ARCLID: Accurate and Robust Characterization of Long Insertions and Deletions in Genome

Sajad Tavakoli[1], Rasmus John Normand Frandsen[1], Marjan Mansourvar[1]*

[1]Department of Biotechnology and Biomedicine (DTU Bioengineering), Technical University of Denmark, Søltofts Plads, Kongens Lyngby, 2800, Denmark

*Corresponding Author: marjma@dtu.dk

Abstract:

Accurate detection of genomic structural variants (SVs) remains challenging, particularly for large events and at low sequencing coverage. We present ARCLID, a novel deep learning-based SV caller for PacBio HiFi data that treats SVs as objects in pileup images. Evaluated on diverse real and synthetic datasets across varying coverage levels, ARCLID demonstrates reliable and consistent accuracy, even for SVs larger than 1 kbp. Crucially, it maintains high performance on samples with lower sequencing depths (10X, 5X). This capability to preserve accuracy at lower depths provides a significant practical advantage for cost-constrained projects, enabling robust SV discovery without requiring deep sequencing. ARCLID represents a promising advancement toward more accessible and efficient long-read SV analysis for diverse research applications.

Keywords: Structural Variant; Long Reads; Sequencing Depth; Deep Learning; YOLO

1. Introduction

Structural variants (SVs) are large-scale genomic alterations in the human genome, including insertions, deletions, duplications, inversions, translocations, etc. that can significantly influence gene regulation, function, and ultimately phenotypic diversity [1], [2]. These variants are also increasingly recognized as key contributors to human disease, with SVs having been linked to conditions such as Huntington's disease, driven by pathogenic CAG repeat expansions in the HTT gene [3], and 22q11.2 deletion syndrome (DiGeorge syndrome), caused by recurrent deletions on chromosome 22 [4]. Among SVs, insertions and deletions (indels) are the most frequent forms and have been implicated in diverse genetic disorders [5] and evolutionary genomics [6]. Despite the significant biological insights that SV analysis can offer, accurate detection and genotyping remain challenging due to the complex, heterogeneous nature of genomic data [7]. In recent years, advances in sequencing technologies have spurred the development of numerous computational methods for SV identification. However, many of these tools still face limitations in accuracy, scalability, and robustness when applied across varying sequencing coverages and SV sizes.

Detection of SVs from genomic sequencing data has traditionally followed two main directions: assembly-based and read alignment–based approaches [8]. Assembly-based methods reconstruct long contiguous sequences (contigs) from raw reads and then compare the assembled genome to a reference to identify SVs. This strategy can better capture complex regions like MHC, enabling precise detection of large and complex SVs, especially in repetitive

or challenging genomic regions [8]. However, assembly-based approaches require substantial computational resources and depend on high sequencing coverage [8]. In contrast, read alignment-based SV calling detects SVs directly from sequencing reads mapped to a reference genome, making it more practical for routine applications and lower coverage data [8]. However, the approach struggles to identify large SVs.

Read alignment-based SV calling, especially from long reads, typically follows a multi-step process. First, sequencing reads are aligned to a reference genome using long-read mappers such as minimap2 [9] or pbmm2 (a minimap2 wrapper for PacBio data) which provide detailed information on mismatches, gaps, and breakpoints. From these alignments, characteristic signals are extracted, including split reads [10], discordant read pairs [7], local shifts in read depth[11], and large gaps in CIGAR strings (summarizing the alignment information) [12], each of which may indicate the presence of an SV. These raw signals are then aggregated through clustering; grouping reads that support the same putative variant and at the same time filtering out noise from sequencing errors. The clustered evidence is refined to improve breakpoint resolution, often by reanalyzing supporting reads or constructing consensus sequences. Candidate variants are subsequently classified into standard SV types. Finally, genotypes are assigned by comparing the proportion of reads supporting the variant versus the reference allele, allowing the determination of heterozygous or homozygous states for the individual SV. This traditional approach, employed by tools such as Sniffles [13], Sniffles2 [14], cuteSV [15], and SVIM [16], and pbsv [17], is highly dependent on clustering thresholds and hand-crafted features/rules for integrating evidence.

These SV callers detect structural variants by searching for intra-alignment and inter-alignment signatures in aligned reads. Where intra-alignment signatures correspond to gaps in either the read or the reference, as indicated by the CIGAR string, and the inter-alignment signatures reflect discordances between segments of a split read [16]. In these tools, feature extraction is generally static and platform-specific, requiring manual parameter tuning to accommodate differences in sequencing technologies (e.g., Nanopore vs. PacBio) or sequencing coverage. Furthermore, multiple evidence sources, such as split-read patterns, coverage depth, etc., are often integrated in a sequential or additive manner, potentially limiting flexibility in the analysis. Sniffles2[14], the latest version of Sniffles, builds on these traditional steps but incorporates a novel algorithm designed to automatically adjust parameters, thereby alleviating the need for manual tuning [14].

In this study, where we develop ARCLID, we test a new approach to SV calling based on the deep learning paradigm, using a one-stage convolutional neural network to analyze pileup images of reads. In the new approach, the model automatically learns to recognize complex patterns from the data, thereby mitigating the need for manual feature engineering. ARCLID simultaneously localizes SV breakpoints and classifies each variant's type as well as genotype in a single inference pass, providing an integrated solution to SV calling.

ARCLID's workflow comprises five sequential steps. First, aligned reads, provided in BAM or CRAM format, are received and preprocessed (converting to numerical values, filtering, etc.) to fit the ARCLID data structure. Next, key meaningful features are extracted and encoded

into pileup images. These images are then processed in parallel by two YOLOv11x-based convolutional networks [18]: one optimized for variants smaller than 2 kb (the images that cover 5K bp) and the other for variants larger than 2 kb (the images that cover 50K bp). Each network simultaneously localizes breakpoints, classifies SVs, and genotypes them. Finally, the models' outputs are post-processed to remove likely false positives and the results are converted into a standard VCF file. To ensure robustness across both sequencing depth and variant size, the deep learning models were trained on datasets subsampled to 28×, 10×, and 5× coverage. Figure 1 illustrates the complete ARCLID pipeline.

In the following sections, we elaborate on the design of our method, describe the implementation of each component of our pipeline, and provide a comprehensive evaluation of its performance relative to existing tools. We also discuss the implications of our findings for future research in structural variant analysis and outline potential avenues for extending this work to include other types of genomic variants.

Results

Benchmarking Strategy and Datasets

We benchmarked ARCLID alongside well-established SV callers, Sniffles2, CuteSV, SVIM, and pbsv, using PacBio HiFi data from HG002 (released by GIAB) [19] , Simulated HiFi data by VISOR [20], as well as HG00733 and NA19240 (from HGSVC) [21]. It is worth mentioning that Chromosomes 1-10 of HG002 were used to train the models, chromosome 11 for internal evaluation, and chromosomes 12-22 for benchmarking. The GIAB dataset provides high-confidence regions and includes ~12,000 long insertions and deletions, making it a reliable standard for evaluating variant calling performance. Furthermore, simulated HiFi data containing synthetic variants provides a reliable resource for benchmarking, as these datasets are generated based on pre-defined variant sets, where the exact type and location of each variant are already known. As a result, the ground truth is unambiguous, allowing for precise evaluation of variant calling methods without the uncertainty and noise that may be present in real sequencing data. In contrast, the HGSVC dataset is more suitable for discovery, lacking validated regions (high-confidence regions like HG002 in GIAB) [19], [21] but containing over 25,000 SVs per sample [21]. We used the HG002 and simulated sample for evaluation of ARCLID's performance, while HG00733 and NA19240 allowed us to assess the new tool's generalizability.

Benchmarking Modes and Metrics

To benchmark ARCLID and other available SV callers, we employed Truvari [22] in two evaluation modes. In the strict mode, we required highly stringent matching criteria with parameters set to $ref_dist = 50\,bp$, $pctsim = 0.9$, and $pctovl = 0.9$, ensuring precise breakpoint agreement and allele similarity between predicted and reference variants. In the relaxed mode, we applied more tolerant criteria with $ref_dist = 1\,Kbp$, $pctsim = 0.7$, and $pctovl = 0$, allowing for approximate matches and broader overlap tolerance. This dual approach enabled us to assess performance under exact-match conditions as well as under more

flexible benchmarking conditions typically used in other studies, where slight differences in breakpoints or allele similarity are still considered acceptable.

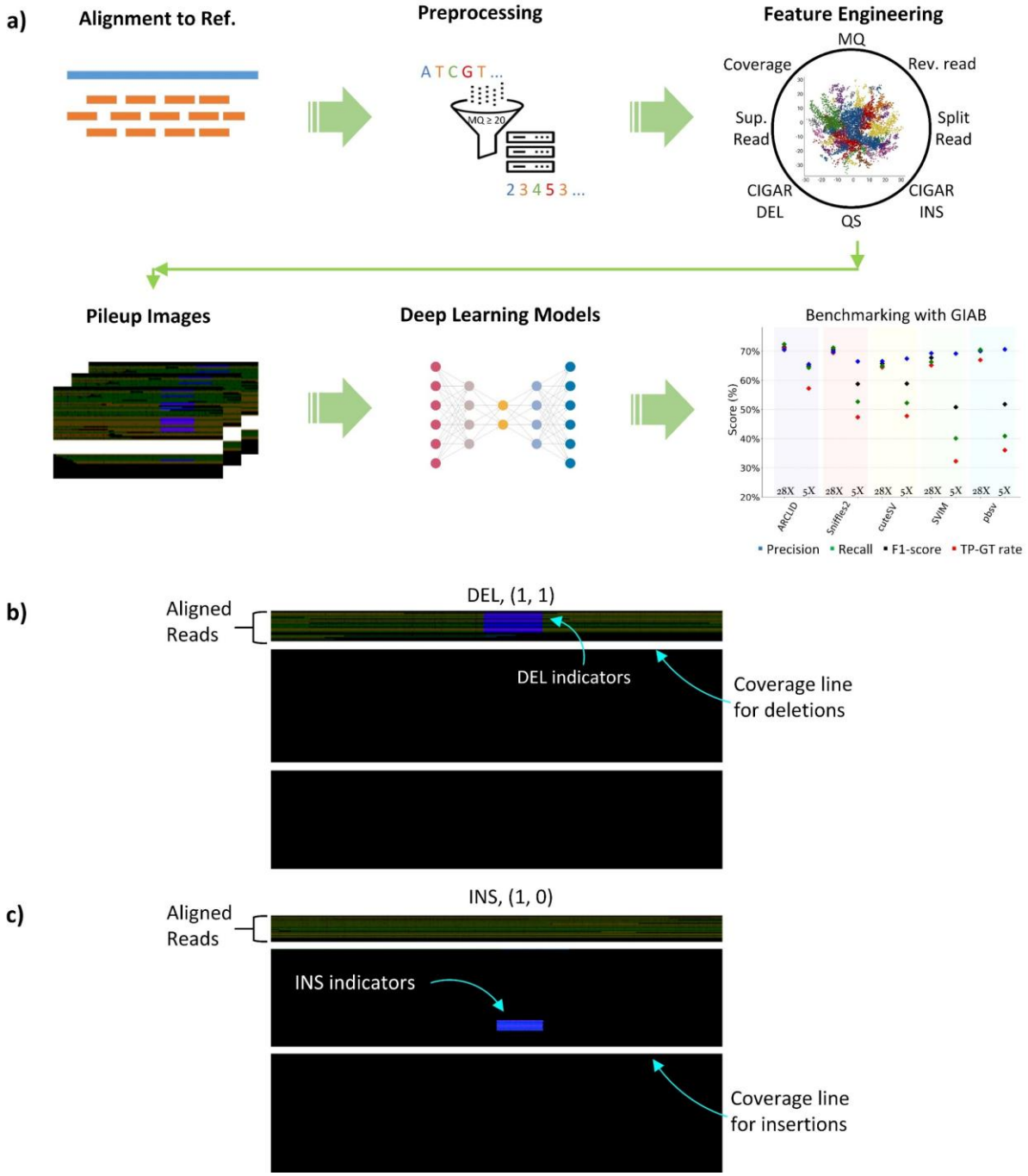

Figure 1. a) The workflow of ARCLID and performance evaluation in strict mode. In the preprocessing step, aligned sequences are converted to numerical values, and reads with mapping quality lower than 20 are filtered out. Then, meaningful features and signatures are extracted from alignment information, and these features are converted to pileup images. After that, the pileup images are injected into two YOLOv11x. One model identifies SVs smaller than 2Kbp, and another model identifies SVs larger than 2Kbp. The scatter plot (second row of images to the right) compares ARCLID's performance with other tools. When coverage is reduced from 28× to 5×, ARCLID's performance shows only a minimal drop across all metrics, with the values remaining close to each other. In contrast, the metrics for other tools exhibit a large divergence. **b) and c)** Two examples of pileup images for insertions and deletions. b) shows a DEL (deletions) with the genotype '1/1' that has been correctly characterized, while c) shows an INS (insertions) with genotype of '0/1.' These two pileup images are from HG002 GIAB benchmarking, chromosome 11, 28X coverage. The pileup images are able to support samples up to 150X coverage.

We used four metrics to capture/describe the tools' performance which are precision (Equation 1), recall (Equation 2), F1-score (Equation 3), and true positive genotyping rate (TP-GT Rate, Equation 4). The TP-GT rate represents the proportion of SVs that are not only correctly detected and classified but also correctly genotyped, relative to the total number of SVs. Because it requires correct genotyping in addition to detection, this value is always lower than the recall, which considers only the fraction of SVs correctly identified regardless of genotype accuracy. To keep the paper short and concise, only the evaluations in strict mode are tabulated and demonstrated, however, the full results for both modes across all coverages and SV sizes are available in Supplementary Tables (Tables1-14).

$$Prec. = \frac{TP}{TP+FP} \qquad \text{Equation (1)}$$

$$rec. = \frac{TP}{TP+FN} \qquad \text{Equation (2)}$$

$$F1 = \frac{2 \times prec. \times rec.}{prec. + rec.} \qquad \text{Equation (3)}$$

$$TP_GT\ rate = \frac{number\ of\ correctly\ classified\ and\ genotyped}{number\ of\ all\ SVs} \qquad \text{Equation (4)}$$

In the equations 1-4, TP (true positives) denotes the number of structural variants (SVs) correctly identified by the method, FP (false positives) represents the number of incorrectly predicted SVs that are not present in the ground truth, and FN (false negatives) refers to the number of SVs present in the ground truth but missed by the method. $Prec.$ (precision) measures the proportion of predicted SVs that are correct, while $rec.$ (recall) measures the proportion of true SVs that are correctly identified. The $F1\ score$ is the harmonic mean of precision and recall. The $TP_GT\ rate$ is defined as the proportion of SVs that are both correctly classified by type and correctly genotyped, relative to the total number of SVs in the ground truth.

Performance Across Coverage and SV Sizes

The evaluation of ARCLID's performance on HG002 chromosomes 12-22 (Figure 2 and Figure 3 along with Table 1 and 2), highlights the superior accuracy and robustness of ARCLID across varying coverages and variant size ranges. Given Figure 2, as coverage decreases, a noticeable decline in performance is observed for all tools. However, ARCLID demonstrates greater stability, with its F1-score decreasing modestly from 0.71 to 0.64, compared to Sniffles2, the second-best performer, whose F1-score drops from 0.70 to 0.58. In terms of robustness to variant size, ARCLID maintains relatively strong performance as variant lengths increase: in 28X coverage, by moving from 50–100 bp SV size to the 5–10 Kbp, its F1-score and true positive genotyping rate (TP-GT rate) decline from 0.74 and 0.75 to 0.44 and 0.42, respectively. In contrast, Sniffles2 exhibits a more significant drop, with F1-score and genotyping rate falling from 0.75 and 0.77 to 0.18 and 0.05, respectively.

Similar trends are seen for simulated data (Figure 4 and Table 3), where recall and F1-score consistently decrease as sequencing coverage drops from 25× to 5×. At higher coverages (25× and 15×), all tools perform relatively well, with recall values above 0.9 for most callers, leading to high F1-scores (>0.88). ARCLID, in particular, maintains a balanced precision–recall tradeoff, which results in stable F1-scores across different coverages. In contrast, tools such as SVIM and pbsv show more variability, with recall decreasing more sharply at lower coverage, which in turn drives their F1-scores down. At 10× coverage, the performance gap between callers becomes more pronounced, where ARCLID and sniffles2 still retain F1-scores around 0.85, while SVIM drops below 0.74. The trend is most striking at 5× coverage, where the recall of SVIM and cuteSV falls significantly, yielding F1-scores below 0.75 and even close to 0.5 in some cases. This demonstrates that ARCLID is more robust in low-coverage scenarios, while the other callers are more sensitive to reduced sequencing depth. It is also worth noting that in this table, we do not report the true positive genotyping rate, as the simulated variants generated by VISOR lack genotypic information. Therefore, the evaluation presented here focuses solely on precision, recall, and F1-score as the most reliable measures under these conditions. Moreover, since all simulated variants generated by VISOR were shorter than 500 bp, we do not compare the tools' performance across different variant sizes.

The analysis of HG00733 and NA19240 from the HGSVC dataset (Table 4) illustrates the generalizability of the tool's performance. However, these results are not intended for benchmarking purposes, as HGSVC lacks high-confidence regions and is primarily designed for variant discovery. Table 4 shows that ARCLID delivers reasonable performance, with the highest precision (0.7) for both HG00733 and NA19240 among all SV callers.

Resource Usage Evaluation

We benchmarked ARCLID against other SV callers in terms of wall-clock time, CPU time, and memory usage using the HG002 28X across different numbers of cores and threads. The evaluation was performed on a high-performance computing (HPC) system equipped with an NVIDIA V100 GPU, as ARCLID relies on two deep learning models and requires GPU acceleration. Under this setup, ARCLID completed SV calling in approximately 32 minutes with 4 cores and 4 threads. When restricted to a single core and single thread, the wall-clock time increased to about 200 minutes. In comparison, Sniffles2 did not finish even after 24 hours in the single core/thread setting. However, when running with 4, 8, or 16 cores and threads, Sniffles2 was fastest SV caller. Overall, ARCLID demonstrates efficient scalability across computational settings and provides a practical solution for researchers. Full evaluation results, including detailed resource usage, are reported in Supplementary Information, Table 1.

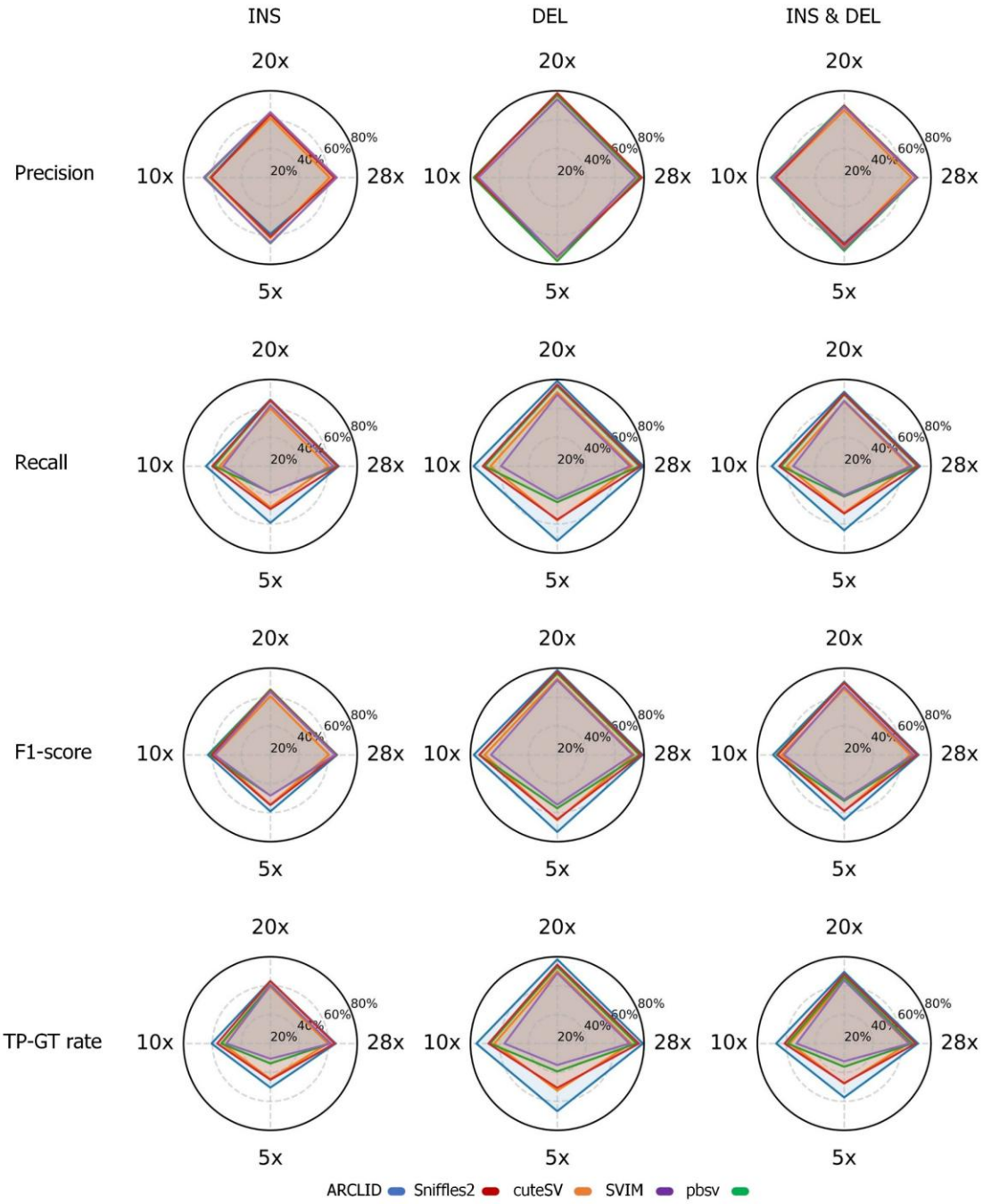

Figure 2. Evaluation of ARCLID's performance on the HG002 dataset across different coverages in strict mode for INS (first column), DEL (second column), and both INS and DEL (third column).

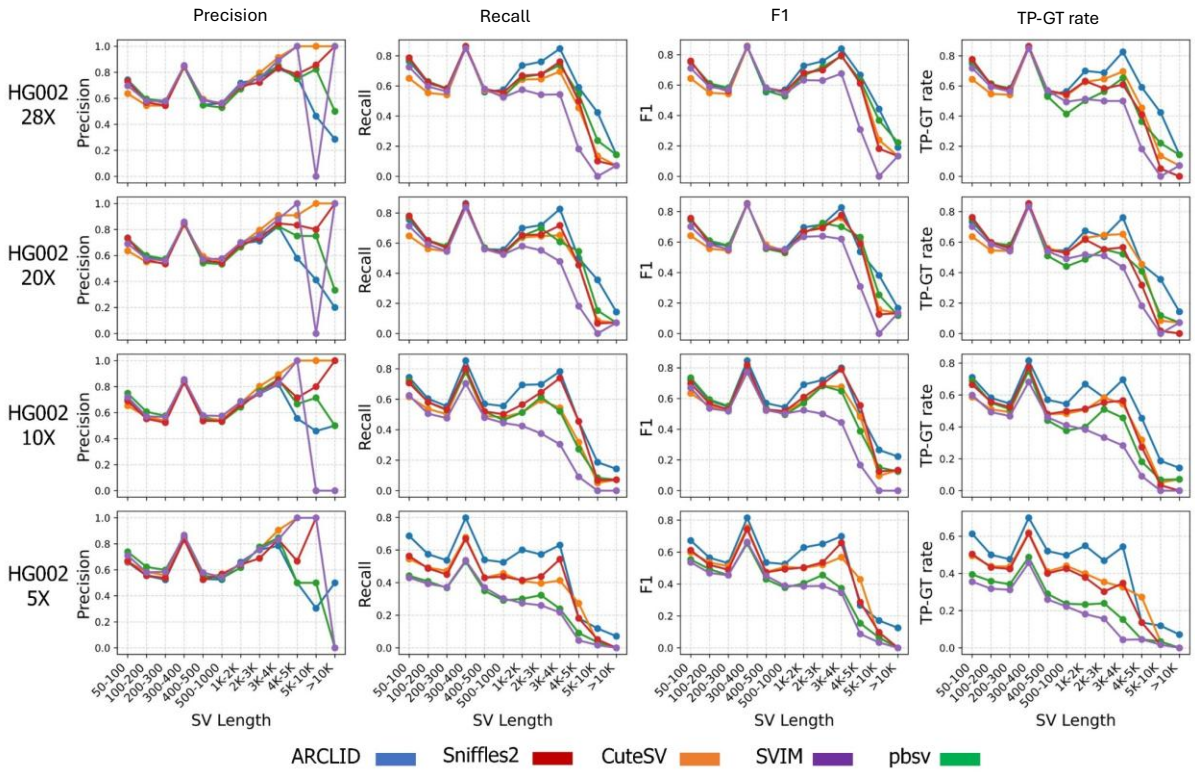

Figure 3. ARCLID's performance across SV size for all coverages on the HG002 HiFi (strict mode) datasets. Rows show sequence coverages (28×, 20×, 10×, 5×); columns show Precision, Recall, F1, and TP-GT rate. ARCLID maintains the highest F1/recall and TP-GT rate across coverages, with the advantage most pronounced for long SVs (≥1 kb) and at low coverage (10×–5×). All tools show declining performance for very large SVs, but ARCLID degrades less severely than other tools. It is also worth noting that the precision of some tools for very large SVs appears to jump to 100% (e.g., for SVs >10 kb, Sniffles2, CuteSV, and SVIM at 28× coverage; and similarly for SVs >4 kb at other coverages). This occurs because these tools detect only a single SV and miss all other SVs. Consequently, according to the definition of precision $Prec. = \frac{TP}{TP+FP}$, their reported precision becomes very high, in some cases reaching 100%. While this value is technically correct, it is misleading in practice because it does not account for the many undetected SVs, which are reflected instead in the recall.

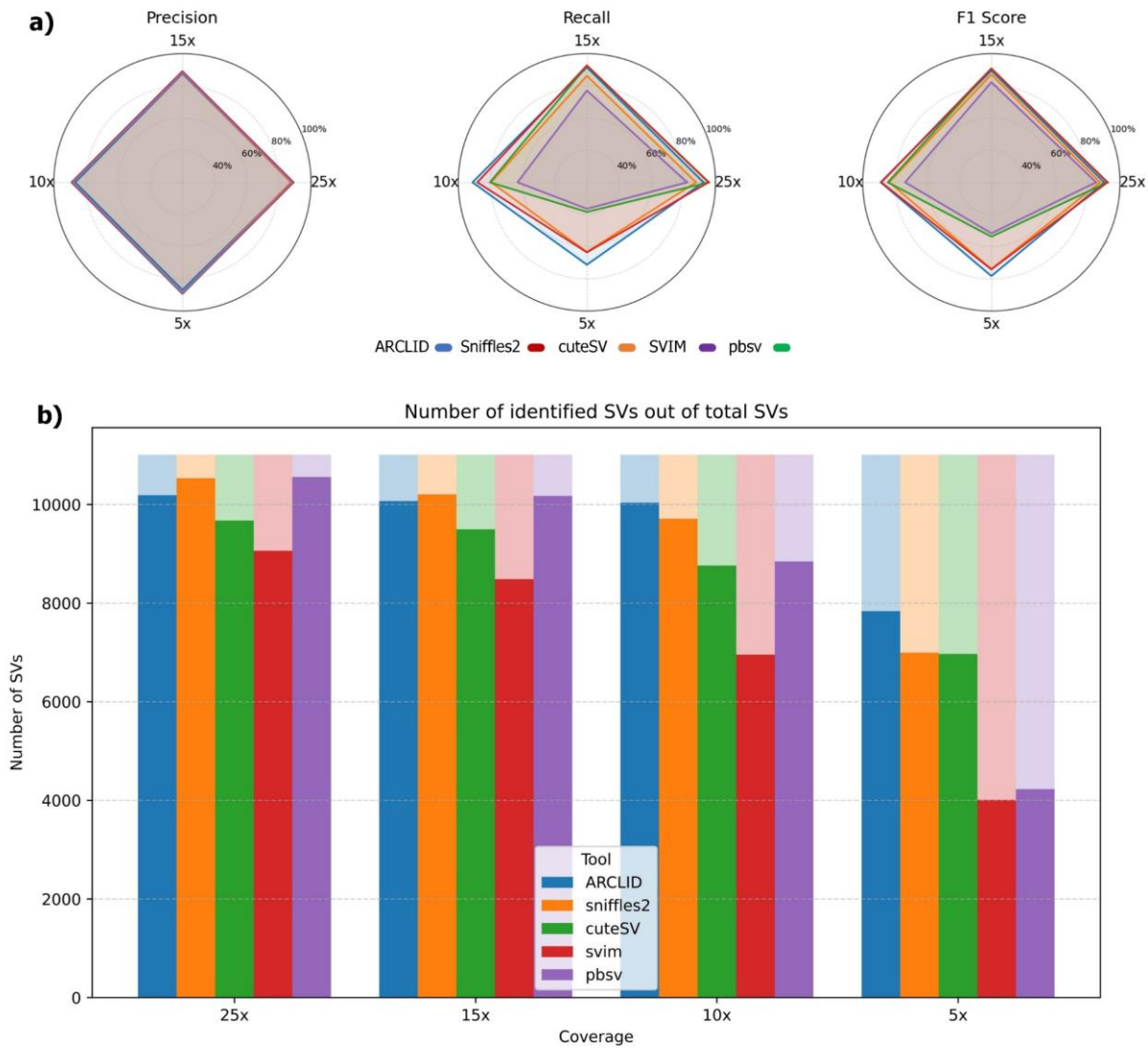

Figure 4. Evaluation of ARCLID's performance on the simulated HiFi reads. a) Radar charts show the tools' performance across various coverages in terms of Precision, Recall, and F1-score. And, **b)** the stacked bars indicate the number of SVs identified by each tool relative to the total SV set. ARCLID consistently detects a larger fraction of SVs across all coverage levels, particularly outperforming other tools at lower coverages.

Table 1. Numerical evaluation on HG002 in strict mode across different coverages for both INS and DEL (third column in Figure 2).

Coverage	Tool	Prec.	Rec.	F1	TP-GT rate
HG002-HiFi 28X	ARCLID	0.70	0.72	0.71	0.71
	Sniffles2	0.70	0.71	0.70	0.69
	CuteSV	0.67	0.65	0.66	0.65
	SVIM	0.69	0.66	0.67	0.65
	pbsv	0.70	0.70	0.70	0.67
HG002-HiFi 20X	ARCLID	0.70	0.71	0.70	0.70
	Sniffles2	0.70	0.70	0.70	0.68
	CuteSV	0.67	0.65	0.66	0.64
	SVIM	0.69	0.65	0.67	0.64
	pbsv	0.70	0.70	0.70	0.66
HG002-HiFi	ARCLID	0.68	0.70	0.69	0.67

10X	Sniffles2	0.67	0.65	0.66	0.61
	CuteSV	0.67	0.60	0.63	0.58
	SVIM	0.70	0.55	0.62	0.53
	pbsv	0.71	0.64	0.67	0.59
HG002-HiFi 5X	ARCLID	0.66	0.64	0.65	0.57
	Sniffles2	0.66	0.53	0.59	0.47
	CuteSV	0.67	0.52	0.59	0.48
	SVIM	0.69	0.40	0.51	0.32
	pbsv	0.71	0.41	0.52	0.36

Table 2. Numerical evaluation of ARCLID on HG002 28X across various SV sizes in strict mode, compared to other SV tools. Metrics marked with – are undefined due to division by zero (e.g., when no variants were detected in that category).

SV size	Tool	Prec.	Rec.	F1	TP-GT rate
50-100 bp	ARCLID	0.74	0.76	0.75	0.75
	Sniffles2	0.73	0.79	0.76	0.78
	CuteSV	0.64	0.65	0.64	0.64
	SVIM	0.70	0.73	0.71	0.72
	pbsv	0.73	0.77	0.75	0.76
100-200 bp	ARCLID	0.59	0.62	0.60	0.61
	Sniffles2	0.57	0.62	0.59	0.61
	CuteSV	0.54	0.55	0.54	0.55
	SVIM	0.58	0.60	0.59	0.59
	pbsv	0.59	0.63	0.61	0.61
200-300 bp	ARCLID	0.56	0.58	0.57	0.58
	Sniffles2	0.54	0.58	0.56	0.58
	CuteSV	0.54	0.54	0.54	0.54
	SVIM	0.58	0.56	0.57	0.56
	pbsv	0.58	0.58	0.58	0.58
300-400 bp	ARCLID	0.84	0.86	0.85	0.86
	Sniffles2	0.85	0.86	0.85	0.86
	CuteSV	0.84	0.86	0.85	0.86
	SVIM	0.85	0.85	0.85	0.85
	pbsv	0.85	0.86	0.85	0.85
400-500 bp	ARCLID	0.55	0.56	0.55	0.54
	Sniffles2	0.59	0.58	0.58	0.57
	CuteSV	0.59	0.57	0.58	0.57
	SVIM	0.59	0.58	0.58	0.57
	pbsv	0.55	0.58	0.56	0.53
0.5-1Kbp	ARCLID	0.56	0.57	0.56	0.56
	Sniffles2	0.56	0.56	0.56	0.54
	CuteSV	0.56	0.54	0.55	0.54
	SVIM	0.56	0.52	0.54	0.49
	pbsv	0.53	0.52	0.52	0.41
1K-2Kbp	ARCLID	0.72	0.74	0.73	0.70
	Sniffles2	0.69	0.67	0.68	0.63
	CuteSV	0.69	0.64	0.66	0.63
	SVIM	0.70	0.58	0.63	0.51
	pbsv	0.67	0.66	0.66	0.50
2K-3Kbp	ARCLID	0.75	0.76	0.75	0.69
	Sniffles2	0.72	0.68	0.70	0.58
	CuteSV	0.79	0.65	0.71	0.65
	SVIM	0.75	0.54	0.63	0.50

	pbsv	0.76	0,68	0.72	0.56
3K-4Kbp	ARCLID	0.83	0.85	0.84	0.83
	Sniffles2	0.83	0.76	0.79	0.61
	CuteSV	0.91	0.70	0.79	0.70
	SVIM	0.89	0.54	0.67	0.50
	pbsv	0.85	0.74	0.79	0.65
4K-5Kbp	ARCLID	0.76	0.59	0.66	0.59
	Sniffles2	0.79	0.50	0.61	0.41
	CuteSV	1	0.45	0.62	0.45
	SVIM	1	0.18	0.31	0.18
	pbsv	0.75	0.55	0.63	0.36
5K-10Kbp	ARCLID	0.46	0.42	0.44	0.42
	Sniffles2	0.86	0.10	0.18	0.05
	CuteSV	1	0.14	0.25	0.14
	SVIM	-	0	-	0
	pbsv	0.82	0.24	0.37	0.22
>10Kbp	ARCLID	0.29	0.14	0.19	0.14
	Sniffles2	1	0.07	0.13	0
	CuteSV	1	0.07	0.13	0.07
	SVIM	1	0.07	0.13	0.07
	pbsv	0.50	0.14	0.22	0.14

Table 3. Numerical evaluation of ARCLID on simulated HiFi data across various coverages in strict mode, compared to the performance of other SV tools.

Coverage	Tool	Prec.	Rec.	F1
Sim-HiFi 25X	ARCLID	0.88	0.93	0.90
	Sniffles2	0.89	0.96	0.92
	CuteSV	0.87	0.88	0.87
	SVIM	0.88	0.82	0.85
	pbsv	0.89	0.96	0.92
Sim-HiFi 15X	ARCLID	0.87	0.91	0.89
	Sniffles2	0.89	0.93	0.91
	CuteSV	0.88	0.86	0.87
	SVIM	0.88	0.77	0.82
	pbsv	0.89	0.92	0.90
Sim-HiFi 10X	ARCLID	0.87	0.91	0.89
	Sniffles2	0.89	0.88	0.88
	CuteSV	0.88	0.80	0.84
	SVIM	0.88	0.63	0.73
	pbsv	0.89	0.80	0.84
Sim-HiFi 5X	ARCLID	0.87	0.71	0.78
	Sniffles2	0.89	0.64	0.74
	CuteSV	0.89	0.63	0.74
	SVIM	0.89	0.36	0.51
	pbsv	0.89	0.38	0.53

Table 4. Evaluation of ARCLID on two HGSVC samples, NA19240 and HG00733, compared to other SV tools. CuteSV crashed after three days of processing the input bam files for both samples. Unlike GIAB, HGSVC does not provide high-

confidence regions, making the truth set and subsequent benchmarking less reliable. Therefore, this experiment was performed primarily to assess the generalizability of ARCLID's performance on datasets from independent resources.

Sample	Tool	Prec.	Rec.	F1	TP-GT rate
NA19240-HiFi 27X	ARCLID	0.70	0.55	0.62	0.51
	Sniffles2	0.67	0.61	0.64	0.56
	CuteSV	NA	NA	NA	NA
	SVIM	0.70	0.51	0.59	0.48
	pbsv	0.65	0.58	0.61	0.53
HG00733-HiFi 32X	ARCLID	0.70	0.55	0.62	0.51
	Sniffles2	0.66	0.60	0.63	0.56
	CuteSV	NA	NA	NA	NA
	SVIM	0.70	0.53	0.60	0.50
	pbsv	0.68	0.57	0.62	0.52

Discussion

In this study, we introduce ARCLID, an accurate and robust structural variant (SV) caller designed to meet the challenges of diverse sequencing conditions. ARCLID extracts informative features from aligned reads by transforming them into pileup images and simultaneously identifies, classifies, and genotypes SVs. This is achieved using two advanced deep learning models (YOLOv11x) that automatically learn both low-level and high-level features in the pileup images. By framing SV detection as an object detection problem, similar to approaches in computer vision, ARCLID leverages the power of deep learning to capture subtle SV signatures and handle a wide range of variant sizes.

Comparative analyses against widely used SV callers, including Sniffles2, cuteSV, SVIM, and pbsv, showed that these conventional tools often struggle in detecting SVs at lower sequencing coverages and for very large SVs. This limitation is especially critical in real-world scenarios, where researchers frequently work with low-coverage sequencing data due to budget or time constraints. In such contexts, the need for an SV caller that maintains high accuracy under suboptimal conditions is particularly pressing. Our benchmarking results demonstrate that ARCLID effectively addresses this need, offering consistent performance across the typically troublesome coverage levels and SV sizes, making it suitable for both academic research and industrial applications.

We evaluated ARCLID with Truvari under two benchmark modes, relaxed mode ($ref_dist = 1000\,bp$, $pctsim = 0.7$, $pctovl = 0$) and strict mode ($ref_dist = 50\,bp$, $pctsim = 0.9$, $pctovl = 0.9$), using multiple datasets: HG002 from GIAB, synthetic data generated by VISOR, and HG00733 and NA19240 from HGSVC. The evaluation showed that ARCLID maintains significantly higher robustness than previously published methods when coverage decreases, even for large SVs. This capability is critical in scenarios where only sparse data are available, such as cost-constrained population genomics studies that trade sequencing depth for population breadth [23]; clinical settings with limited or degraded DNA from small biopsies or FFPE samples [24]; and outbreak investigations requiring rapid, low-cost sequencing to guide public health action [25]. These findings raise the question of whether improved SV calling

algorithms, rather than costly increases in sequencing coverage, might offer the more cost-effective route to high-quality variant discovery?

Sequencing costs are a critical consideration in large-scale and clinical genomics projects. For example, Simonetti *et al.* [26] mentioned that the COVseq protocol for SARS-CoV-2 surveillance multiplexes hundreds of samples on a single run, allocating only ~0.5–1 million reads per sample (a modest read budget), yet still achieves ≥10× coverage across ≥95 % of the 29 kb genome for under US $15 per genome. Also, according to Genohub [27], whole-genome sequencing of a human sample at ~30× coverage with the PacBio Revio system, including library preparation, costs approximately US $2,000, producing ~90 Gb of HiFi data. In comparison, the older Sequel II system would require three SMRT cells to achieve similar throughput, at a cost of around US $8,700. While ~ 30× coverage has traditionally been viewed as the benchmark for accurate SV detection, ARCLID's robustness at lower coverages challenges this paradigm.

To illustrate the trade-off between sequencing cost, depth, and variant calling performance, we assumed a scenario in which SV calling was performed on the HG002 sample from GIAB using different tools at varying coverage levels (Figure 5). Sequencing costs were roughly estimated from Genohub (https://genohub.com/ngs/, accessed August 2025), an online platform for comparing and ordering sequencing services, for PacBio Sequel II HiFi sequencing (turnaround ~10 days, including library preparation) and should be regarded as approximate rather than exact values. This analysis highlights how accuracy scales with coverage: while all tools show reduced performance at lower depths, ARCLID maintains nearly the same performance at 10× coverage as at 30× coverage, with only a modest drop in accuracy. Importantly, this translates into a potential reduction in sequencing cost from ~8,200 USD (30×) to ~2,900 USD (10×), while still achieving almost identical results. These findings underscore ARCLID's ability to deliver reliable SV detection at substantially reduced sequencing depths, offering a practical pathway to lowering sequencing costs in both population genomics and clinical applications.

While ARCLID demonstrates strong robustness and accuracy for detecting insertions and deletions from PacBio HiFi data, it currently has some limitations. First, ARCLID does not yet support Oxford Nanopore Technologies (ONT) sequencing data, which is increasingly used in both research and clinical settings. Second, its current scope is restricted to insertion and deletion events and does not cover other SV types such as inversions, duplications, or translocations. Moreover, like most deep learning–based approaches, ARCLID's performance is influenced by the quantity and diversity of training data; access to larger and more heterogeneous datasets would likely further improve its accuracy and generalizability. These limitations are actively being addressed, and future versions of ARCLID will extend compatibility to ONT data, expand its detection capabilities to encompass a broader spectrum of structural variation, and leverage more extensive training datasets to maximize performance.

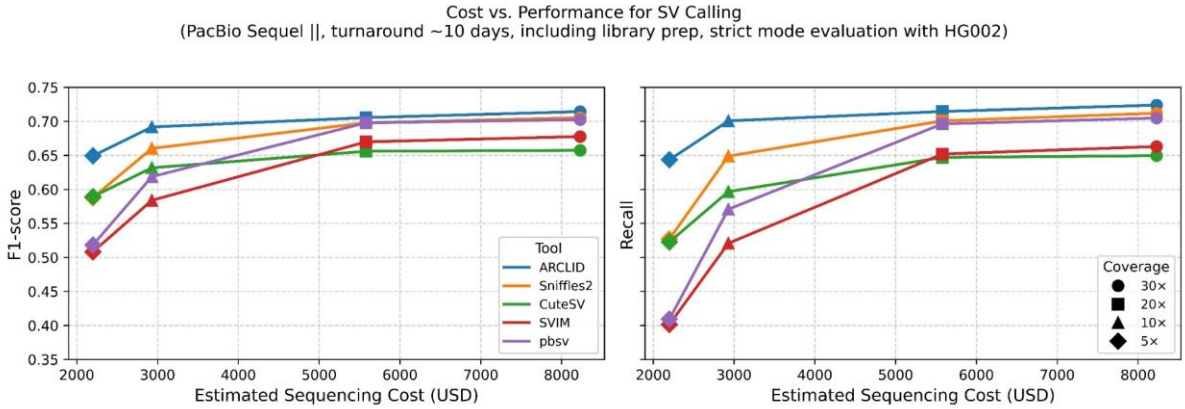

Figure 5. Estimated cost versus performance of structural variant calling with PacBio Sequel II HiFi sequencing (HG002, strict mode). F1-score (left) and recall (right) are shown based on various coverages (30×, 20×, 10×, 5×) and sequencing cost for all tools. The cost values are based on Genohub's webpage and represent a rough estimation of sequencing costs for different coverages. This analysis illustrates an imagined trade-off between sequencing depth, cost, and variant calling performance. ARCLID consistently maintains the highest F1-score and recall across all coverages, highlighting its potential to achieve reliable performance at lower coverage and thereby reduce sequencing costs compared to other tools.

Conclusion

In this work, we introduced ARCLID, a deep learning–based structural variant caller that leverages image representations of long-read alignments to achieve accurate detection and genotyping of insertions and deletions. Comprehensive benchmarking showed that ARCLID consistently outperforms or matches existing tools across variant sizes and sequencing depths, with a particular strength in detecting large SVs and sustaining robust performance even at low coverage (10×–5×). By enabling reliable SV discovery at reduced sequencing depths, ARCLID lowers sequencing costs and mitigates the need for deep sequencing, offering clear advantages for large-scale population genomics and clinical applications where resources and sample quality are often limiting. While the current version focuses on PacBio HiFi data and insertion/deletion events, the framework is extendable to additional SV classes and other sequencing technologies. Together, these results position ARCLID as a practical and cost-effective solution for accurate SV discovery, bridging the gap between advances in long-read sequencing and routine genomic studies.

Methods

Overview of ARCLID

The ARCLID approach consists of five main steps: (1) preprocessing alignment reads (2) extracting features, (3) generating three-channel pileup images from alignment files, (4) localizing (identifying), classifying, and genotyping structural variants (SVs) using two YOLOv11x models, and (5) postprocessing. A detailed description of these steps is provided in the following subsection.

Feature Extraction and Pileup Images

Two versions of pileup images are generated. The first version covers regions of 5 Kbp with a 1 Kbp overlap, designed to capture shorter indels (<2 Kbp). The second version spans 50 Kbp with a 10 Kbp overlap, intended to capture longer indels (>2 Kbp). The 5 Kbp images are used as input to the first YOLO model, while the 50 Kbp images are used for the second model. The input images are resized to (640, 640) in order to be injected to the models. We adopted this approach because very long SVs can span several pileup images, making them difficult to detect once divided.

Each image is constructed to support a sequencing depth of up to 150 reads per position. In these images, columns correspond to each base pair position in the reference genome, and rows correspond to aligned reads. The first channel accommodates the sequences in reads, the second channel stores insertions, deletions, and soft-clipped flags, the third channel is dedicated to the combination of quality scores and mapping qualities, and the fourth channel is to highlight if inter-alignment signals (if a read is split, supplementary, or reversed). It is worth noting that channels 2-4 are used for the input of the deep learning models and channel 1 is used when postprocessing.

The first and the second channel

To convert base pairs to numerical values, nucleotides A, T, C, and G are encoded as 40, 80, 120, and 160, respectively, while deletion (' D ') and insertion (' I ') flags are assigned values of 200 and 250, respectively.

Furthermore, the first row of the first channel is reserved for the used reference genome, while the second-row records the number of reads cover each position. Rows 2 to 150 contain the aligned sequences along with deletion flags (extracted from the CIGAR string). Rows 3-150 are to highlight sequences ‘D’ flags from CIGAR string, while rows 150 to 300 are designated for ‘I‘ flags. Finally, a horizontal line is drawn to represent the average coverage of the alignment file. This coverage line is equivalent to average coverage of the sample and helps the deep learning model to better understand coverage variations. Examples of the generated pileup images are presented in Figure 1 b and c.

Generation of the pileup image is not straightforward as the presence of deletion and insertions will introduce positional shifts between the reads and the reference sequence, meaning that populating the image simply by loading the aligned reads directly into the pileup image from the first aligned position to the last will cause a discrepancy between the displayed SV position and its corresponding label in the groundtruth VCF file. Therefore, careful handling of this process is essential to prevent misalignment, which could mislead the deep learning model during training. To address this issue, deletion flags ('D') are incorporated alongside aligned sequences in columns 2–150, while base pairs associated with insertion flags ('I') are extracted from the reads and placed in columns 150–300. This approach effectively mitigates positional shifts introduced by insertions, ensuring accurate SV representation in the pileup images.

The third channel

To combine mapping quality and quality scores for individual base reads, the two scores (from the aligner and sequencing machine, respectively) are converted from PHRED score into probabilities of error (Equations 5, and 6).

$$p_b = 10^{-\frac{Q_b}{10}} \qquad \text{Equation (5)}$$

$$p_m = 10^{-\frac{Q_m}{10}} \qquad \text{Equation (6)}$$

Where a base call has a PHRED quality score Q_b, it represents the probability of base-call error p_b. Similarly, the mapping quality Q_m describes the likelihood that the read is incorrectly mapped. Its associated probability of mapping error p_m.

There are now two probabilities of error: one from the base call and one from the mapping. Assuming (for simplicity) that these errors are independent events, we can combine them to find the overall error probability for a given base alignment. The error could occur if either: a) The base is called incorrectly, b) The read is mapped incorrectly, or c) Both. Under independence, the combined error probability $p_{combined}$ can be expressed using the complement rule:

- Probability that the base is correct: $1 - p_b$
- Probability that the read is mapped correctly: $1 - p_m$

The probability that both are correct is equal to Equation (7). Thus, the probability that *either* is wrong (the combined error probability) is computed by Equation (8).

$$(1 - p_b)(1 - p_m) = 1 - p_b - p_m + p_b p_m \qquad \text{Equation (7)}$$

$$p_{combined} = 1 - (1 - p_b)(1 - p_m) = p_b + p_m - p_b p_m \qquad \text{Equation (8)}$$

Intuitively, Equation (4) says If both the base call and mapping are high quality, the probability that either source contributes an error is somewhat larger than each individual probability, but not simply additive due to their intersection. Eventually, after computing Equation (4), we convert it back into PHRED score through Equation (9).

$$Q_{combined} = -10\log\,(p_b + p_m - p_b p_m) \qquad \text{Equation (9)}$$

The fourth channel

The fourth channel is to highlight inter-alignment signatures. To encode these signatures, we used a decision tree to classify each read. Figure 6 shows the binary decision tree designed to encode reads based on whether they are split, supplementary, or mapped in reverse orientation. Figure 6 illustrates the decision tree, including the root node ("is split?"), the internal nodes ("is supplementary?" and "is reversed?"), and the leaf nodes ("111", "110", etc.). In this decision tree, certain states are logically impossible, for example, cases where a read is not split but is marked as supplementary. Clearly, a read cannot be supplementary if it is not split. Therefore,

the two states 011 and 010 cannot occur. Leaf nodes demonstrate how a binary value is assigned to a read, then the binary value is converted to decimal one, and ultimately the value is multiplied by 20 to increase the separation between values, thereby helping the deep learning model better differentiate them.

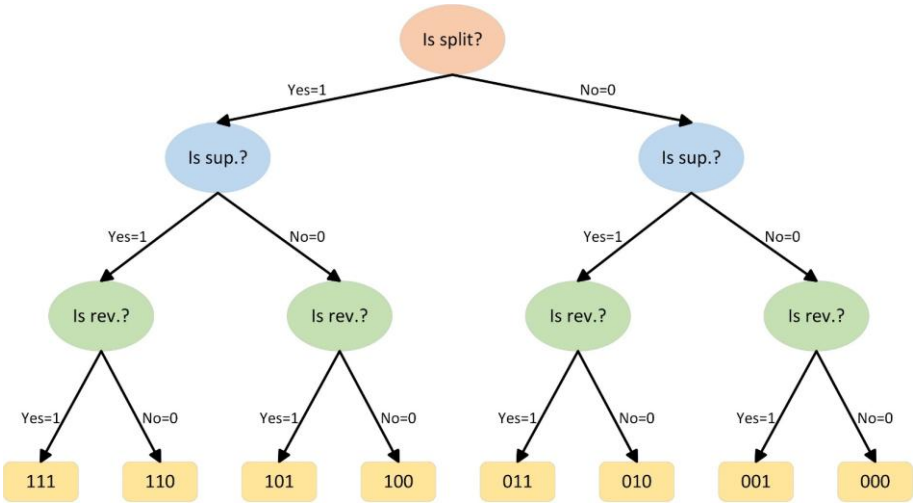

Figure 6. Decision tree for categorizing reads as split, supplementary, or reversed. The decision tree begins at the root node, which determines whether the read is split. The subsequent internal nodes check whether the read is supplementary or reversed. At the leaf nodes, each read is assigned a binary value. Finally, these binary values are converted to decimal form and multiplied by 20 to enhance the distance between categories.

Deep Learning Model and Training

The deep learning model employed in this study is an advanced object detection framework commonly used in computer vision tasks. We used YOLOv11x, which comprises 56.9 million parameters, and re-trained it using the HG002 sample from the 'Genome in a Bottle' (GIAB) consortium due to the high accuracy and reliability of its labels. Chromosomes 1–10 were allocated to the training set, chromosome 11 to the evaluation set, and chromosomes 12–22 to the testing set. HG002 HiFi data with various coverage levels (28×, 10×, 5×) were employed for training. The models were trained for 200 epochs with a batch size of 16, using pre-trained weights and an IoU threshold of 0.5. This IoU threshold was adjusted from the default value of 0.7 to 0.5 because variant positions are often shifted due to mapping or alignment algorithms. Figure 1 in the Supplementary Information illustrates the training process curves.

Additionally, because mapping algorithms often introduce shifts in the reported positions of structural variants (SVs), we were required to manually curate the pileup images to ensure that the variants' positions in the truth set exactly matched their locations after alignment. Without this step, the YOLO models became confused during training, as the coordinate annotations did not correctly align with the SV signals in the images, leading to poor model convergence. Initially, we attempted to train the models without manual curation, but this resulted in unsatisfactory learning performance. To overcome this issue, we manually curated approximately 50,000 SV pileup images, which substantially improved the training process and allowed the models to learn more effectively.

Postprocessing

Following the processing of YOLO's outputs, variants located within 100 base pairs and sharing the same SV class and genotype are merged. Subsequently, the candidate variants undergo a refinement process by revisiting their genomic positions within the pileup image. During this step, the variants are evaluated based on the existing indicators of ‘ D ‘ and ‘ I ‘ signals from CIGAR string present in the image; a variant is retained only if these indicators align with the predictions provided by the model. For those variants that are validated through this process, a consensus sequence is then derived by referencing their respective positions in the pileup image. Finally, the resulting variant information is systematically organized and recorded in a Variant Call Format (VCF) file.

Data Preparation

The alignment BAM file for HG002 (aligned to the hg37 reference genome using pbmm2) is publicly available; therefore, we directly employed this file without realigning the raw reads. In contrast, alignment files for NA19240 and HG00733 are not publicly available. For these samples, we downloaded the raw reads and aligned them to the hg38 reference genome using pbmm2. In addition, simulated reads were generated from the hg38 reference genome using VISOR. Detailed procedures for read mapping and generating simulated data are provided in the Supplementary Information (Execution Parameters).

Author Contributions

Sajad Tavakoli: Conceptualization; methodology; software; validation; benchmarking; formal analysis; data curation and annotation; writing-original draft; writing-review; visualization; project administration. **Rasmus John Normand Frandsen:** Conceptualization; project administration; supervision; validation; writing-editing; writing-review. **Marjan Mansouravar:** Conceptualization; project administration; supervision; validation; writing-review.

Acknowledgements

The authors acknowledge DTU computing center for enabling the useof high performance computing clusters for this research. The authors received no financial support for the research, authorship, and/or publication of this article. The project has been carried out within the *Bio Digitalization and Data Science Group (BDD Group)* at the Department of Bioengineering, Technical University of Denmark (DTU).

Conflict of Interest Statement

The authors declare no conflicts of interest.

Data Availability

All data used in this study are publicly available and can be downloaded from the following links:

- Reference Genome hg37 and hg38:
  - hg37: ftp://ftp-trace.ncbi.nih.gov/1000genomes/ftp/technical/reference/phase2_reference_assembly_sequence/hs37d5.fa.gz
  - hg38: http://hgdownload.soe.ucsc.edu/goldenPath/hg38/bigZips/hg38.fa.gz
- HG002 HiFi aligned reads: ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/HG002_NA24385_son/PacBio_CCS_15kb/alignment/HG002.Sequel.15kb.pbmm2.hs37d5.whatshap.haplotag.RTG.10x.trio.bam
- HG002 truth set: https://ftp-trace.ncbi.nlm.nih.gov/giab/ftp/data/AshkenazimTrio/analysis/NIST_SVs_Integration_v0.6/HG002_SVs_Tier1_v0.6.vcf.gz
- HG002 high-confidence regions: https://ftp-trace.ncbi.nlm.nih.gov/giab/ftp/data/AshkenazimTrio/analysis/NIST_SVs_Integration_v0.6/HG002_SVs_Tier1_v0.6.bed
- HG00733 HiFi reads: https://ftp.1000genomes.ebi.ac.uk/vol1/ftp/data_collections/HGSVC2/working/20190925_PUR_PacBio_HiFi/
- NA19240 HiFi reads: https://ftp.1000genomes.ebi.ac.uk/vol1/ftp/data_collections/HGSVC2/working/20191005_YRI_PacBio_NA19240_HiFi/
- HG00733 and NA19240 truth set: https://ftp.1000genomes.ebi.ac.uk/vol1/ftp/data_collections/HGSVC2/working/20210806_PAV_VCF/
- Simulated HiFi Reads: The entire process to generate simulated HiFi reads is provided in Supplementary Information.

Code Availability

ARCLID is publicly available on GitHub: https://github.com/sajadtavakoli/ARCLID

References

[1] R. L. Collins *et al.*, “A structural variation reference for medical and population genetics,” *Nature*, vol. 581, no. 7809, pp. 444–451, 2020, doi: 10.1038/s41586-020-2287-8.

[2] J. Weischenfeldt, O. Symmons, F. Spitz, and J. O. Korbel, “Phenotypic impact of genomic structural variation: insights from and for human disease,” *Nat Rev Genet*, vol. 14, no. 2, pp. 125–138, 2013, doi: 10.1038/nrg3373.

[3] Z. Matuszek *et al.*, “Base editing of trinucleotide repeats that cause Huntington’sdisease and Friedreich’s ataxia reduces somatic repeat expansions in patient cells and in mice,” *Nat Genet*, vol. 57, no. 6, pp. 1437–1451, 2025, doi: 10.1038/s41588-025-02172-8.

[4] D. M. McDonald-McGinn *et al.*, “22q11.2 deletion syndrome,” *Nat Rev Dis Primers*, vol. 1, no. 1, p. 15071, 2015, doi: 10.1038/nrdp.2015.71.

[5] G. A. Logsdon *et al.*, “Complex genetic variation in nearly complete human genomes,” *Nature*, vol. 644, no. 8076, pp. 430–441, 2025, doi: 10.1038/s41586-025-09140-6.

[6] B. D. Redelings, I. Holmes, G. Lunter, T. Pupko, and M. Anisimova, “Insertions and Deletions: Computational Methods, Evolutionary Dynamics, and Biological Applications,” *Mol Biol Evol*, vol. 41, no. 9, p. msae177, Sep. 2024, doi: 10.1093/molbev/msae177.

[7] V. Popic *et al.*, “Cue: a deep-learning framework for structural variant discovery and genotyping,” *Nat Methods*, vol. 20, no. 4, pp. 559–568, 2023, doi: 10.1038/s41592-023-01799-x.

[8] Y. H. Liu, C. Luo, S. G. Golding, J. B. Ioffe, and X. M. Zhou, “Tradeoffs in alignment and assembly-based methods for structural variant detection with long-read sequencing data,” *Nat Commun*, vol. 15, no. 1, p. 2447, 2024, doi: 10.1038/s41467-024-46614-z.

[9] H. Li, “Minimap2: pairwise alignment for nucleotide sequences,” *Bioinformatics*, vol. 34, no. 18, pp. 3094–3100, Sep. 2018, doi: 10.1093/bioinformatics/bty191.

[10] K. Ye, M. H. Schulz, Q. Long, R. Apweiler, and Z. Ning, “Pindel: a pattern growth approach to detect break points of large deletions and medium sized insertions from paired-end short reads,” *Bioinformatics*, vol. 25, no. 21, pp. 2865–2871, Nov. 2009, doi: 10.1093/bioinformatics/btp394.

[11] M. Cretu Stancu *et al.*, “Mapping and phasing of structural variation in patient genomes using nanopore sequencing,” *Nat Commun*, vol. 8, no. 1, p. 1326, 2017, doi: 10.1038/s41467-017-01343-4.

[12] N. Gaitán and J. Duitama, “A graph clustering algorithm for detection and genotyping of structural variants from long reads,” *Gigascience*, vol. 13, p. giad112, Jan. 2024, doi: 10.1093/gigascience/giad112.

[13] F. J. Sedlazeck *et al.*, “Accurate detection of complex structural variations using single-molecule sequencing,” *Nat Methods*, vol. 15, no. 6, pp. 461–468, 2018, doi: 10.1038/s41592-018-0001-7.

[14] M. Smolka *et al.*, “Detection of mosaic and population-level structural variants with Sniffles2,” *Nat Biotechnol*, vol. 42, no. 10, pp. 1571–1580, 2024, doi: 10.1038/s41587-023-02024-y.

[15] T. Jiang *et al.*, “Long-read-based human genomic structural variation detection with cuteSV,” *Genome Biol*, vol. 21, no. 1, p. 189, 2020, doi: 10.1186/s13059-020-02107-y.

[16] D. Heller and M. Vingron, “SVIM: structural variant identification using mapped long reads,” *Bioinformatics*, vol. 35, no. 17, pp. 2907–2915, Sep. 2019, doi: 10.1093/bioinformatics/btz041.

[17] “Pacific Biosciences. PacificBiosciences/pbsv: pbsv - PacBio structural variant (SV) calling and analysis tools.” https://github.com/PacificBiosciences/pbsv

[18] Rahima Khanam and Muhammad Hussain, “YOLOv11: An Overview of the Key Architectural Enhancements,” *ArXiv*, 2024. https://arxiv.org/abs/2410.17725

[19] J. M. Zook *et al.*, “A robust benchmark for detection of germline large deletions and insertions,” *Nat Biotechnol*, vol. 38, no. 11, pp. 1347–1355, 2020, doi: 10.1038/s41587-020-0538-8.

[20] D. Bolognini, A. Sanders, J. O. Korbel, A. Magi, V. Benes, and T. Rausch, “VISOR: a versatile haplotype-aware structural variant simulator for short- and long-read sequencing,” *Bioinformatics*, vol. 36, no. 4, pp. 1267–1269, Feb. 2020, doi: 10.1093/bioinformatics/btz719.

[21] M. J. P. Chaisson *et al.*, “Multi-platform discovery of haplotype-resolved structural variation in human genomes,” *Nat Commun*, vol. 10, no. 1, p. 1784, 2019, doi: 10.1038/s41467-018-08148-z.

[22] A. C. English, V. K. Menon, R. A. Gibbs, G. A. Metcalf, and F. J. Sedlazeck, “Truvari: refined structural variant comparison preserves allelic diversity,” *Genome Biol*, vol. 23, no. 1, p. 271, 2022, doi: 10.1186/s13059-022-02840-6.

[23] M. E. Allentoft *et al.*, “Population genomics of post-glacial western Eurasia,” *Nature*, vol. 625, no. 7994, pp. 301–311, 2024, doi: 10.1038/s41586-023-06865-0.

[24] S. Basyuni *et al.*, “Large-scale analysis of whole genome sequencing data from formalin-fixed paraffin-embedded cancer specimens demonstrates preservation of clinical utility,” *Nat Commun*, vol. 15, no. 1, p. 7731, 2024, doi: 10.1038/s41467-024-51577-2.

[25] D. F. Nieuwenhuijse, A. van der Linden, R. H. G. Kohl, R. S. Sikkema, M. P. G. Koopmans, and B. B. Oude Munnink, “Towards reliable whole genome sequencing for

outbreak preparedness and response," *BMC Genomics*, vol. 23, no. 1, p. 569, 2022, doi: 10.1186/s12864-022-08749-5.

[26] M. Simonetti *et al.*, "COVseq is a cost-effective workflow for mass-scale SARS-CoV-2 genomic surveillance," *Nat Commun*, vol. 12, no. 1, p. 3903, 2021, doi: 10.1038/s41467-021-24078-9.

[27] Kristyn Liu, "PacBio Revio: High-Quality Long Reads at an Affordable Price." https://blog.genohub.com/2024/02/16/pacbio-revio-high-quality-long-reads-at-an-affordable-price/?utm_source=chatgpt.com