC cohorts. Repeat annotation was applied to both variant types. Allele counts per snarl are shown to the left of each bar, and the fraction of repeat-overlapping snarls is noted to the right. **c**, The AF distribution of retained SSVs and CSVs, stratified into frequent (AF >5%), infrequent or rare (AF <1%) categories. **d**, A frequent SSV (left, deletion of *LCE3B*/*LCE3C*) and two rare alleles (middle, SSV; right, CSV involving a duplication flanked by a deletion) at the same locus. The rare alleles introduce greater disruption to the *LCE* gene cluster. **e**, Example of a novel and rare CSV locus (a duplicated inversion flanked by a deletion) that increases the copy number of *AMY1C*, a gene with known dosage variability.

n = 274, 22%) and frequent (>5%, *n* = 203, 17%) variants. Notably, the proportion of rare CSVs exceed that of SSVs (46% versus 61%; Fig. 4c and Supplementary Table 11), highlighting the need to better characterize these overlooked rare events.

Rare CSV alleles. Swave captured rare CSV alleles at both multi-allelic (Fig. 4d) and bi-allelic genomic regions (Fig. 4e). In total, 237 (31%) of these rare alleles were identified within mixed snarls (*n* = 103) also containing SSV alleles (Supplementary Table 12). Among these, 70 snarls featured more common SSVs than CSVs, indicating that the CSVs were minor alleles that either introduced new breakpoints atop existing SSVs (Fig. 4d) or restructured this locus entirely (Extended Data Fig. 7d). For example, the well-characterized 32-kb deletion affecting *LCE3B*/*LCE3C* genes (Fig. 4d, left), which was strongly associated with psoriasis and estimated to have persisted for at least 45,000 years[45], was found alongside two novel singleton alleles (Extended Data Fig. 7a). One extended the deletion to 67 kb (Fig. 4d, middle) and the other introduced a 58-kb downstream duplication precisely flanking the ancestral 32-kb deletion (Fig. 4d, right, and Extended Data Fig. 7b). Another notable case occurred upstream of *VIPR2*, a gene linked to schizophrenia risk[46,47]. This region exhibited highly variable repeat expansions and contractions across populations without affecting the gene sequence itself, yet Swave identified a rare 28-kb scarred inversion that partially disrupted the *VIPR2* gene body (Extended Data Fig. 7d).

The remaining 518 rare alleles (69%) were from bi-allelic CSV snarls (*n* = 436; Supplementary Table 12) containing only a wild-type allele alongside the CSV allele. Although these snarls occurred outside of other loci, they could still influence the same gene by introducing novel

https://doi.org/10.1038/s41588-026-02538-6

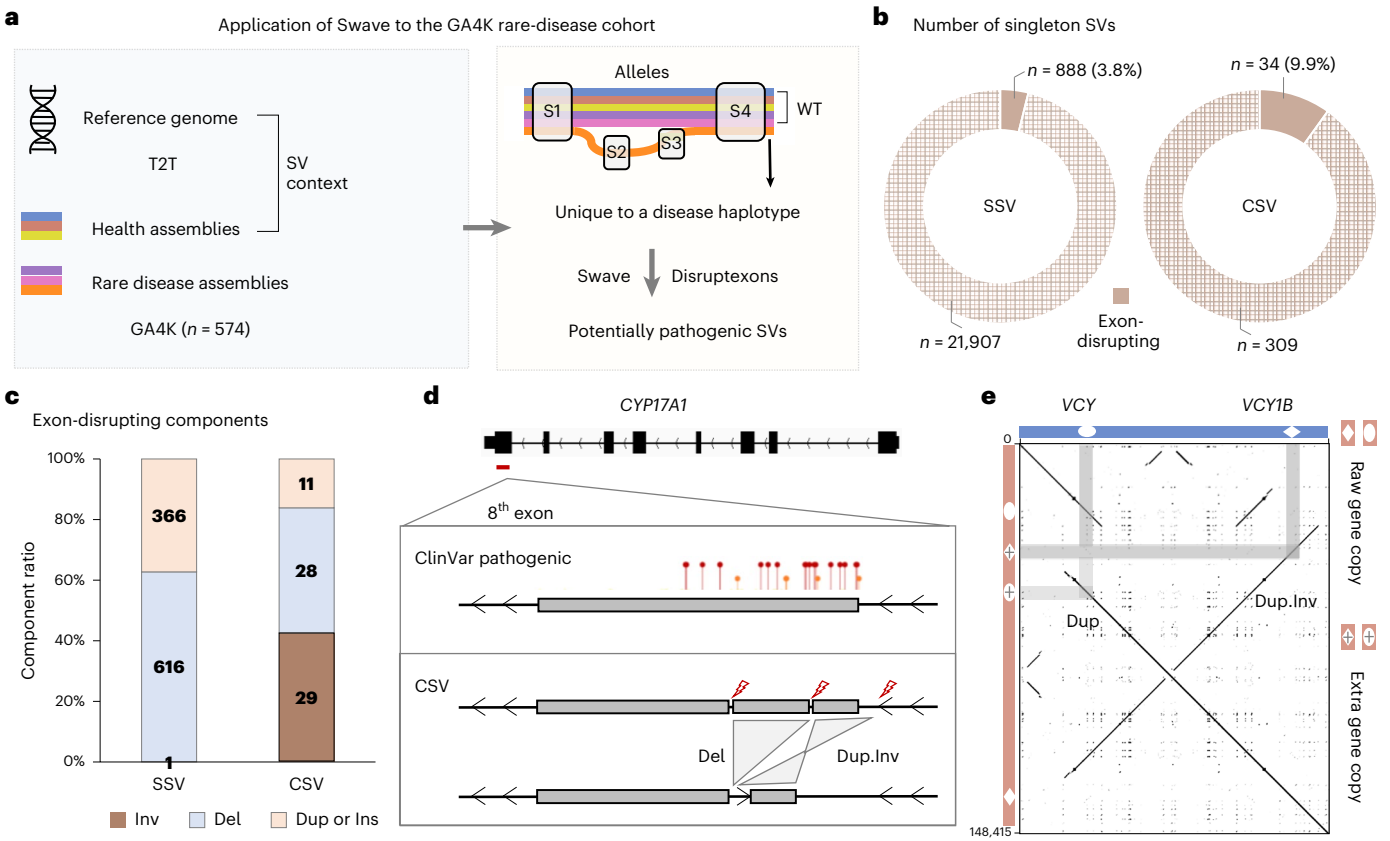


Fig. 5 | Discovery of potentially pathogenic SVs in the GA4K rare-disease cohort. a, Schematic of Swave's application to the GA4K rare disease cohort. Assemblies from both healthy and rare-disease cohorts were combined to construct a pangenome graph. All 47 HPRC samples were included in the pangenome graph. Alleles exclusive to a single disease genome (singletons) were considered potentially pathogenic variants. **b**, Counts of singleton SSVs and CSVs. Exon-disrupting events are indicated as solid pie segments. The proportion of exon-disrupting CSVs was more than twice that of SSVs. **c**, Component type distribution among exon-disrupting singleton SSVs and CSVs. Exon-disrupting inversions were more frequently observed as CSV subcomponents ($n = 29$) than as standalone SSVs ($n = 1$). **d**, A singleton CSV (a duplicated inversion flanked by a deletion) disrupting the eighth exon of *CYP17A1* that harbors multiple known pathogenic (red pushpins) and probably pathogenic (short orange pushpins) small variants in ClinVar. **e**, Example of a CSV (a duplicated inversion flanked by a duplication) that doubled the copy number of two paralogous genes *VCY* and *VCY1B* through distinct subcomponents.

CSV loci, for example, a novel and rare CSV locus (AF of 0.3%; Fig. 4e) on the *AMY1C* gene, known for a highly variable copy number driven by palindromic and tandem repeats[22,23]. This novel event occurred downstream of the repeats rather than within it as commonly observed (Extended Data Fig. 7c). The variant consisted of a 36-kb duplicated inversion, adding one *AMY1C* copy, that replaced a 33-kb sequence located downstream of the repeat boundary.

Resolved SVs in rare-disease cohort

Identifying pathogenic SVs in rare genetic diseases requires careful exclusion of those found in healthy cohorts, which are typically considered as benign, to pinpoint SVs specifically presented in disease genomes. A recent effort constructed the largest publicly available rare disease pangenome so far, comprising 287 pediatric disease genomes (574 haplotypes) from the Genomic Answers for Kids (GA4K)[21], alongside 94 HPRC haplotypes as healthy control genomes. Here, we applied Swave to this pangenome to profile SVs and uncover novel complex patterns potentially associated with disease (Fig. 5a).

Swave identified 343 singleton CSVs unique to GA4K genomes (Fig. 5b and Supplementary Table 13). These variants, supported by only a single disease haplotype and absent from all healthy and other disease haplotypes, were predominantly complex inversions ($n = 307$; 90%). Although there were far fewer singleton CSVs than singleton SSVs ($n = 22{,}795$; Supplementary Table 14), the proportion of exon-disrupting CSVs ($n = 34$, 9.9%; Supplementary Table 13) was more than twice that of SSVs ($n = 888$, 3.9%; Fig. 5b and Supplementary Table 14). Given the much higher number of SSVs, it was unexpected that 29 out of 30 exon-disrupting inversions were found as subcomponents of CSVs rather than standalone SSVs (Fig. 5c), highlighting Swave's ability to resolve inversion complexity. Most of the exon-disrupting CSVs (30/34) altered only a single gene by modifying exon content through one or multiple CSV subcomponents. One example involved a duplicated inversion flanked by a deletion that disrupted the eighth exon of *CYP17A1* (Fig. 5d). While small variants affecting this exon have already been classified as pathogenic for congenital adrenal hyperplasia in ClinVar (Supplementary Table 15), this more complex rearrangement illustrates Swave's capacity to uncover previously undetected variant forms. Annotation of the altered gene sequence using four tools revealed that the eighth coding exon was incomplete compared with the original gene structure (Extended Data Fig. 8a), leading to the loss of residues 415–508 (Extended Data Fig. 8b), including residue 442, a critical Fe-binding site of the heme group (Extended Data Fig. 7c). Among the remaining four exon-disrupting CSVs, three affected paralogous genes through copy number changes. A particularly illustrative case involved *VCY*/*VCY1B* paralog genes (Fig. 5e): one duplication subcomponent increased the *VCY* copy number, while an inversion duplication subcomponent simultaneously increased *VCY1B* copies. Both paralogs were copy gained within a single CSV event but through two distinct subcomponents.

In comparison, exon-disrupting singleton SSVs ($n = 888$; Supplementary Table 13) were structurally simpler: 613 were deletions, and 274 were insertions or duplications, while only 1 was an inversion. This singleton inversion (411 bp) reversed the second exon of *HYLS1*, a

gene implicated in hydrolethalus syndrome (Extended Data Fig. 8d). While most exon-disrupting singleton SSVs were small, 85 exceeded 10 kb in length, with the largest one reaching 69 kb. For instance, a 43-kb deletion spanned introns 6–14 of *FRAS1*, a gene associated with Fraser syndrome[48,49] (Extended Data Fig. 8e). Collectively, these findings demonstrated Swave's capability to resolve intricate and previously overlooked SSV and CSV patterns within rare-disease-risk genes, providing a critical resource for downstream investigations of pathogenic mechanisms.

Discussion

The pangenome graph currently offers the most powerful framework for integrating hundreds of high-quality genome assemblies. To address the persistent challenge of resolving SVs, particularly CSVs, from pangenome graph-embedded alleles, we developed Swave. Unlike existing pangenome-based methods that infer SV types primarily from the length differences between REF and ALT alleles, Swave leverages precise alignments and introduces projection waves to denoise repetitive sequences and support SV classification. Given the series format of waves, Swave uses a Bi-LSTM network to capture both upstream and downstream contexts, notably improving prediction accuracy. These design innovations together underpin Swave's effectiveness and accuracy in SV characterization. We also evaluated the runtime and memory consumption across datasets ranging from 130 to 668 haplotypes (Extended Data Fig. 3c and Supplementary Table 16), demonstrating the high computational efficiency of Swave on both a personal computer (4.13–12.07 h, 16–23 GB memory) and a cluster node (1.37–2.53 h and 17–24 GB memory).

We applied Swave to healthy cohorts, generating population-scale SV profiles that serve as valuable references for future analyses. Given the structural complexity and high polymorphism, inversions were prioritized. Remarkably, using only assemblies, Swave achieved breakpoint accuracy comparable to the consensus of multiple sequencing technologies and revealed previously unrecognized complex and polymorphic inversion patterns. The population-scale power of pangenome graphs further enable us to explore SVs in large rare-disease cohort. Applying Swave to GA4K pangenome allowed us to identify singleton SVs as potentially pathogenic ones. While experimental validation was beyond the scope of this study, many of these newly discovered variants disrupted known disease-associated genes or introduced novel alterations beyond known pathogenic variants. Considering that over 90% of GA4K samples remained undiagnosed after standard analysis of microarray and sequencing data, SV-level investigation using Swave may contribute to uncovering the mechanisms behind these previously unexplained rare diseases.

The false negatives of Swave were primarily attributed to its reliance on qualities of genome assemblies and pangenome graphs. Although it demonstrated strong genotyping accuracy across population-scale samples, a subset of genotypes remained missing. These cases were attributed to incomplete assemblies, as no sequences from carrier genomes were mapped to the corresponding snarls (Extended Data Fig. 9a). Additional analysis revealed that these problematic snarls were concentrated near centromeres and telomeres (Extended Data Fig. 9b), suggesting that continued improvements in assembly quality, particularly in these highly repetitive regions, will be the key to close the remaining gaps.

The majority of false positives arose from dispersed duplications, where the source sequence originated either from distant loci on the same chromosome (Extended Data Fig. 10a) or from different chromosomes (Extended Data Fig. 10b). This limitation stems from the current dotplot design, which only aligns relatively local sequences. As a result, distant source sequences were not captured and were instead reported as insertions (Extended Data Fig. 10c). Incorporating boosted dotplots with synthesized or multisource reference regions would be a promising direction for future improvements.

Overall, Swave expands the utility of pangenome for comprehensive SV discovery and interpretation. Future application could leverage Swave and pangenome graphs to enhance the evolutionary analysis of SSVs and CSVs across diverse human ethnic groups or comparison between human and closely related primates, enabling the reconstruction of SV-driven evolutionary trajectories. Clinically, Swave offers a promising way for building comprehensive SV catalogs that facilitate the identification of pathogenic SVs, advancing our understanding of genetic disease mechanisms and supporting the diagnosis of genetic diseases.

Online content

Any methods, additional references, Nature Portfolio reporting summaries, source data, extended data, supplementary information, acknowledgements, peer review information; details of author contributions and competing interests; and statements of data and code availability are available at https://doi.org/10.1038/s41588-026-02538-6.

References

1. Ahsan, M. U., Liu, Q., Perdomo, J. E., Fang, L. & Wang, K. A survey of algorithms for the detection of genomic structural variants from long-read sequencing data. *Nat. Methods* **20**, 1143–1158 (2023).
2. Wang, S. et al. De novo and somatic structural variant discovery with SVision-pro. *Nat. Biotechnol.* **43**, 181–185 (2025).
3. Ding, W. et al. Adaptive functions of structural variants in human brain development. *Sci. Adv.* **10**, eadl4600 (2024).
4. Collins, R. L. & Talkowski, M. E. Diversity and consequences of structural variation in the human genome. *Nat. Rev. Genet.* **26**, 443–462 (2025).
5. Li, Y. et al. Patterns of somatic structural variation in human cancer genomes. *Nature* **578**, 112–121 (2020).
6. Mahmoud, M. et al. Structural variant calling: the long and the short of it. *Genome Biol.* **20**, 246 (2019).
7. Lin, J. et al. SVision: a deep learning approach to resolve complex structural variants. *Nat. Methods* **19**, 1230–1233 (2022).
8. Chen, Y. et al. Deciphering the exact breakpoints of structural variations using long sequencing reads with DeBreak. *Nat. Commun.* **14**, 283 (2023).
9. Popic, V. et al. Cue: a deep-learning framework for structural variant discovery and genotyping. *Nat. Methods* **20**, 559–568 (2023).
10. Smolka, M. et al. Detection of mosaic and population-level structural variants with Sniffles2. *Nat. Biotechnol.* **42**, 1571–1580 (2024).
11. Denti, L., Khorsand, P., Bonizzoni, P., Hormozdiari, F. & Chikhi, R. SVDSS: structural variation discovery in hard-to-call genomic regions using sample-specific strings from accurate long reads. *Nat. Methods* **20**, 550–558 (2023).
12. Wang, S. & Ye, K. Deep-learning based representation and recognition for genome variants-from SNVs to structural variants. *Natl Sci. Rev.* **11**, nwae335 (2024).
13. Olson, N. D. et al. Variant calling and benchmarking in an era of complete human genome sequences. *Nat. Rev. Genet.* **24**, 464–483 (2023).
14. Liu, Y. H., Luo, C., Golding, S. G., Ioffe, J. B. & Zhou, X. M. Tradeoffs in alignment and assembly-based methods for structural variant detection with long-read sequencing data. *Nat. Commun.* **15**, 2447 (2024).
15. Ebert, P. et al. Haplotype-resolved diverse human genomes and integrated analysis of structural variation. *Science* **372**, abf7117 (2021).
16. Heller, D. & Vingron, M. SVIM-asm: structural variant detection from haploid and diploid genome assemblies. *Bioinformatics* **36**, 5519–5521 (2021).
17. Liao, W. W. et al. A draft human pangenome reference. *Nature* **617**, 312–324 (2023).
18. Gao, Y. et al. A pangenome reference of 36 Chinese populations. *Nature* **619**, 112–121 (2023).

19. Logsdon, G. A. et al. Complex genetic variation in nearly complete human genomes. *Nature* **644**, 430–441 (2025).
20. Collins, R. L. et al. A structural variation reference for medical and population genetics. *Nature* **581**, 444–451 (2020).
21. Groza, C. et al. Pangenome graphs improve the analysis of structural variants in rare genetic diseases. *Nat. Commun.* **15**, 657 (2024).
22. Yilmaz, F. et al. Reconstruction of the human amylase locus reveals ancient duplications seeding modern-day variation. *Science* **386**, eadn0609 (2024).
23. Bolognini, D. et al. Recurrent evolution and selection shape structural diversity at the amylase locus. *Nature* **634**, 617–625 (2024).
24. Plender, E. G. et al. Structural and genetic diversity in the secreted mucins MUC5AC and MUC5B. *Am. J. Hum. Genet.* **111**, 1700–1716 (2024).
25. Jeffares, D. C. et al. Transient structural variations have strong effects on quantitative traits and reproductive isolation in fission yeast. *Nat. Commun.* **8**, 14061 (2017).
26. Kirsche, M. et al. Jasmine and Iris: population-scale structural variant comparison and analysis. *Nat. Methods* **20**, 408–417 (2023).
27. English, A. C., Menon, V. K., Gibbs, R. A., Metcalf, G. A. & Sedlazeck, F. J. Truvari: refined structural variant comparison preserves allelic diversity. *Genome Biol.* **23**, 271 (2022).
28. Zheng, Z. Y. et al. A sequence-aware merger of genomic structural variations at population scale. *Nat. Commun.* **15**, 960 (2024).
29. Jayakodi, M. et al. Structural variation in the pangenome of wild and domesticated barley. *Nature* **636**, 654–662 (2024).
30. Bian, P. et al. A graph-based goat pangenome reveals structural variations involved in domestication and adaptation. *Mol. Biol. Evol.* **41**, msae251 (2024).
31. Li, H., Feng, X. & Chu, C. The design and construction of reference pangenome graphs with minigraph. *Genome Biol.* **21**, 265 (2020).
32. Hickey, G. et al. Pangenome graph construction from genome alignments with Minigraph-Cactus. *Nat. Biotechnol.* **42**, 663–673 (2024).
33. Garrison, E. et al. Building pangenome graphs. *Nat. Methods* **21**, 2008–2012 (2024).
34. Cui, Y., Peng, C., Xia, Z., Yang, C. & Guo, Y. A survey of sequence-to-graph mapping algorithms in the pangenome era. *Genome Biol.* **26**, 138 (2025).
35. Garrison, E. et al. Variation graph toolkit improves read mapping by representing genetic variation in the reference. *Nat. Biotechnol.* **36**, 875–879 (2018).
36. Andreace, F., Lechat, P., Dufresne, Y. & Chikhi, R. Comparing methods for constructing and representing human pangenome graphs. *Genome Biol.* **24**, 274 (2023).
37. Garrison, E., Kronenberg, Z. N., Dawson, E. T., Pedersen, B. S. & Prins, P. A spectrum of free software tools for processing the VCF variant call format: vcflib, bio-vcf, cyvcf2, hts-nim and slivar. *PLoS Comput. Biol.* **18**, e1009123 (2022).
38. Jiang, T. et al. Long-read-based human genomic structural variation detection with cuteSV. *Genome Biol.* **21**, 189 (2020).
39. Zook, J. M. et al. A robust benchmark for detection of germline large deletions and insertions. *Nat. Biotechnol.* **38**, 1347–1355 (2020).
40. Porubsky, D. et al. Recurrent inversion polymorphisms in humans associate with genetic instability and genomic disorders. *Cell* **185**, 1986–2005 (2022).
41. Vollger, M. R. et al. Segmental duplications and their variation in a complete human genome. *Science* **376**, abj6965 (2022).
42. Yang, J. & Chaisson, M. J. P. TT-Mars: structural variants assessment based on haplotype-resolved assemblies. *Genome Biol.* **23**, 110 (2022).
43. Zhao, X., Weber, A. M. & Mills, R. E. A recurrence-based approach for validating structural variation using long-read sequencing technology. *Gigascience* **6**, 1–9 (2017).
44. Quinlan, A. R. & Hall, I. M. BEDTools: a flexible suite of utilities for comparing genomic features. *Bioinformatics* **26**, 841–842 (2010).
45. Pajic, P., Lin, Y. L., Xu, D. & Gokcumen, O. The psoriasis-associated deletion of late cornified envelope genes *LCE3B* and *LCE3C* has been maintained under balancing selection since human Denisovan divergence. *BMC Evol. Biol.* **16**, 265 (2016).
46. Ago, Y., Asano, S., Hashimoto, H. & Waschek, J. A. A. Probing the VIPR2 microduplication linkage to schizophrenia in animal and cellular models. *Front. Neurosci.* **15**, 717490 (2021).
47. Chen, C. H. et al. Identification of rare mutations of the vasoactive intestinal peptide receptor 2 gene in schizophrenia. *Psychiatric Genet.* **32**, 125–130 (2022).
48. Pitera, J. E., Scambler, P. J. & Woolf, A. S. Fras1, a basement membrane-associated protein mutated in Fraser syndrome, mediates both the initiation of the mammalian kidney and the integrity of renal glomeruli. *Hum. Mol. Genet.* **17**, 3953–3964 (2008).
49. Slavotinek, A., Li, C., Sherr, E. H. & Chudley, A. E. Mutation analysis of the FRAS1 gene demonstrates new mutations in a propositus with Fraser syndrome. *Am. J. Med. Genet. A* **140a**, 1909–1914 (2006).

Methods

Swave methodology

Pangenome graph construction and allele extraction. Swave applies Minigraph for pangenome graph construction (Extended Data Fig. 1a), which supports the identification of SVs larger than 50 bp. Alternative tools such as PGGB[33] and Minigraph-cactus[32] perform more fine-grained sequence alignment and graph construction suitable for smaller variants (for example, single nucleotide polymorphisms and Indels), which are not involved in this study. The resulting graph is encoded in GFA format (Extended Data Fig. 1b), comprising nodes (sequence segments with associated lengths) and edges (connections between nodes). Paths, formed by concatenating edge-linked nodes, represent individual assembly or reference traversals through the graph.

Minigraph offers a realignment process (the '–call' option) that maps each assembly back to the graph to recover its traversal paths. Graph regions where the assembly paths diverge are decomposed into snarls, which are considered as candidate SV loci (Extended Data Fig. 1b). For each snarl, Swave extracts traversal allele paths and reconstruct full sequences using node information in the GFA. These allele sequences are then passed to downstream modules for dotplot-based realignment and SV type classification. Allele frequencies are computed by tallying the number of carrier assemblies per variant (Extended Data Fig. 1c). When phased assemblies are provided, each haplotype is processed independently during path parsing from the pangenome graph, and SVs are detected on each haplotype-specific path. Sample-level genotypes are then generated by combining the haplotype-level genotypes using the '|' symbol, following the input haplotype order (Extended Data Fig. 1d).

Dotplot image representation. To evaluate sequence-level difference between REF and ALT alleles, Swave generates three types of dotplot images:

- REF2REF: alignment of reference sequence against itself, revealing the genomic baseline of REF
- ALT2ALT: alignment of alternative sequence against itself, revealing the genomic baseline of ALT
- REF2ALT: alignment of alternative sequence (*Y* axis) against the reference sequence (*X* axis), revealing the structural divergence between them

(1) Dotplot generation: Swave begins by extracting all the *k*-mers (default $k = 30$ bp, stride of 1 bp) from the longer sequence (REF or ALT), then traverses *k*-mers in the shorter sequence to identify exact forward or reverse matches. For example (Extended Data Fig. 2a, top), at the condition of REF.length > ALT.length, Swave collects all the *k*-mers from the REF sequence. For genome position REF_i, the *k*-mer where size *k* is defined as the *k* bp subsequence from REF_i: REF_{i+k}. If it matches with the *k*-mer from ALT_j, Swave places a dot at the coordinate of REF_i, ALT_j in the image. The matching orientations (forward and reverse) are recorded. Then, Swave moves to next *k*-mer $REF_{i+\text{stride}}$ and repeats the above process. After constructing the full dot matrix, Swave identifies linear dot clusters exceeding 50 bp in length (aligns with Swave's focus on SVs), which represent structural continuity between the reference and allele, serving as the skeleton of the dotplot.

(2) Dotplot optimization: As the *k*-mer-based approach lacks single-base resolution owing to the fact that the alignment gap flanking SV breakpoints may appear up to $(k-1)$ bases longer than their true lengths (Extended Data Fig. 2a, top), Swave performs localized base-level remapping by examining whether the bases can exactly match at the *k*-mer stop-matching boundaries (Extended Data Fig. 2a, bottom), which are defined as the two end points of the lines identified above. To enhance efficiency, this is applied only to lines uniquely present in the REF2ALT dotplot, excluding those shared with REF2REF or ALT2ALT dotplots, which typically reflect repetitive genomic backgrounds rather than informative SV signals.

Dotplot projection for waves. A major challenge in using dotplot images is the massive and redundant dots caused by genome repetitive sequences. To suppress noise from repetitive sequences and enhance SV signals, Swave introduces a novel representation form called projection waves, which distills alignment information at each genomic position from the dotplot image (Fig.1b, left). For example, when projecting a dotplot onto its *X* axis, Swave traverses each position X_i along the *X* axis and counts the number of dots aligned across all *Y* axis rows (Y_1 to Y_n) at X_i. Therefore, the wave summarizes the matching dot number at each genome position within the projection axis. Each projection operation will generate two waves—one for forward matches and one for reverse matches. This dual-wave design preserves strand orientations, aiding in the detection of inversions.

(1) Background wave: Swave first projects the REF2REF dotplot onto its *X* axis to generate the dual wave representing the reference sequence background. Similarly, it projects the ALT2ALT dotplot to obtain ALT allele background. These background waves reflect the internal repetitive features of each allele sequence. The average value of these background waves reflects the level of repetitiveness. For example, a sequence without any repetitive components will have a wave displaying a uniform distribution with an average value close to 1 (Extended Data Fig. 2b, left). By contrast, a sequence with dense repetitive contents produces an obviously fluctuating wave with an elevated average wave value (Fig.1b, middle, and Extended Data Fig. 2b, right).

(2) SV-indicating wave: Next, Swave projects the REF2ALT dotplot onto the *X* axis (reference sequence) to generate the SV-indicating dual wave (Fig.1b, right). Dots that form the lines identified above are assigned a higher weight in the projection process, with the default weight set to the average value of the background wave. By comparing the resulting SV-indicating dual wave to the REF2REF background dual wave, different SV types lead to characteristic wave shifts: deletions manifest as local wave reduction, duplications as elevations and inversions exhibit new emerging value peaks in the wave-recording reversely matched dots. In addition, as insertions can be interpreted as relative deletions that happened in the REF sequence, Swave also projects the REF2ALT dotplot onto the *Y* axis (alternative sequence). Insertions will manifest as reduced waves compared with the ALT2ALT background waves, analogous to how deletions alter the REF2REF background waves.

RNN for SV classification. The projection waves are stored in a series data format, and, therefore, Swave leverages an RNN architecture for SV type prediction.

(1) RNN design: Wave-derived data are encoded as sequences of four-feature tuples per reference position (Extended Data Fig. 3a): (i) genome position, (ii) average value of background wave, and (iii, iv) differences between SV-indicating and background waves for forward and reverse matches, respectively. To reduce redundancy, Swave merges consecutive genome positions with identical values for the latter three features, and then replaces the first feature of individual positions with the spanning lengths. Nevertheless, a single SV region may still contain multiple tuples due to the presence of genome repeats (Extended Data Fig. 3a). To address this, Swave employs a Bi-LSTM architecture as the core of RNN (Extended Data Fig. 3b), enabling the model to consider the entire context. This contextual awareness helps accurately classify multiple tuples as belonging to the same SV type.

(2) RNN training: Swave uses a simulated dataset, which provides the exact accurate SV types and breakpoints, to train the RNN mode. Real-world SV datasets are derived from callers, leading to potential inaccuracy and limited coverage of inversions and duplications. The simulated dataset comprises five SSVs, including insertion, deletion, inversion, duplication and inverted duplication. For each simulated SSV, its corresponding dotplot and projection waves are generated as described above, and labels for each wave tuple are directly assigned on the basis of the coordinates of the simulated SV region.

Following the pipeline applied by SVision-pro, 1,000 events were simulated for each of the five SSV types using the VISOR randomregion.R module. The simulated lengths followed a normal distribution with a mean of 500 bp and variance of 150 bp. Note that, for the two duplication SSVs, dispersed events and tandem events were equally simulated (500 events for each). The simulated dataset is split 70:30 into training and validation sets. During the training procedure, the batch size is set to 128 and the learning rate is set to 0.0001. The loss function is defined as the cross entropy loss. Adam Optimizer is used to guild the training process. In addition, an early stopping strategy is implemented to determine the best trained models. This strategy ends the training when the validation accuracy remains unchanged (within a tolerance of 0.001) for a continuous period of 30 epochs. The training process ended at the seventh epoch, and the trained model achieved an average of 0.99 classification accuracy on the validation set.

Benchmarking methodology

Assembly alignment. All genome assemblies were aligned to the human reference genome using minimap2[50] with the following parameters: '--MD -x asm20 -m 10000 -z 10000,50 -r 50000 --end-bonus=100 --secondary=no -O 5,56 -E 4,1 -B 5 -a --eqx -Y'. These settings were derived from previous studies[15,51] and offered improved alignments from default configurations.

Pangenome graph construction. The reference genome and sample assemblies were sequentially inputted into the Minigraph with parameter '-cxggs' for pangenome graph construction. To avoid naming conflicts that may generate warnings in Minigraph during graph construction, all contigs were renamed to include sample and haplotype identifiers. Following construction, the input sequences were mapped back to the graph and the corresponding paths for each graph snarls were determined using the Minigraph–call function. Full assembly-specific paths through the graph were then reconstructed using a custom script (see 'Code availability') that concatenates individual snarl paths sorted by reference genome positions.

SV calling and merging. Assembly-based callers included PAV[15] and SVIM-asm[16], whereas LRS-based callers included SVision-pro[2], Cue[9], Debreak[8], cuteSV[38], pbsv and Sniffles2[10]. Pangenome-based SV calling approaches generally classify SVs on the basis of allele length difference within snarls. Simple and bi-allelic snarls were directly classified into deletions or insertions on the basis of allele length differences. By contrast, complex and multi-allelic snarls were first processed to reduce allele complexity before classification. In this evaluation, we applied a widely adopted pipeline in recent pangenome studies (for example, HPRC[17] and CPC[18]) where the pangenome graphs were processed sequentially using vg[35], vcfbub and vcfwave[37] (vg-vcfwave for short) for SV discovery.

Pangenome-based methods, such as Swave and vg-vcfwave, can directly output genotype-aware population callset. By contrast, both LRS- and assembly-based callers require integration of single-sample callsets into a population-level set, typically using external merging tools such as SURVIVOR[25], Jasmine[26] and Truvari[27]. Sniffles2 incorporates an internal merging module to aggregate individual callsets. SVision-pro implements a trio-detection mode, while its population-detection mode is under development and currently depends on external merging. Further refinement of the merged results was performed using PanPop PART[28].

Individual-level benchmark. For SSV detection, the HG002 assembly and tier 1 high-confidence SVs set as well as human genome hg19 were used as ground truth. PAV and SVIM-asm were run with default parameters. For vg-vcfwave, the pipeline was initialized with 'vg deconstruct' to produce a VCF file recording the snarl and path information. Vcfbub and vcfwave were then executed following the released script by HPRC. For CSV detection, the same ground-truth dataset used in the SVision-pro study was adopted. Swave called CSVs from the pangenome graph built with human reference hg38 and template genome with all CSVs inserted. The assembly-based SVision-pro callset was obtained by applying SVision-pro on the alignment results between temple genome and hg38 reference genome. The read-based SVision-pro callset was obtained from its publication. For both SSVs and CSVs, performance was assessed using Truvari, calculating the F1-score against the corresponding ground truth.

Trio-level benchmark. Pangenome graphs were constructed for each trio along with human reference genome T2Tv2.0. All merge tools (Jasmine, SURVIVOR and Truvari), and the refinement tool PanPop, were executed following their official instruction to obtain the integrated trio callset including genotypes for each trio member. Swave and vg-vcfwave could directly output the callset comprising genotypes for each individual. To calculate the Mendelian consistency, we first collected the genotypes of parents and generated the list of all possible child genotypes. If the actual child genotype matched with anyone in the list, we determined it as a consistent one.

Population-level benchmark. To assess genotyping accuracy at the population level, pangenome graphs were built using reference genome T2Tv2.0 and 130 haplotypes from HGSVC. Callers were executed as the same as in the trio-level benchmark. Genotype completeness was evaluated at the haplotype level by measuring the proportion of missing haplotype (denoted as '.') across all SV records.

Computational inversion validation. To computationally validate inversions, we reconstructed the inversion-feature sequence by reversing the reference corresponding to each detected inversion region. The sequence was then mapped to the carrier haplotype using mappy, a python interface to minimap2. Carrier haplotype sequence was reconstructed by concatenating sequences of nodes from the pangenome graph corresponding to the haplotype path. If the detected inversion is true positive, the inversion-feature sequence would forwardly and continuously map to the carrier haplotype sequence with no clips. Therefore, mapping integrity was defined as the ratio of aligned length to total inversion-feature length. We also leveraged the validation metrics from two published tools, TT-mars and Vapor. Both the tools measure whether the inversion-feature sequence exhibits a better match to the carrier haplotype than the wild-type reference sequence, with TT-mars using an aligner (mappy) and Vapor using dotplot to get the mapping results. Their core validation modules were integrated into the overall mapping integrality framework mentioned above. For CSVs with multiple inversion subcomponents, each inversion substructure was validated independently.

Gene and repeat annotations. Gene annotation was performed using ANNOVAR[52] following the official instructions. For CSVs, each subcomponent was annotated separately. For duplications, both source and inserted regions were annotated. SVs intersecting 'exons' or 'splicing' annotations were flagged for downstream analysis. Repeat annotation

was performed using Tandem Repeat Finder[53] with recommended parameters on both the reference and alternative sequences. A repeat ratio was calculated for each allele as the fraction of repeat-annotated bases over total length. SVs were classified as 'highly repetitive regions' if the maximum repeat ratio of either the reference and alternative sequence annotation exceeded 80%.

Reporting summary

Further information on research design is available in the Nature Portfolio Reporting Summary linked to this article.

Data availability

All the published reference genomes, sample assemblies and SV callsets are presented in Supplementary Table 17. The callsets on healthy and disease cohorts produced by Swave are available via Zenodo at https://doi.org/10.5281/zenodo.18229680 and https://doi.org/10.5281/zenodo.18425621 (refs. 54,55).

Code availability

Swave is available via GitHub at https://github.com/songbowang125/Swave.git (ref. 56). The custom scripts and scripts for reproducing the results in this paper are available via GitHub at https://github.com/songbowang125/Swave-Utils.git (ref. 57).

References

50. Li, H. Minimap2: pairwise alignment for nucleotide sequences. *Bioinformatics* **34**, 3094–3100 (2018).
51. Vollger, M. R. et al. Improved assembly and variant detection of a haploid human genome using single-molecule, high-fidelity long reads. *Ann. Hum.Genet.* **84**, 125–140 (2020).
52. Wang, K., Li, M. & Hakonarson, H. ANNOVAR: functional annotation of genetic variants from high-throughput sequencing data. *Nucleic Acids Res.* **38**, e164 (2010).
53. Benson, G. Tandem repeats finder: a program to analyze DNA sequences. *Nucleic Acids Res.* **27**, 573–580 (1999).
54. Wang, S. Swave call on healthy cohort. *Zenodo* https://doi.org/10.5281/zenodo.18229680 (2026).
55. Wang, S. Swave call on disease cohort. *Zenodo* https://doi.org/10.5281/zenodo.18425621 (2026).
56. Wang, S. Swave code. *Zenodo* https://doi.org/10.5281/zenodo.18229263 (2026).
57. Wang, S. Swave Utils code. *Zenodo* https://doi.org/10.5281/zenodo.18229275 (2026).

Acknowledgements

K.Y. is supported by the National Key R&D Program of China (grant no. 2022YFC3400300) and National Science Foundation of China (grant nos. 32125009 and 32430017). S.W. is supported by the National Science Foundation of China (grant no. 323B2015)

Author contributions

K.Y. designed and supervised the research. S.W. developed the algorithm and performed the performance evaluation and downstream analysis. T.X. and P.Z. analyzed the impact of SVs.

Competing interests

The authors declare no competing interests.

Additional information

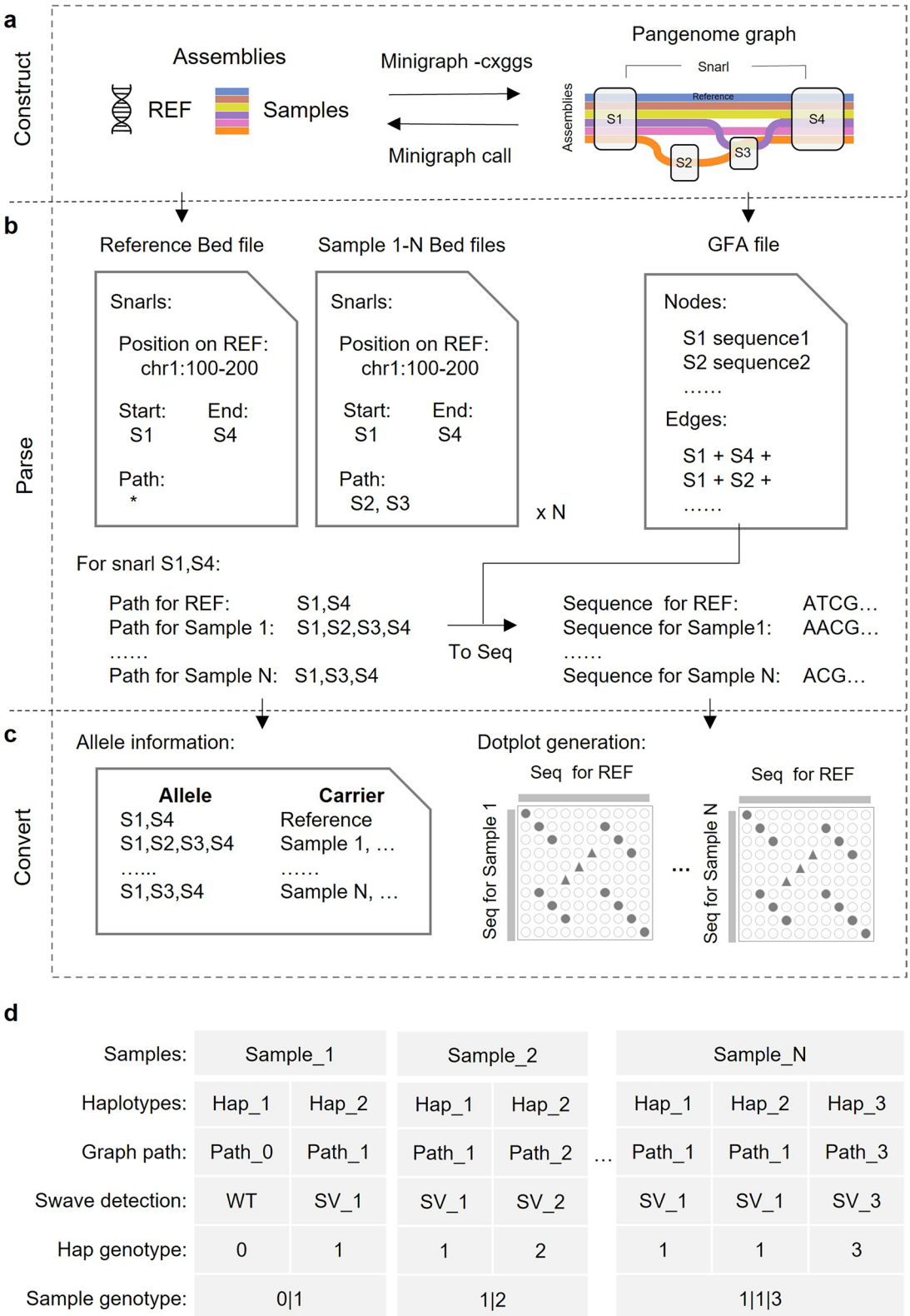
Extended data is available for this paper at https://doi.org/10.1038/s41588-026-02538-6.

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41588-026-02538-6.

Correspondence and requests for materials should be addressed to Kai Ye.

Peer review information *Nature Genetics* thanks the anonymous reviewer(s) for their contribution to the peer review of this work. Peer reviewer reports are available.

https://doi.org/10.1038/s41588-026-02538-6

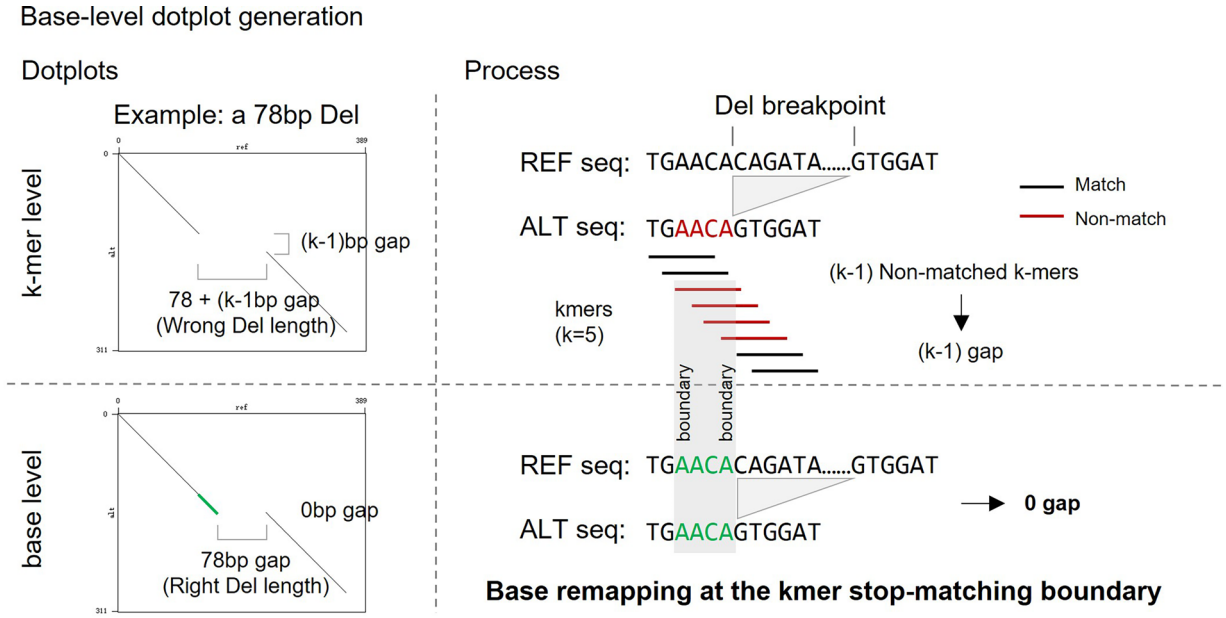


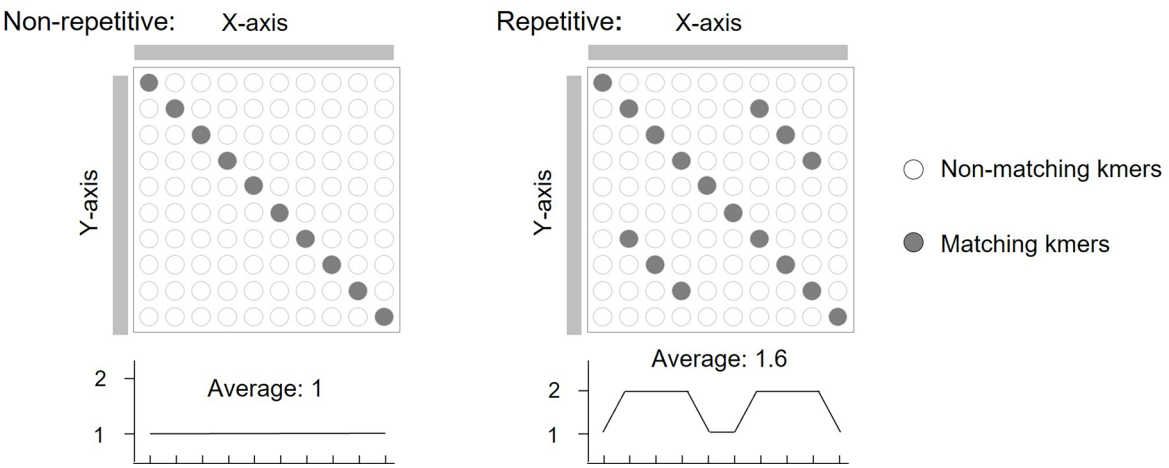
d

Samples:	Sample_1		Sample_2		…	Sample_N		
Haplotypes:	Hap_1	Hap_2	Hap_1	Hap_2		Hap_1	Hap_2	Hap_3
Graph path:	Path_0	Path_1	Path_1	Path_2		Path_1	Path_1	Path_3
Swave detection:	WT	SV_1	SV_1	SV_2		SV_1	SV_1	SV_3
Hap genotype:	0	1	1	2		1	1	3
Sample genotype:	0|1		1|2			1|1|3		

Extended Data Fig. 1 | Overview of pangenome construction and allele extraction in Swave. a, Construction of pangenome graph using Minigraph with both reference and sample assemblies. The resulting graph is saved in GFA format, which encodes node sequences and directed edges between nodes. **b**, Assembly paths are recovered using –call function in Minigraph. Regions where paths diverge (Snarls) are identified as candidate structural variant loci. Allele sequences for each snarl are reconstructed by extracting the corresponding node sequences from the GFA. **c**, Based on the Minigraph–all outputs, Swave determines carrier assemblies for each allele and proceeds to generate dotplots for each reference-alternative pair in the next processing module. **d**, Swave's handling of phasing information and multi-allelic loci. Sample genotypes are obtained by joining all haplotype genotypes.

https://doi.org/10.1038/s41588-026-02538-6



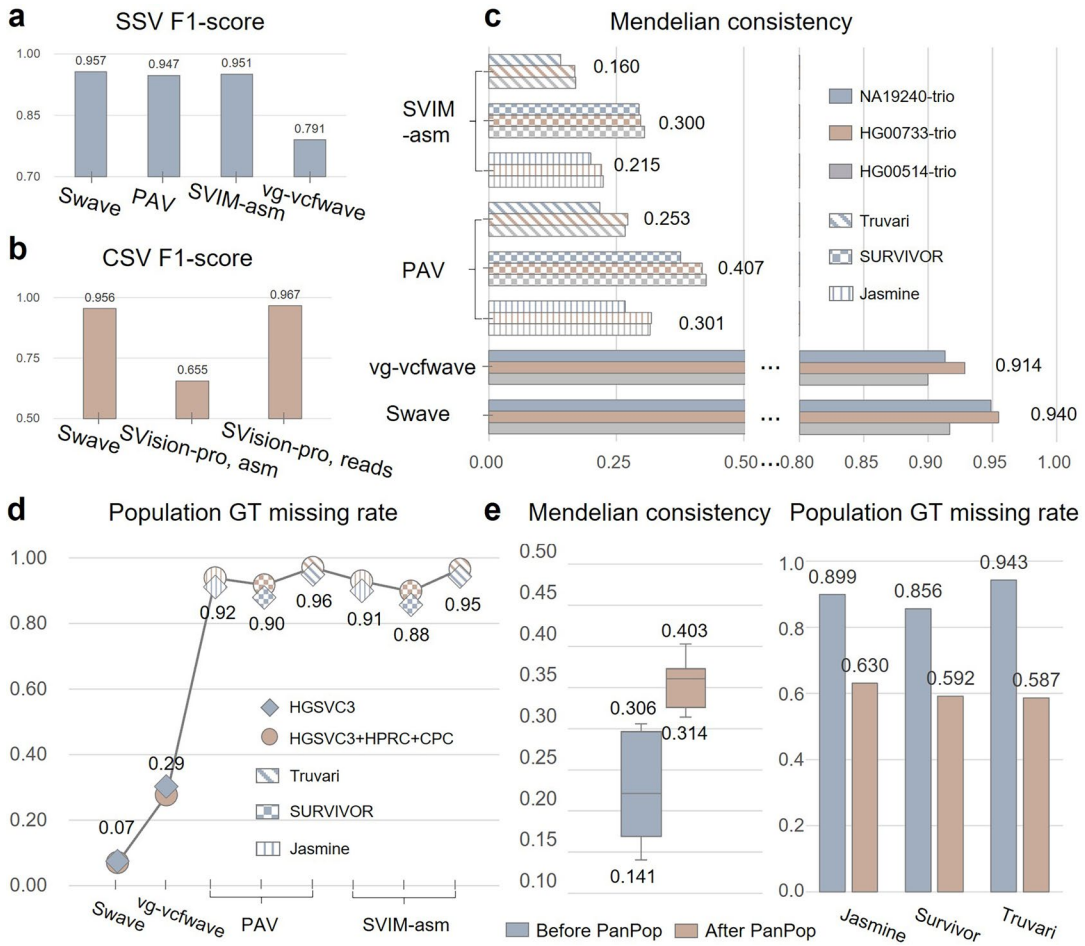


Extended Data Fig. 2 | Dotplot generation and projection. a, Base-level refinement of kmer-based dotplots. Initial alignment introduces (k-1) base gaps near SV breakpoints. Swave performs base-level remapping at kmer stop-matching boundaries to improve breakpoint resolution for downstream SV classification. **b**, Influence of genomic repeats on wave patterns. Dense, repetitive regions generate abundant spurious matches in dotplots, resulting fluctuating wave signals upon projection.

https://doi.org/10.1038/s41588-026-02538-6

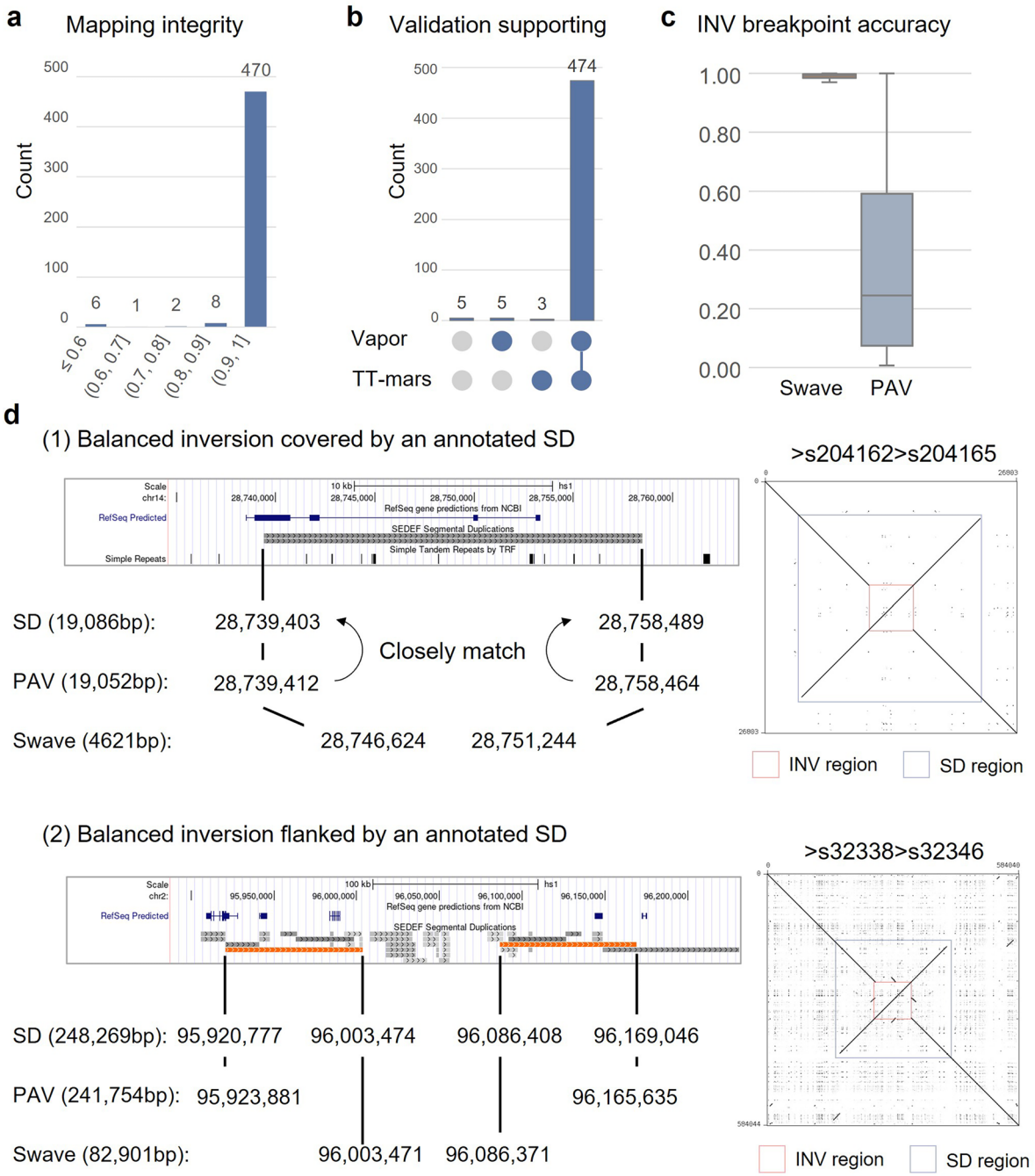
Extended Data Fig. 3 | Recurrent Neural Network for SV classification in Swave. **a**, Projected wave signals are encoded as four-element tuples per genomic segment, comprising span length, background average wave value, and the differences between SV-implying and background waves for both forward and reverse matches. These tuples serve as the input for the RNN classification model. **b**, A one-layer Bi-LSTM with 64 hidden units forms the core of the RNN, enabling context-aware classification of SV components across the sequence. **c**, The time and memory consumption were performed using three datasets, including HGSVC3 (130 haplotypes), Health cohort (HGSVC3 + HPRC + CPC, 334 haplotypes) and Disease cohort (GA4K, 574 haplotypes). Using a person computer (CPU: Intel Core i9-13900K, Max Memory: 32GB), Swave run with 8 threads. Using a computing cluster node (CPU: Intel Xeon Gold 6240 R, Max Memory: 376GB), the computational process of Swave could be accelerated by using 24 threads.

https://doi.org/10.1038/s41588-026-02538-6



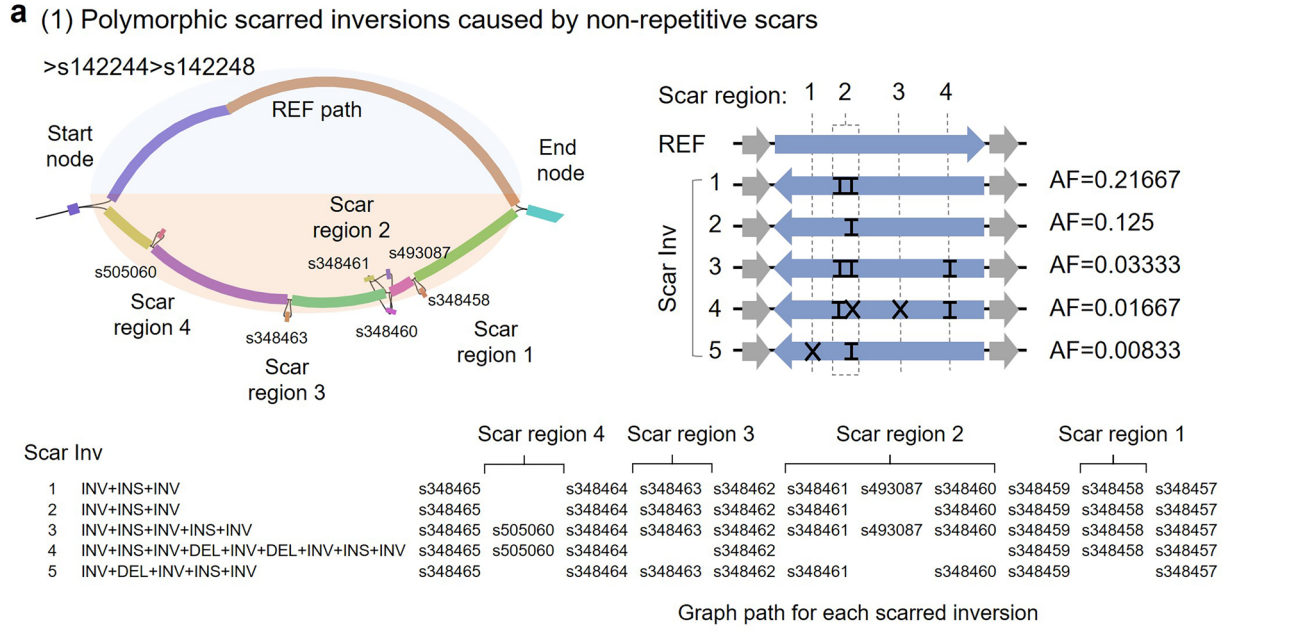
Extended Data Fig. 4 | Performance evaluation results. a, F1-score comparison for simple structural variant (SSV) detection. **b**, F1-score comparison for complex structural variant (CSV) detection. **c**, Mendelian consistency across three trio datesets. Average consistencys are noted on the plot. **d**, Genotyping (GT) missing rate across two population datasets. Average missing rates on the two datasets are noted on the plot. **e**, Improvements in genotyping performance following PanPop refinement. We applied SVIM-asm followed by 3 merging tools on 3 trios, respectively, making n = 9. The boxplot defines the median (Q2, 50th percentile), first quartile (Q1, 25th percentile) and third quartile (Q3, 75th percentile). The bounds of box, that is interquartile range (IQR), of the boxplot is between Q1 and Q3. The minima and maxima values are defined as Q1-1.5*IQR and Q3 + 1.5*IQR, respectively. The whiskers are values between minima and Q1 as well as between Q3 and maxima. Values falling outside the Q1 – Q3 range are plotted as outliers of the data.

https://doi.org/10.1038/s41588-026-02538-6



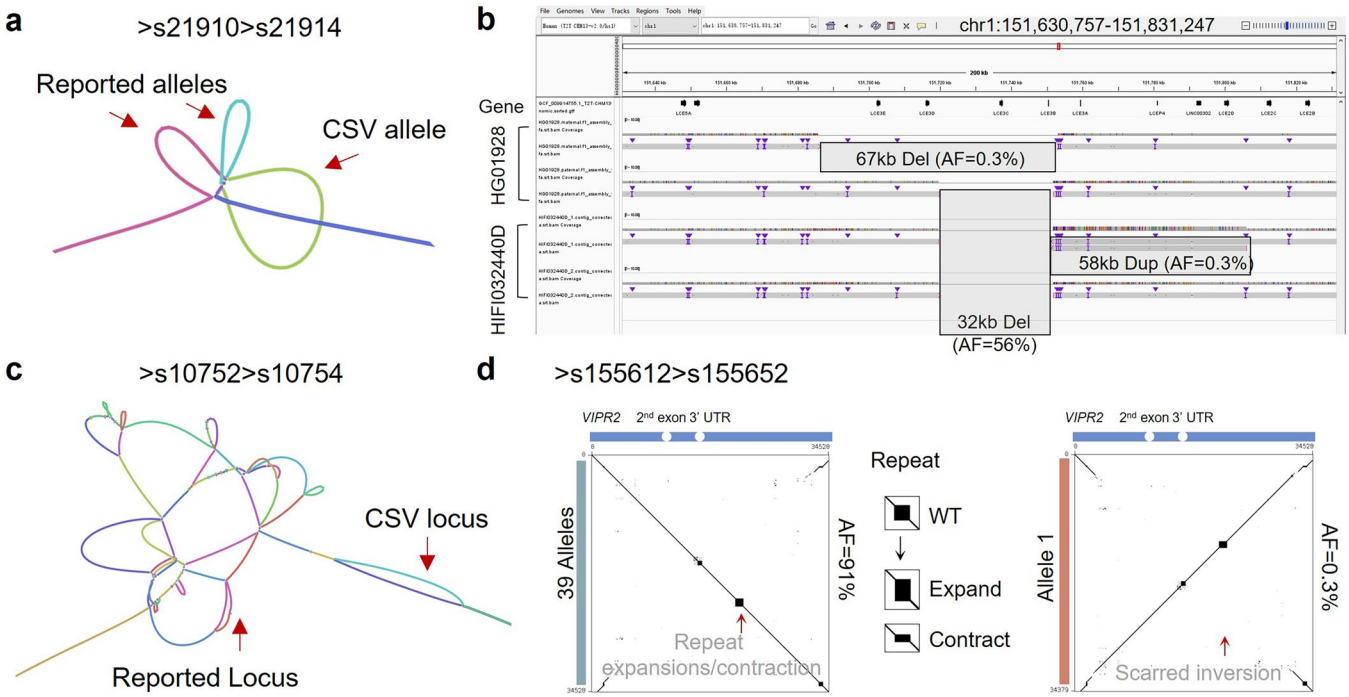
Extended Data Fig. 5 | Validation and illustration of inversions. a and **b**, Validation results for all detected balanced and complex inversions. Three orthogonal metrics were applied: mapping integrity, TT-mars, and Vapor. **c**, Comparison of breakpoint accuracy of the 52 overlapped inversions between Swave and PAV. The boxplot defines the median (Q2, 50th percentile), first quartile (Q1, 25th percentile) and third quartile (Q3, 75th percentile). The bounds of box, that is interquartile range (IQR), of the boxplot is between Q1 and Q3. The minima and maxima values are defined as Q1-1.5*IQR and Q3 + 1.5*IQR, respectively. The whiskers are values between minima and Q1 as well as between Q3 and maxima. Values falling outside the Q1 – Q3 range are plotted as outliers of the data. **d**, Illustration of breakpoint distortion caused by inverted segmental duplications (SDs). While PAV's breakpoints are frequently shifted due to alignment ambiguity, Swave maintains accurate breakpoint placement within repetitive regions.

https://doi.org/10.1038/s41588-026-02538-6



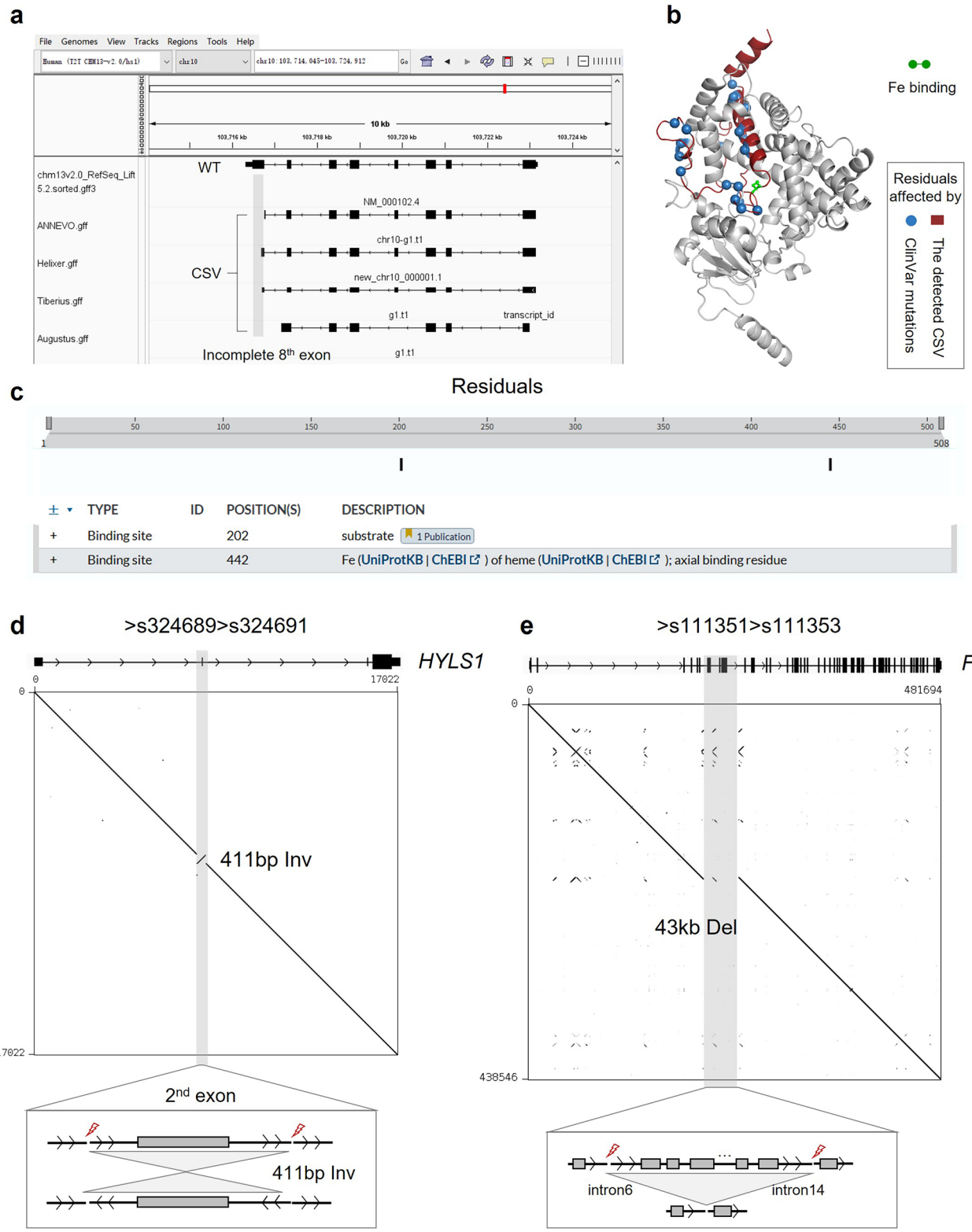
Extended Data Fig. 6 | Characterization of polymorphic scarred inversions. a, Example of a polymorphic scarred inversion snarl containing five distinct alleles (1), generated by combinatorial arrangements of five unique internal scars across four genomic regions. The most complex variant includes four separate scars (2). **b**, Length distribution of all detected scars (n = 81), ranging from 61 bp to 18,451 bp. **c**, Repeat annotation of all scars (n = 81). **d**, Example of polymorphic scarred inversions driven by repetitive elements, where two repeat expansions give rise to insertion scars of difference lengths.

https://doi.org/10.1038/s41588-026-02538-6



Extended Data Fig. 7 | Rare and complex structural variants revealed by Swave. a, Pangenome graph structure of snarl '>s21910 > s21914'. A rare CSV allele introduced a novel traversal path not observed among the reported alleles. **b**, IGV snapshot of snarl '>s21910 > s21914', illustrating co-occurrence of two distinct SSVs and one CSV. The rare SSV (67 kb deletion) extended the common 32 bp deletion, where the rare CSV (a duplication flanked by a deletion) added a 58 kb duplication at the right breakpoint of the frequent 32 kb simple deletion. **c**, Pangenome graph of snarl '>s10752 > s10754', showing a novel CSV locus, structurally distinct from previously reported variants. **d**, Illustration of a rare scarred inversion that partially disrupts the coding structure of *VIPR2*, a gene associated with neuropsychiatric disorder.

https://doi.org/10.1038/s41588-026-02538-6



Extended Data Fig. 8 | How potentially pathogenic structural variants affect genes. a, Mapping of residue-level disruptions caused by ClinVar pathogenic variants and the CSV detected by Swave. **b**, Structural annotation of the CYP17A1 protein highlights two functional binding sites, as sourced from UniProt. **c** Mapping of residue-level disruptions caused by ClinVar pathogenic variants and the CSV detected by Swave. **d**, Schematic of a simple structural variant, a 411 bp inversion, disrupting the 2nd exon of gene *HYLS1*, a gene implicated in Hydrolethalus Syndrome. **e**, Representation of a 43 kb deletion spanning introns 6 to 14 of gene *FRAS1*, a gene associated with Fraser syndrome.

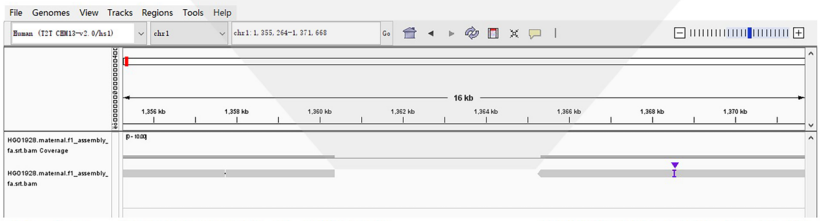
https://doi.org/10.1038/s41588-026-02538-6

a Four snarls with missing GT:

chrom	start	end	Snarl		
chr1	1360670	1360670	>s916	>s917	.
chr1	1362767	1362833	>s917	>s919	.
chr1	1364471	1364523	>s919	>s921	.
chr1	1364798	1364838	>s921	>s924	.

Mapping result:

No mapping

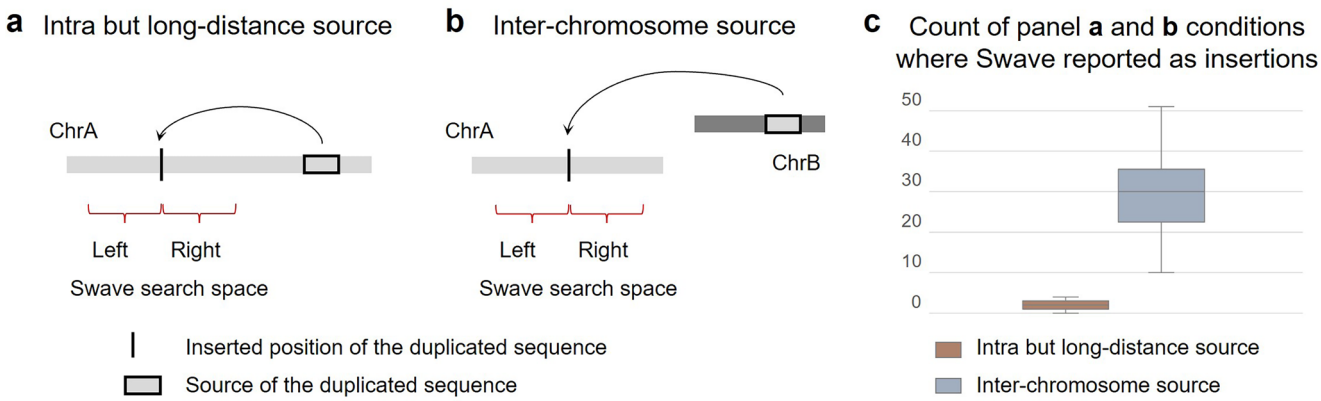


Contig: HG01928#2#h2tg000161l HG01928#2#h2tg000183l

b The distribution of snarls with missing genotypes

Extended Data Fig. 9 | Genotyping incompleteness associated with unresolved pangenome graph regions. a, Mapping results of a carrier assembly exhibiting missing genotypes across four snarls. **b**, Genome-wide distribution of snarls with missing genotypes across HGSVC samples. The Y-axis indicates the number of assemblies lacking mappable sequence at each snarl.

https://doi.org/10.1038/s41588-026-02538-6



Extended Data Fig. 10 | Misclassified dispersed-duplications into insertions. Dispersed duplications—where the source sequence originates from distant loci (**a**) on the same chromosome or from different chromosomes (**b**)—pose challenges for Swave. When generating dotplots, Swave extends the reference regions by twice the length of the alternative sequence on both sides. Consequently, if the duplicated source sequence lies outside this extended window, Swave fails to capture it and instead reports it as an insertion. **c**, Using the 65 samples from HGSVC, we compared Swave's outputs with dispersed duplications reported by SVision-pro. We found that Swave misclassified 0–4 duplications with distant same-chromosome sources and 10–51 duplications with cross-chromosome sources as insertions. The boxplot defines the median (Q2, 50th percentile), first quartile (Q1, 25th percentile) and third quartile (Q3, 75th percentile). The bounds of box, that is interquartile range (IQR), of the boxplot is between Q1 and Q3. The minima and maxima values are defined as Q1-1.5*IQR and Q3 + 1.5*IQR, respectively. The whiskers are values between minima and Q1 as well as between Q3 and maxima. Values falling outside the Q1 – Q3 range are plotted as outliers of the data.

nature portfolio

Corresponding author(s): Kai Ye

Last updated by author(s): Jan 13, 2026

Reporting Summary

Nature Portfolio wishes to improve the reproducibility of the work that we publish. This form provides structure for consistency and transparency in reporting. For further information on Nature Portfolio policies, see our Editorial Policies and the Editorial Policy Checklist.

Statistics

For all statistical analyses, confirm that the following items are present in the figure legend, table legend, main text, or Methods section.

n/a	Confirmed
☐	☒ The exact sample size (*n*) for each experimental group/condition, given as a discrete number and unit of measurement
☒	☐ A statement on whether measurements were taken from distinct samples or whether the same sample was measured repeatedly
☐	☒ The statistical test(s) used AND whether they are one- or two-sided *Only common tests should be described solely by name; describe more complex techniques in the Methods section.*
☒	☐ A description of all covariates tested
☒	☐ A description of any assumptions or corrections, such as tests of normality and adjustment for multiple comparisons
☐	☒ A full description of the statistical parameters including central tendency (e.g. means) or other basic estimates (e.g. regression coefficient) AND variation (e.g. standard deviation) or associated estimates of uncertainty (e.g. confidence intervals)
☐	☒ For null hypothesis testing, the test statistic (e.g. *F*, *t*, *r*) with confidence intervals, effect sizes, degrees of freedom and *P* value noted *Give P values as exact values whenever suitable.*
☒	☐ For Bayesian analysis, information on the choice of priors and Markov chain Monte Carlo settings
☒	☐ For hierarchical and complex designs, identification of the appropriate level for tests and full reporting of outcomes
☐	☒ Estimates of effect sizes (e.g. Cohen's *d*, Pearson's *r*), indicating how they were calculated

Our web collection on statistics for biologists contains articles on many of the points above.

Software and code

Policy information about availability of computer code

Data collection	wget (GNU v1.14) was used to download all published datasets from their official ftps.
Data analysis	Pangenome graph construction: Minigraph (0.20-r559) Assembly alignment: Minimap2 (2.17-r941) Structural variant calling: Swave (1.0), PAV (2.3.4), SVIM-asm (1.0.3), vg (1.50.1), vcfbub (0.1.0), vcfwave (latest), SVision-pro (v2.5), Sniffles2 (v2.6.3), debreak (v1.0.2), cuteSV (v2.1.2), cue (version cue2, latest), pbsv (2.11.0) Structural variant mering: Jasmine (1.1.5), SURVIVOR (1.0.7), Truvari (3.5.0), PanPop (NC2024) Structural variant annotation: Annovar (2022-08-02 22:48:58 -0400), TRF (trf409.linux64) The scripts for reproducing the results in this paper are available at GitHub (https://github.com/songbowang125/Swave-Utils.git)

For manuscripts utilizing custom algorithms or software that are central to the research but not yet described in published literature, software must be made available to editors and reviewers. We strongly encourage code deposition in a community repository (e.g. GitHub). See the Nature Portfolio guidelines for submitting code & software for further information.

Data

Policy information about availability of data

All manuscripts must include a data availability statement. This statement should provide the following information, where applicable:

- Accession codes, unique identifiers, or web links for publicly available datasets
- A description of any restrictions on data availability
- For clinical datasets or third party data, please ensure that the statement adheres to our policy

All the published reference genomes, sample assemblies, and SV callsets are listed in Supplementary Table 17. The callsets on healthy and disease cohorts produced by Swave are shared in Zenodo

Research involving human participants, their data, or biological material

Policy information about studies with human participants or human data. See also policy information about sex, gender (identity/presentation), and sexual orientation and race, ethnicity and racism.

Reporting on sex and gender	Not such results
Reporting on race, ethnicity, or other socially relevant groupings	Not such results
Population characteristics	Not such results
Recruitment	Not relevant
Ethics oversight	Not relevant

Note that full information on the approval of the study protocol must also be provided in the manuscript.

Field-specific reporting

Please select the one below that is the best fit for your research. If you are not sure, read the appropriate sections before making your selection.

☒ Life sciences ☐ Behavioural & social sciences ☐ Ecological, evolutionary & environmental sciences

For a reference copy of the document with all sections, see nature.com/documents/nr-reporting-summary-flat.pdf

Life sciences study design

All studies must disclose on these points even when the disclosure is negative.

Sample size	For the population-level structural variant detection, we collected public assemblies, including HGSVC (130 haplotypes), HPRC (88 haplotypes) and CPC (116 haplotypes). The number was determined by the accessible and published data number from the three flagship human projects. The number was sufficient for evaluation Swave due to they have high-quality genome assemblies and pangenome graphs.
Data exclusions	Six haplotypes from three samples in HPRC were excluded due to the duplicated existence in HGSVC
Replication	Replication was not relevant to our study. This study used deterministic algorithms without statistical analysis, and this study aims to demonstrate Swave and its application to structural variant detection with pangenome graph.
Randomization	Randomization was not relevant to our study. Swave is a deterministic method. and all analysis in this study was done with preexisting data sources.
Blinding	Blinding was not relevant to our study. We used publicly available data, no data acquisition or statistical analysis was involved

Reporting for specific materials, systems and methods

We require information from authors about some types of materials, experimental systems and methods used in many studies. Here, indicate whether each material, system or method listed is relevant to your study. If you are not sure if a list item applies to your research, read the appropriate section before selecting a response.

Materials & experimental systems

n/a	Involved in the study
☒	☐ Antibodies
☒	☐ Eukaryotic cell lines
☒	☐ Palaeontology and archaeology
☒	☐ Animals and other organisms
☒	☐ Clinical data
☒	☐ Dual use research of concern
☒	☐ Plants

Methods

n/a	Involved in the study
☒	☐ ChIP-seq
☒	☐ Flow cytometry
☒	☐ MRI-based neuroimaging

Plants

Seed stocks	Not relevant to our study
Novel plant genotypes	Not relevant to our study
Authentication	Not relevant to our study