OXFORD

*Briefings in Bioinformatics*, 2023, **24(2)**, 1–11
https://doi.org/10.1093/bib/bbad071
Advance access publication date 3 March 2023
**Problem Solving Protocol**

# Linear: a framework to enable existing software to resolve structural variants in long reads with flexible and efficient alignment-free statistical models

Chenxu Pan, René Rahn, David Heller and Knut Reinert

Corresponding author: Chenxu Pan, Department of Mathematics and Computer Science, Freie Universität Berlin, Takustr. 9, Berlin 14195, Germany. Telephone: +49 30 838 75222 Fax:+49 (0)30 838-475222; E-mail chenxu.pan@fu-berlin.de

## Abstract

Alignment is the cornerstone of many long-read pipelines and plays an essential role in resolving structural variants (SVs). However, forced alignments of SVs embedded in long reads, inflexibility of integrating novel SVs models and computational inefficiency remain problems. Here, we investigate the feasibility of resolving long-read SVs with alignment-free algorithms. We ask: (1) Is it possible to resolve long-read SVs with alignment-free approaches? and (2) Does it provide an advantage over existing approaches? To this end, we implemented the framework named Linear, which can flexibly integrate alignment-free algorithms such as the generative model for long-read SV detection. Furthermore, Linear addresses the problem of compatibility of alignment-free approaches with existing software. It takes as input long reads and outputs standardized results existing software can directly process. We conducted large-scale assessments in this work and the results show that the sensitivity, and flexibility of Linear outperform alignment-based pipelines. Moreover, the computational efficiency is orders of magnitude faster.

**Keywords:** alignment-free approach, graph generative model, structural variants resolution, long-read analysis.

## Introduction

Structural variants (SVs) are one of the most prominent topics in genetics [1, 2]. The topic relates to many fields of research [3]. For instance, many human diseases, such as autism and cancer, are associated with genomic rearrangement [4, 5]. In the past years, comprehensive techniques such as whole genome sequencing of next-generation sequencing (NGS) were applied to the resolution of novel SVs [6, 7]. However resolving complex rearrangement remains challenging in the analysis of long-read SVs [8–10].

Biotechnological advances in sequencing are astounding and have led to several interesting sequencing platforms with different key parameters regarding read length, error profile and sequencing costs [11–13]. Mainstream long-read sequencing technologies such as Pacific Biosciences (PacBio) and Oxford Nanopore Technologies (ONT) sequencing are different from the NGS sequencing [14] with respect to read length and error profiles. Progress in long-read platforms and corresponding pipelines have facilitated the better prediction of genomic rearrangements [15, 16]. Short reads are effective in resolving single nucleotide variants and small insertions and deletions (indels) [11, 17], whereas larger SVs > 100 bps are more amenable to long-read sequencing [18, 19]. This is shown in many publications, in which novel SVs missed by short-read pipelines [20] are resolved by long-read pipelines [21–26].

Alignment is the cornerstone of long-read SV detection pipelines, and therefore, fundamental alignment algorithms are critical to the overall performance of long-read SV detection pipelines [16, 17, 27, 28]. Unlike short reads, long reads are noisy and likely to contain complex rearrangements [26, 29]. Efforts have been made to develop algorithms that can process complex rearrangements in long reads in the past years. Nevertheless, aligning long-read complex rearrangements remains difficult, since long-read alignment algorithms commonly have to seek a compromise between effectiveness and efficiency [30, 31]. Moreover, aligning SVs-enriched sequences is particularly computationally demanding. Limited by computational complexity, alignment-based algorithms commonly employ heuristics to simplify the computational complexity, which inevitably introduces unpredictable bias, such as forced alignment of SVs, into results. Furthermore, it is challenging to integrate new algorithms of identifying novel SVs into existing alignment-based frameworks since aligners commonly require additional optimizations, such as the hardware acceleration for the identification of insertions and deletions [32–34], to make the algorithms computationally practical.

Here, we present an aLIgNment-freE framework for resolving long-read vARiants named Linear. It takes as input long reads and outputs alignment-free results compatible with existing alignment-based software, such as SVs callers and visualization tools. The framework adopts approaches grounded in the statistical model with optimizations making the computation efficient. Large-scale assessments show that Linear can effectively resolve

**Chenxu Pan** is a PhD student in the department of Mathematics and Computer Science, Freie Universität Berlin.
**René Rahn** is an analyst and programmer in the department of Mathematics and Computer Science, Freie Universität Berlin.
**David Heller** is a PhD student in the department of Computational Molecular Biology, Max Planck Institute for Molecular Genetics.
**Knut Reinert** is a professor in the department of Mathematics and Computer Science, Freie Universität Berlin and Max Planck Fellow at the Max-Planck-Institute for Molecular Genetics.
**Received:** August 29, 2022. **Revised:** January 13, 2023. **Accepted:** February 8, 2023

different types of long-read SVs. Performance in aspects of sensitivity, diversity and flexibility is generally better than aligners. Moreover, Linear requires less memory consumption and about 3.5% runtime of aligners to run. Therefore it is promising to develop efficient alignment-free algorithms for long-read SVs identification based on the framework.

# Results

## Evaluation of simulated data

Diversity (spectrum) is a key metric for long-read SV detection. We use different kinds of simulated SVs planted in long reads to assess the SVs spectrum each software can identify. We particularly evaluated three key parameters of the spectrum, the type of SVs, the length of SVs and the sequencing error on average.

The dataset comprises 48 groups of SVs embedded in simulated PacBio and ONT reads. We simulated three different levels of sequencing error on average, namely 0.5%, 15% and 20% for PacBio reads, and 10%, 15% and 20% for ONT reads. Sequencing error of 0.5% is to simulate the highly accurate long sequencing reads, also known as the PacBio HiFi reads, and 10%, 15%, 20% are to simulate the long raw reads. The PacBio reads are simulated using PBSIM [35]. The ONT reads are simulated using NanoSim [36]. The error profile of the long reads is based on the alignment of real long reads downloaded from the project Genome in a bottle (GiaB) [37, 38]. Furthermore, 12 types of SVs comprising insertions (INS)/deletions (DEL), duplications (DUP) and inversions (INV) ranging from 200*bps* to 1*k*bps, are independently simulated and planted at random positions of each simulated read by using the R package RSVSim [39].

We use long-read aligners Minimap2, NGMLR, SKSV [40], which is a very efficient SV detector for PacBio HiFi reads, and Linear to process the simulated reads. We enabled the '-x map-pb', '-x map-ont' options of Minimap2 and the '-x pacbio', '-x ont' options of NGMLR for PacBio and ONT reads. We run submoduls of SKSV 'index' and 'aln' for indexing the reference genomes and processing reads. We did not use SVs callers due to the following considerations:

1. Since SVs are simulated with given sizes, types and positions in reads, it is straightforward to directly compare the results of each software with the simulated SVs.
2. Forced alignment is a bottleneck of detecting nonlinear SVs such as INVs and DUPs. SVs callers are unlikely to recover SVs from forced alignment since forced alignment loses all information about the SVs.

Therefore, we did not use SVs callers in the assessment of simulated reads and SVs to eliminate the interference of SVs caller.

Figure 1 shows the spectrum measured by the recall and the precision of each software. (Supplementary Figures 1 and 2 are the spectrum of other long-read aligners). As expected, the aligners are significantly more effective in identifying insertions and deletions than inversions and duplications, since both aligners are explicitly optimized for insertions and deletions by applying the convex gap model or the two-stage affine gap model. However, the downside of the optimization is that it is compute-intensive and ineffective in identifying inversions and duplications. In contrast, the overall performance of Linear is comparable to the aligners. Furthermore, it is notably more effective in identifying inversions and duplications, particularly in lengths within 500bps. These SVs are commonly forcibly aligned by aligners and can hardly be corrected by downstream analysis since they lose almost all critical information of the SVs. In contrast, alignment-free models can better process these short nonlinear rearrangements embedded in reads. As a result, Linear identifies a broader spectrum of SVs than others.

We also assess the deviation of alignment-free results by evaluating the distances between detected breakpoints of SVs and the true ones. Figure 2A shows the overall empirical cumulative distribution (eCDF) of deviations of breakpoints. The mean of deviations is 8.54bps and 94.9% are within 25bps. Supplementary figure 3 shows deviations for each type of simulated SVs in the assessment. As the control group, we evaluated true positive rates (TPRs) and false positive rates (FPRs) of regular reads without SVs planted. Supplementary section 3.4 described the criteria of the assessment. Figure 2B is the receiver operating characteristic curve (ROC) curve of Linear and aligners for SVs-free sequences. We found in the figure that TPRs and FPRs of Linear are comparable to the aligners. Specifically, Minimap2 generates the most accurate results. The conclusion of SVs-free sequences is consistent with the assessment of simulated SVs, where Minimap2 performs better on collinear SVs (INSs and DELs). The FPRs of Linear and NGMLR are relatively higher compared to Minimap2. We investigated the results and found higher FPRs of Linear and NGMLR for highly repeated regions. However, it is partly due to the models of Linear and NGMLR tending to resolve repeated regions as nested structures rather than simple collinear structures and thus reporting more secondary results, which are regarded as false positives in the assessment.

## Evaluation of genuine data

### *Data protocol and software compatibility test*

We set up the assessment based on datasets of Ashkenazim Trio HG002 and NA12878 HG001 from GiaB. The assessment data comprises PacBio raw reads, HiFi reads and ONT Nanopore sequencing reads, which can be accessed at the GiaB or the Sequence Read Archive (SRA) site.

1. HG002 PacBio raw reads: https://ftp-trace.ncbi.nlm.nih.gov/giab/ftp/data/AshkenazimTrio/HG002_NA24385_son/PacBio_MtSinai_NIST/PacBio_fasta.
2. HG002 PacBio HiFi reads: https://ftp-trace.ncbi.nlm.nih.gov/giab/ftp/data/AshkenazimTrio/HG002_NA24385_son/PacBio_CCS_15kb_20kb_chemistry2/reads/
3. HG002 ONT reads comprising SRR18363750, SRR18363747, SRR18363749 archives are from the SRA database.
4. NA12878 PacBio CCS reads comprising SRR1950266-SRR1950290 archives from the SRA database.

We also use the SVs dataset from https://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/analysis/PacBio_pbsv_05212019/ as the ground truth, which is detected by Minimap2 and the SVs caller PBSV. We prepared the high-quality SVs callset based on the dataset by validating its SVs with the NGMLR-Sniffles pipeline [34]

We tested the compatibility of Linear with PBSV and cuteSV [41]. We first use Linear to process the PacBio and ONT reads. Then we use PBSV and cuteSV to call SVs from the results of Linear. Finally, we filter the results to remove SVs of length $\leqslant$ 100bps and supported by less than 5 reads. Table 1 summarizes the number of SVs detected by the two pipelines. In the test, PBSV and cuteSV can directly process the results of Linear and generate the SVs file correctly. We use PBSV submodules 'discover' and 'call' for SVs calling. cuteSV and all other software in the assessment ran with default parameters tailored for PacBio and ONT reads.

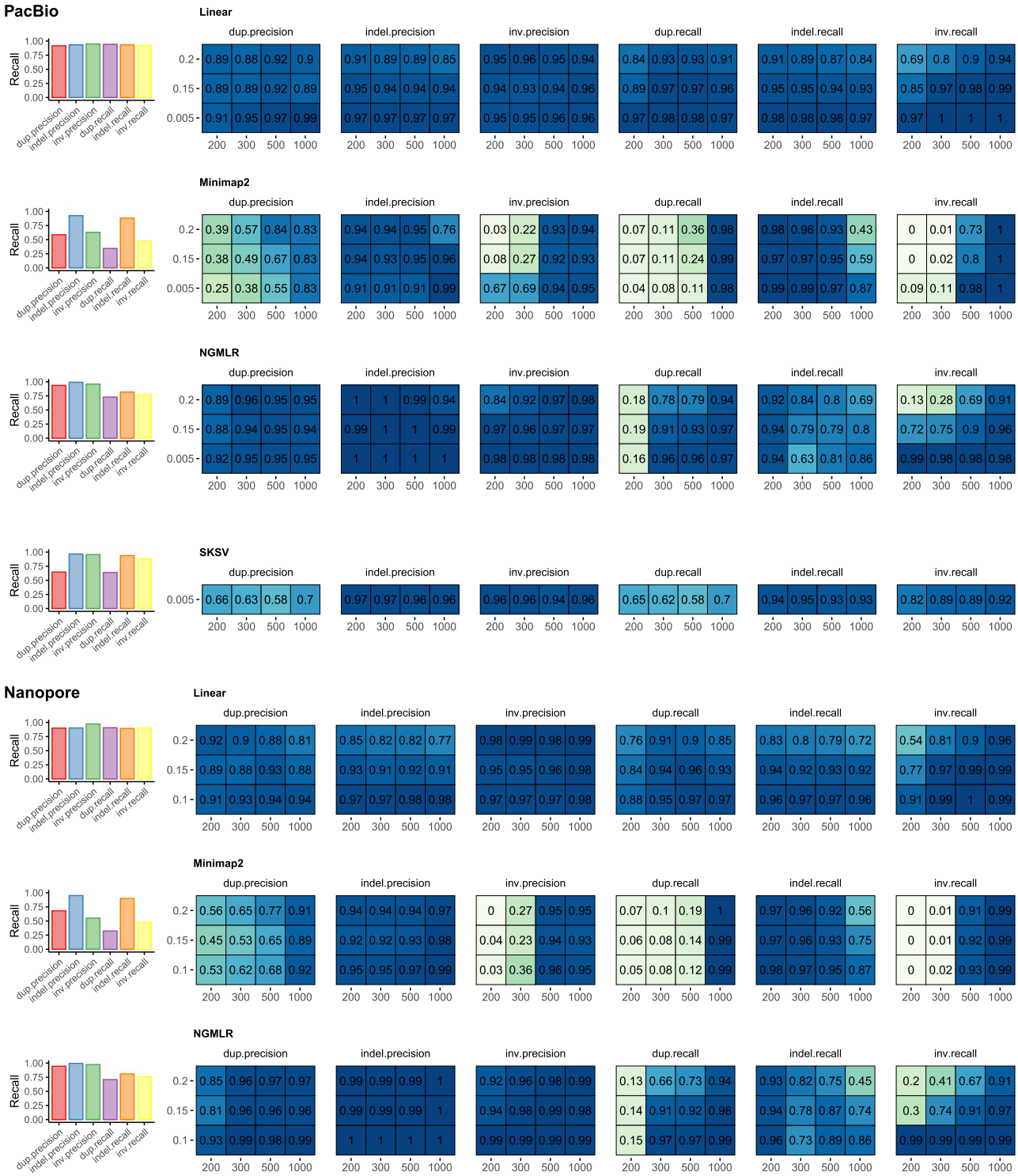
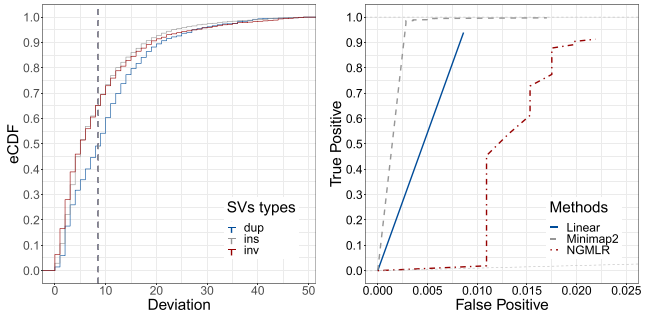



**Figure 1.** Recall and precision of detecting simulated SVs including (tandem) duplications, inversions, insertions and deletions of different lengths from 200 to 1000bps embedded in long reads of average sequencing error rate 0.05 (simulated HiFi reads), 0.15, 0.2 for PacBio reads, and 0.1, 0.15, 0.2 for ONT reads. The spectrum are colored according to the recall and the precision of each type of SVs. The bar plots are the average recall and precision of each software.

We tested the compatibility of Integrative Genomics Viewer (IGV). Figure 3A is an example of visualized alignment-free results of HG002 PacBio raw reads comprising a deletion of 844bps. Expression 11 in the materials and methods section formally defines the alignment-free cigars, which convert alignment-free results to the format compatible with IGV. The expression defines the virtual gaps reflecting the expected deviations of the results from the true ones. Since the expected deviations of alignment-free results are commonly less than 50bps, the image of alignment-free results, such as Figure 3A, comprises many short random gaps, namely the virtual gaps. However, gaps over 50bps, such as about 844bps in 32× reads, are from a real deletion since the gap lengths are significantly ($P$-value $< 0.05$) larger than the expected deviation.

**Table 1.** Summary of SVs in HG001 and HG002 detected by Linear-based pipelines. The column of 'Support reads' is the minimum supporting reads. Columns of INV, DUP, INS, DEL are identified inversions, duplications, insertions and deletions. Dataset of PacBio reads are processed by Linear-PBSV. Dataset of ONT reads are processed by Linear-cuteSV

| Dataset | Platforms | Coverage | Average length | Total SVs | Support reads | INV | DUP | INS | DEL |
|---|---|---|---|---|---|---|---|---|---|
| Ashkenazim HG002 | PacBio raw | 55 | 7912bps | 16109 | $\geqslant 5$ | 361 | 1944 | 6694 | 7110 |
| | | | | | $\geqslant 10$ | 324 | 1386 | 5424 | 6014 |
| | ONT raw | 45 | 7371bps | 23866 | $\geqslant 5$ | 117 | 6383 | 6171 | 11194 |
| | | | | | $\geqslant 10$ | 79 | 5095 | 4891 | 8930 |
| NA12878 HG001 | PacBio HiFi | 29 | 5011bps | 16845 | $\geqslant 5$ | 444 | 3433 | 5494 | 7474 |
| | | | | | $\geqslant 10$ | 308 | 1354 | 3755 | 5493 |




(A) Deviations of detected SVs (B) ROC of SVs-free reads

**Figure 2.** (A) is the eCDF of deviations of SVs detected by Linear from the true ones. (B) as the control, is the ROC curve of long reads without SVs planted.

## *Recalling real SVs*

Simulation cannot reflect all aspects of biological data. Therefore, we evaluate the performance of Linear-based pipeline on real data. It is worth noting that real datasets used as the ground truth inevitably contain a considerable amount of false negative and false positive data since the approaches for detecting SVs are commonly heuristics. It is difficult to precisely evaluate the recall and precision on real SVs due to lacking 'gold-standard' datasets.

Table 2 is the summary of identified SVs in the high-quality callset of HG001 and HG002. Identified and indicated SVs in the table are validated by comparing the endpoints of events detected by Linear to the true ones. The column of PBSV in the table shows SVs detected by Linear-PBSV. In the results, Linear can identify most events in the reads, while a considerable amount of identified events cannot be recalled by PBSV. We investigated these SVs and found that PBSV successfully detects the events, but cannot compute the breakpoints for the events because the deviations of endpoints are beyond the bound of PBSV for consistent breakpoints. Therefore, PBSV cannot call them. The performance of Linear-PBSV can be further improved if the consistency of the breakpoints generated by Linear can be optimized. Supplementary Figure 4 further shows the deviations and consistency for the two datasets.

We then evaluate the recall and precision corresponding to the coverage of reads required to detect SVs. We use the SVs recorded in the dataset as the true SVs. And the true positive is estimated by the number of detected true SVs. It is worth noting that the estimated precision is in fact lower than the exact one, since a considerable amount of false positives are probably true SVs that were not found before. Figure 4A is the recall and precision of Linear. The maximum precision is 80% when the supporting reads are over 5 ($\mathbf{min}(N_P) = 5$). Thus, we assume Linear needs at least 5 supporting reads to report SVs. Denote $N_{FN}$ the number of false negatives. Then the maximum coverage of reads required to report SVs is $N_{FN}+\mathbf{min}(N_P)$. Figure 4B shows the recall for different levels of coverage when supporting reads $\geqslant 5$. According to the figure, 13 $\sim$ 17$\times$ coverage is sufficient to recall about 90% true SVs with over 5 supporting reads and the precision is about 80% according to Figure 4A.

Furthermore, we set up three different pipelines combining cuteSV with Linear, Minimap2 and SKSV. We apply each pipeline to PacBio raw reads and HiFi reads of HG002. Table 3 shows the recall and precision of each pipeline. The precision of Linear-cuteSV is relatively lower than the two other pipelines. We investigated the results and found that Linear-cuteSV reports significantly more SVs in the centromere than the two other pipelines. Moreover, approximately 30% false positives of Linear-cuteSV are in the centromere. Hence the drop of the precision of Linear-cuteSV is largely attributable to the repetitive regions. On the other hand, Linear-cuteSV detects more SVs, 93.2% recall for the raw read callset and 70.3% recall for the HiFi read callset, than the two other pipelines. Linear-cuteSV especially performs better for PacBio raw reads, with very close precision and significantly (25%) higher recall than Minimap2-cuteSV. The assessment suggests that Linear-cuteSV is an effective pipeline, which is sensitive to long-read SVs, although it is probably overconfident in some highly repeated regions.

## *Case study of nested SVs*

Nested SVs are commonly more complex than basic ones. We did not analyze nested SVs on a large scale due to lacking 'gold-standard' datasets [17, 42]. Since few existing SVs callers are capable of calling nested SVs directly, we apply filtration to basic SVs detected in HG001 and HG002 to search for nested SVs. We specifically search for nested SVs composed of two overlapped basic SVs, one of which is the INV, since it is more difficult than other types of SVs to resolve. In the assessment of simulated SVs discussed above, we found that short INVs embedded in reads are commonly forcibly aligned by aligners. In contrast, Linear performs better in identifying INVs. The assessment of detecting real SVs is consistent with the simulated assessment in Figure 1, where aligners are less effective in identifying inversions of $\leqslant$ 400bps embedded in noisy reads. For instance, Figure 3B–D shows the results of the inversion of 184bps hidden in PacBio raw reads. Furthermore, we found that 95 and 176 candidates of nested SVs in the HG001 and HG002. Figure 3E and F is two candidates of nested SVs based on basic SVs detected by Linear. Figure 3E is an inverted duplication (INVDUP) of 2830bps. Figure 3F is an inverted deletion (INVDEL) composed of an 800bps deletion and a 322bps inversion. It is worth noting that models of Linear for nested SVs are currently built based on features of basic SVs. We can

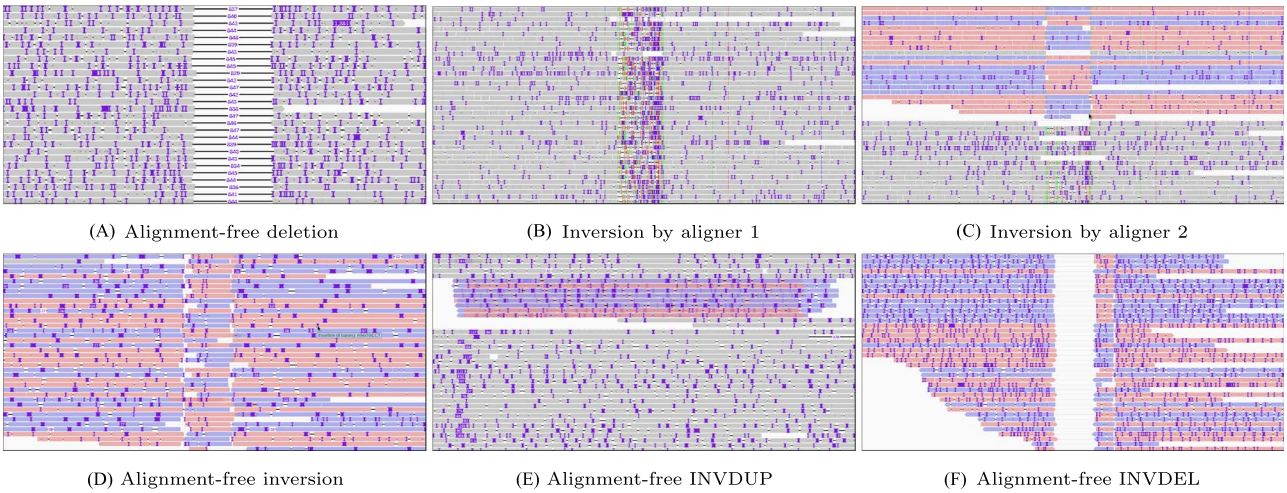



**Figure 3.** (A) is the visualized results of Linear on a subset of HG002 PacBio raw reads comprising a deletion of 844bps. The results are converted to the BAM format and visualized by IGV. 31× events of deletions are detected. The average length of the reported deletion is 843.16bps. The average deviation of the detected endpoints of the deletion from the true ones is only 3.19bps. Other random short gaps in the image are virtual indels, which are discussed in the section of format of alignment-free results, to convert the alignment-free results to the format compatible with IGV. The formal definition of the conversion is denoted by the cigar operation in expression 11. (B and C) are the alignments of Minimap2 and NGMLR of an inversion embedded in HG002 PacBio raw reads. (D) is the alignment-free results of the same inversion. (F and F) are alignment-free results of two nested variants, where (E) contains the so called U-turn of overlapped strands in different colors indicating the event of inverted duplication (INVDUP) and (F) contains the event of inversion flanked by a deletion (INVDEL).

**Table 2.** Summary of SVs detected in the high-quality subsets of HG001 and HG002. The column $Q_1$–$Q_3$ are quantiles of estimated sequencing error. The column 'identified' and 'indicated' are the SVs, which have $\geqslant 5$ supporting reads identified or indicated by Linear. The SVs event in the read is identified or indicated by Linear if the deviation of the detected breakpoints of SVs from the true ones are < 50bps and < 100bps. The column of PBSV are SVs detected by Linear-PBSV

| **Dataset** | **SVs length** [*bps*] | **Sequencing error** [$Q_1, Q_2, Q_3$] | **Identified** [%] | **Indicated** [%] | **PBSV** [%] | **Type** |
|---|---|---|---|---|---|---|
| High-quality subset of Ashkenazim HG002 | < 250 | 0.093, 0.108, 0.169 | 97.8 | 99.1 | 90.7 | INS |
| | | 0.105, 0.151, 0.181 | 97.6 | 99.4 | 96.4 | DEL |
| | [250, 500) | 0.091, 0.106, 0.153 | 98.7 | 99.3 | 92.8 | INS |
| | | 0.088, 0.132, 0.164 | 97.3 | 99.3 | 97.2 | DEL |
| | [500, 750) | 0.094, 0.110, 0.141 | 98.7 | 99.4 | 81.9 | INS |
| | | 0.129, 0.175, 0.209 | 94.3 | 99.1 | 96.1 | DEL |
| | [750, 1000) | 0.090, 0.115, 0.145 | 97.3 | 98.7 | 89.4 | INS |
| | | 0.176, 0.200, 0.260 | 93.1 | 98.7 | 87.5 | DEL |
| | $\geqslant$ 1000 | 0.103, 0.144, 0.199 | 100 | 100 | 88.2 | INS |
| | | 0.145, 0.188, 0.272 | 92.0 | 99.1 | 79.8 | DEL |
| High-quality subset of NA12878 HG001 | < 250 | HiFi reads ≈ 0.5% | 96.4 | 98.8 | 86.4 | INS |
| | | | 98.2 | 99.5 | 88.0 | DEL |
| | [250, 500) | | 96.9 | 98.9 | 98.8 | INS |
| | | | 97.8 | 99.4 | 94.1 | DEL |
| | [500, 750) | | 94.7 | 98.5 | 94.3 | INS |
| | | | 95.6 | 99.2 | 87.7 | DEL |
| | [750, 1000) | | 95.0 | 99.4 | 97.3 | INS |
| | | | 94.6 | 99.0 | 100 | DEL |
| | $\geqslant$ 1000 | | 95.2 | 99.6 | 97.7 | INS |
| | | | 94.4 | 98.0 | 98.4 | DEL |

further optimize the models provided that high-quality datasets of nested SVs are available, and thus, the performance would be substantially improved.

## Evaluation of computational performance

Computational efficiency is one of the key measurements of software. Without loss of generality, we evaluated the computational performance of Linear and mainstream long-read aligners based on a subset of HG002 from https://ftp-trace.ncbi.nlm.nih.gov/giab/ftp/data/AshkenazimTrio/HG002_NA24385_son/PacBio_MtSinai_NIST/PacBio_fasta with the id of 'm150225_013529_42156'. The subset comprises 87999 PacBio raw reads of 7487bps on average. We use raw reads for the assessment since raw reads are commonly more compute-intensive than HiFi reads. We also use the PacBio HiFi reads 'm64011_190830_220126' of HG002 from the https://ftp-trace.ncbi.nlm.nih.gov/giab/ftp/data/AshkenazimTrio/HG002_NA24385_son/PacBio_CCS_15kb_20kb_chemistry2/reads/ for the assessment of SKSV, since SKSV does not support PacBio raw reads. We use the GRCH38 reference genome and run each software with eight threads on

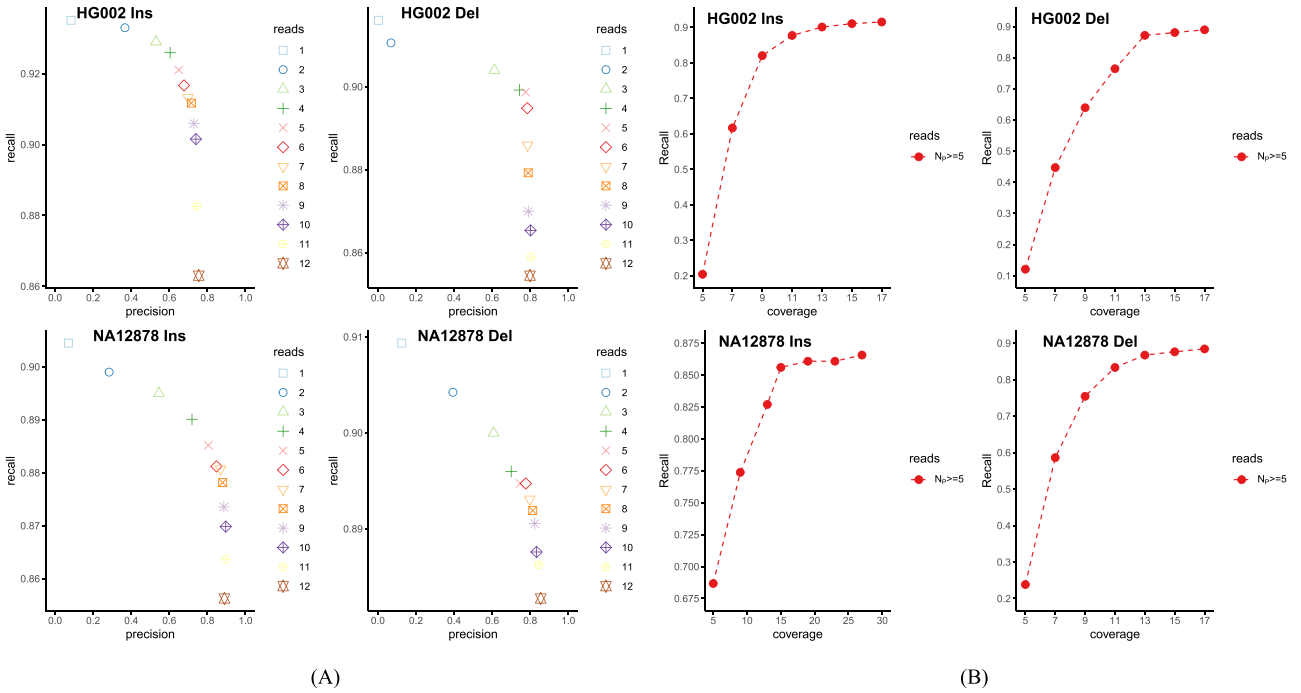



**Figure 4.** Recall and precision of Linear-PBSV on HG002 PacBio raw reads and NA12878 PacBio CCS reads. (A) is the precision versus recall regarding coverage of reads. (B) is the coverage versus recall. SVs are supported by ⩾ 5 reads.

**Table 3.** Recall and Precision of Linear-cuteSV, Minimap2-cuteSV and SKSV-SKSV on HG002 PacBio raw and HiFi reads. Highlighted numbers are the best. SKSV is an SV detector integrating the modified cuteSV for SVs calling. It is designed for PacBio HiFi reads. Thus, rows for PacBio raw reads are empty

| Dataset | Coverage | SVs calller | Recall[%] | | | Precision[%] | | |
|---|---|---|---|---|---|---|---|---|
| | | | Linear | Aligner | SKSV | Linear | Aligner | SKSV |
| HG002 Raw | 72 | cuteSV | **93.21** | 67.98 | / | 75.6 | **77.84** | / |
| HG002 HiFi | 96 | cuteSV | **70.27** | 58.37 | / | 89.15 | 97.10 | / |
| | | SKSV | / | / | 59.51 | / | / | **97.70** |

a machine equipped with 32 cores. We evaluated the memory footprint and runtime including indexing genomes and running the long reads.

Figure 5A is the maximum resident memory of each software. The memory consumption commonly depends on the size of reference genomes rather than the size of the dataset of reads. Linear approximately uses 7GBytes to run long-reads with GRCH38 reference genomes. It is the most memory-efficient in the assessment. Supplementary Figure 5 also shows the maximum resident memory of other software. Figure 5B is the runtime of each software. Supplementary Figure 6 is the runtime of other long-read aligners. For comparison, we show the ratio of the runtime of the software being evaluated and the minimum runtime of all the software. Minimap2 and NGMLR are two aligners explicitly optimized for long reads, and thus, they are significantly faster than other aligners. Linear and SKSV are significantly faster than the two aligners. Linear is 14 and 75 times faster than the aligners.

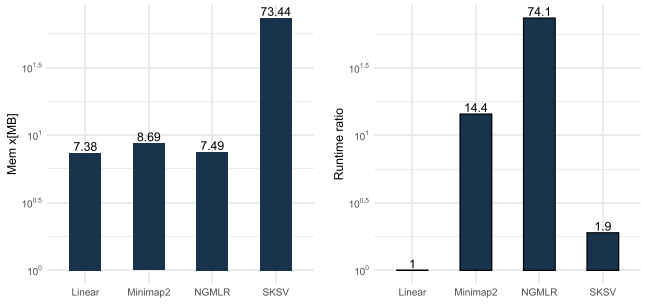



**Figure 5.** The maximum resident memory and runtime each software takes to run long reads with the GRCH38 reference genomes. The vertical axis 'runtime ratio' in (B) equals to $T_x/T_M$, where $T_x$, $T_M$ are the runtime of the software being evaluated and the minimum runtime among all software.

## Discussion

Linear is an alignment-free framework compatible with existing long-read software. To this end, we unified the interface of alignment-based and alignment-free methods and extended the standard SAM/BAM format. The assessments in this work suggest that software such as samtools, SVs callers and visualization tool IGV can directly process alignment-free results of Linear. Furthermore, Linear is an extensible and efficient framework. We use a multi-layered model to simplify the integration and evaluation of long-read SVs models. It is worth noting that we optimized fundamental data structures not discussed here to balance the overall computational performance of memory usage

and runtime. It is more efficient and is less affected by computational limitations than most other software. We conducted assessments on simulated and real datasets to evaluate Linear. The assessments suggest alignment-free methods can improve the recall and computational efficiency of long-read SV detection. It is particularly more effective than aligners in resolving nonlinear SVs, such as inversions and duplications, embedded in long reads.

In this work, we did not evaluate nested SV detection on a large scale due to lacking 'gold-standard' datasets. The assessment is complicated by the fact that benchmark datasets may be missing SVs in the annotation. We also noticed that Linear is probably overconfident in some highly repeated regions. Its performance for repetitive regions is not as good as for non-repeating regions. Thus, we suggest users use more coverage of reads when applying Linear to identifying SVs in repetitive regions. It is because SVs models in Linear are currently for generic usage, which is developed based on the study of basic SVs of non-repeating ones. We did not fully optimize the models for other types of SVs, such as repeats. However, as mentioned, Linear is based on a multi-layered framework which is flexible to extend new models. For future work, it is possible to integrate more specific and effective models, such as machine learning, to enhance the performance of detecting SVs in repetitive regions. Hence it is promising to resolve diverse long-read SVs better as more and more novel models are continuously employed.

# Materials and methods

## Overview of linear

We apply a multi-layered framework ($L_n$) to Linear. It unifies the interface of alignment and alignment-free models and simplifies the evaluation and integration of models.

Denote function $f_i : R_i \rightarrow R_{i+1}$ the ith layer of $L_n$. Then, $L_n$ is given by

$$L_n = \prod_{i=1}^{n} f_i,$$

where $R_{i+1}$ is a subset of $R_i$ and $R_1$ is the Cartesian coordinate system composed of the references and the read.

Furthermore, we define the following metrics to evaluate the overall performance of $L_n$. The error rate of $L_n$ denoted by $E_n$ is given by

$$E_n = 1 - \prod_{i=1}^{n}(1 - e_i),$$

where $e_i \in [0, 1]$ is the error rate of $f_i$.

The deviation of $L_n$ denoted by $D_n$ is given by

$$D_n = \frac{|R_{n+1}|}{\sqrt{|R_1|}} - 1,$$

where $d_i = |R_{i+1}|/\sqrt{|R_i|} - 1$ is the deviation of $f_i$ and $|R_i|$ is the size of $R_i$. Particularly, $d_i \approx 0$ if $f_i$ is the pairwise alignment and $d_i \geqslant 0$ if $f_i$ is an alignment-free function.

The complexity of $L_n$ is given by

$$O_n = \sum_{i=1}^{n} o_i \cdot |R_i|,$$

where $o_i$ is the computational complexity of $f_i$,

We implemented 4 main steps based on the framework to resolve long-read SVs:

1. We apply the word frequency fingerprint and dynamic programming (DP) to validate collinear fragments and search SV candidates (Supplementary 3.1).
2. We build the generative model to compute the likelihood for each SVs candidates and construct the graph of events.
3. We apply the generative model with 01*0 fragments to polish the SVs candidates and improve the precision of endpoints.
4. We redefine the SAM/BAM for the alignment-free results and adapt the results to existing pipelines.

## Generative model for SVs

We develop the generative model to compute the likelihood of assembling SVs. Supplementary Figure 7 shows four SV types built in the generative model. The model is more extensible and accurate in identifying SVs in secondary short fragments than models in mainstream aligners designed for simple high-quality long fragments.

Denote $A_i$ the assembly of $i$ fragments $a_1, a_2, ..., a_i$. Denote $r_1, r_2, ..., r_i$ the subsequence of read sequenced from $a_1, a_2, ..., a_i$ and $R_i$ the subsequence sequenced from $A_i$. The fragment here refers to the object that can be assembled, such as $k$−mers, sub-alignments, etc. The fragment depends on a group of parameters, such as length, and sequencing error, based on which we can compute the likelihood of the occurrence of corresponding subsequences. Assuming each $r_j, 1 \leqslant j \leqslant i$ is sequenced independently, the likelihood of at least one subsequence $r_j$ is sequenced from $A_i$ is given by

$$\begin{aligned} L_i &= L(A_i; R_i) \\ &= \left(1 - p(r_i|a_i)\right) \cdot L_{i-1} + p(r_i|a_i). \end{aligned} \tag{1}$$

Assuming each fragment $a_i$ is composed of a map $m$ and an independent gap $g$ at 5′ end, $p(r_i|a_i)$ is approximated by

$$\begin{aligned} p\left(r_i|a_i\right) &= p_{g,m}(r_i|a_i) = p_g(r_i|a_i)p_m(r_i|a_i) \\ &\approx p_g\left(l_{g,i}\right) p_m\left(l_{m,i}\right), \end{aligned} \tag{2}$$

where $p_{g,m}$ is the joint probability of the gap and map of $r_i$. $p_g$, $p_m$ are probabilities of the gap and map of $r_i$. $l_{g,i}$ and $l_{m,i}$ are the lengths of gaps and maps of $r_i$. Though $p(r_i|a_i)$ depends on other parameters such as the sequencing error, we assume the length $l_{g,i}$ and $l_{m,i}$ are the main parameters to simplify the model. Hence, we use $p_g\left(l_{g,i}\right) p_m\left(l_{m,i}\right)$ for approximation in expression 2.

Assuming there are two types of gaps, regular gaps and gaps of SVs, $p_g(l)$ is given by

$$p_g\left(l\right) = p(l, r) + p(l, v) - p(l, r)p(l, v), \tag{3}$$

where $p(l, r)$ and $p(l, v)$ are the probabilities of regular gaps and gaps of SVs. Since $l \geqslant 0$, we use the cumulative distribution function (CDF) of gamma distribution denoted by $F(x; \alpha, \beta)$ to model the distribution of the length of the regular gap $p(l|r)$. $\alpha$ and $\beta$ depend on parameters such as the sequencing error, length of the fragment and sampling frequency. For fragments of $k$-mers, the empirical distributions of the parameters are shown in Supplementary Figures 8–10. Furthermore, $\alpha$ and $\beta$ corresponding

to the parameters are listed in supplementary table 1. Then, $p(l, r)$ is given by

$$p(l, r) = p(l|r)p(r) = \omega_r \left(1 - F\left(l, \alpha_r, \beta_r\right)\right),$$

where $\omega_r = p(r)$ is the prior probability that a randomly given gap is a regular gap.

Assuming the gap of SVs comprising indel (insertion or deletion) and inversion, the probability $p_\upsilon$ is given by

$$p_{l,\upsilon} = p_{l,inv} + p_{l,indel} - p_{l,inv}p_{l,indel}, \tag{4}$$

where $p_{l,inv} = p(l, inv)$ and $p_{l,indel} = p(l, indel)$ are the probabilities of the inversion and the indel of length $l$ for simplicity. And $p_{l,indel}$ is given by

$$p_{l,indel} = p_{l,ins}(1 - p_{l,del}) + p_{l,del}(1 - p_{l,ins}), \tag{5}$$

where $p_{l,ins}$ or $p_{l,del}$ is the probability of a gap is an insertion or a deletion and its length is $l$.

Given the gap, denote $l_x$ and $l_y$ the length of corresponding subsequence of genome and read. Since $l_x - l_y \in (-\infty, +\infty)$ in the case of indels. We apply the normal distribution to model $p_{l_x,l_y|ins}$ and $p_{l_x,l_y|del}$. Assuming priors $p_{ins} = \omega_{\upsilon,1}$ and $p_{del} = \omega_{\upsilon,2}$, then $p_{l_x,l_y,ins}$, $p_{l_x,l_y,del}$ are given by

$$\begin{cases} p_{l_x,l_y,ins} = p_{ins}p_{l_x,l_y|ins} = \omega_{\upsilon,1}\Phi(l_y - l_x; \mu_1, \delta_1) \\ p_{l_x,l_y,del} = p_{del}p_{l_x,l_y|del} = \omega_{\upsilon,2}\Phi(l_x - l_y; \mu_2, \delta_2), \end{cases} \tag{6}$$

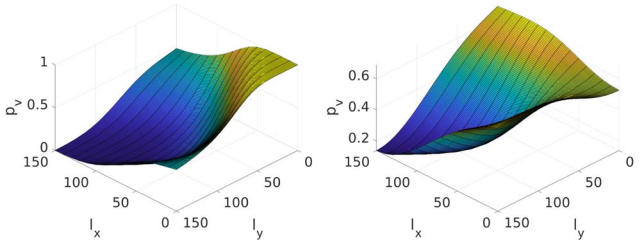
where $\Phi$ is the CDF of normal distribution, $\mu_1$ $\mu_2$ and $\delta_1$ $\delta_1$ are means and SDs of $l_y - l_x$ and $l_x - l_y$ for insertion and deletion.

Genuine inversion may contain embedded copy number variation or be flanked by duplications [43]. Here we assume the distribution of the gap length is irrelevant to the strand of the sequence for generic modeling, while more accurate model can be updated based on thorough study in the future. The distribution of gap length is supposed to be identical whether the sequence is inverted or not. Thus, we use the gamma distribution to model $p_{l_x,l_y,inv}$. Denote the prior of the inversion $p_{inv} = \omega_{\upsilon,3}$, then the probability of the inversion $p_{l_x,l_y,inv}$ is given by

$$\begin{aligned} p_{l_x,l_y,inv} &= p_{l_x,l_y|inv}p_{inv} \\ &= \omega_{\upsilon,3}\left(1 - F(l_x + l_y; \alpha_{inv}, \beta_{inv})\right). \end{aligned} \tag{7}$$

For the probability of the map denoted by $p_m(l)$, assuming the length of the read is $L$, we use variable $l/L$ to compute the likelihood of the corresponding subsequence of map. Though the Beta family is theoretically better to model $l/L \in [0, 1]$. We use $p_m(l) = l/L$ instead to simplify the computation.

In practice, the CDF for each type of the SV discussed above is approximated and stored in a table to improve the computational efficiency. Figure 6 shows two examples of $p_\upsilon$, namely the probability of SVs in expression 4, regarding priors $\omega_{\upsilon,*}$. $p_\upsilon$ in Figure 6A has a higher probability for the regular gap as well as the gap of inversions since $p_\upsilon$ decreases monotonically as $l_x$ or $l_y$ increases. $p_\upsilon$ in Figure 6B has a higher probability for indels since $p_\upsilon$ is maximized at $l_x l_y = 0$. For instance, $p_\upsilon$ of an insertion gap, whose $l_x$ = 10bps and $l_y$ = 150bps, is larger in Figure 6B than in Figure 6A.




(A) $\omega_{\upsilon,*} = 0.45, \omega_{\upsilon,3} = 1.00$, (B) $\omega_{\upsilon,*} = 0.65, \omega_{\upsilon,3} = 0.75$

**Figure 6.** $p_\upsilon$ of $l_x, l_y \in [0, 150)$ with different priors $\omega$, where $* = 1, 2$ and $\mu_1 = \mu_2, \delta_1 = \delta_2$.

```
Insertion 1bp:  TTC-GCC  v_n   = h(TTCGC)⊕h(TCAGC) = 00,10,01,00,00
                || ||
                TCAGC    v_n+1 = h(TCGCC)⊕h(TCAGC) = 00,00,10,01,00

Mismatch  1bp:  TTGGC    v_n   = h(TTGGC)⊕h(TTTGC) = 00,00,01,00,00
                || ||
                TTTGC

Deletion  1bp:  TGACGT   v_n   = h(TGACG)⊕h(TGAGT) = 00,00,00,11,10
                ||| ||
                TGA-GT   v_n+1 = h(GACGT)⊕h(TGAGT) = 01,10,01,00,00
```

**Figure 7.** Detecting three types of $01 * 0$ patterns comprising one gap by counting leading and trailing zeros colored in the figure with bitwise operations.

The assembly $\hat{A}$ maximizing the likelihood function is given by

$$\hat{L}(\hat{A}; r) = \max_{i=1,2,..} \max_{j=1,2,..} L_{i,j}(A_{i,j}; r_{i,j}),$$

where $L_{i,j}$ is the $j$th $L_i$, which has $i$ fragments, defined in the expression 1. We apply DP to compute $\hat{L}$. The computation constructs a graph of $A_{i,j}$ whose vertices are $a_{i,j}$. We compute all $L_{i,j}$ corresponding to $A_{i,j}$ and find the maximum value.

The sequencing error is one of the challenges for alignment and alignment-free approaches. We apply different fragments in each layer $f_i$ to reduce the deviation ($d_i$). We specifically use the $01 * 0$ pattern, which is a pair of matched $k$-mers containing one gap, to approximate the most common $1 - 2$bps sequencing error in long reads [14]. $01 * 0$ patterns can be effectively computed by using bitwise operations. For $k$-mers $s$ and $s_n$ starting from the $n$th base, denote $h(*)$ the function hashing the $k-$mer to an integer. For instance, $h(A) = 00$, $h(C) = 01$, $h(G) = 10$, $h(T) = 11$ in binary. Denote $v_n = h(s) \oplus h(s_n)$ the binary exclusive or values. Denote $l_n$ and $t_n$ the leading zero (clz) and the trailing zero (ctz) of $v_n$ in binary. Then, $(s, s_n)$ is a $01 * 0$ matched pattern if

$$\begin{cases} \text{Insertion:} & t_n + l_{n+1} + 2 - 2k = 0 \\ \text{Mismatch:} & t_n + l_n + 2 - 2k = 0 \\ \text{Deletion:} & l_n + t_{n+1} - 2k = 0. \end{cases} \tag{8}$$

In the implementation, clz and ctz are computed by efficient de Bruijn sequence. Figure 7 illustrates the computation of $01 * 0$ pattern. We use the score metric (Supplementary 3.2) to clip the assembly of fragments.

## Format of alignment-free results

We extend the standard SAM/BAM to enable alignment-based software to utilize alignment-free results. We denote SAM* and $SAM_0^*$ the new and the standard SAM in the following discussion for simplicity. SAM* is a superset of the $SAM_0^*$. It supports both

**Table 4.** Columns of SAM*. Sup stands for the supplementary

| Col | Filed | Description | Support |
|---|---|---|---|
| 1 | QNAME | Query template NAME | Yes |
| 2 | FLAG | Bitwise FLAG | Yes |
| 3 | RNAME | Reference sequence NAME | Yes |
| 4 | POS | 1-based leftmost POSition | Yes |
| 5 | MAPQ | MAPping Quality | Yes |
| 6 | CIGAR | CIGAR string | Extended |
| 7 | RNEXT | Ref name of the mate/next read | Yes |
| 8 | PNEXT | Pos of the mate/next read | Yes |
| 9 | TLEN | Observed Template LENgth | Yes |
| 10 | SEQ | Segment SEQuence | Extended Sup 3.3 |
| 11 | QUAL | Phred-scaled base QUALity+33 | Yes |
| 12 | TAG | Optional tags | Extended Sup 3.5 |

alignment and alignment-free results. Moreover, the SAM* of alignment and the $SAM_0^*$ of alignment are identical. Columns of SAM* are shown in Table 4.

### *Cigar of alignment-free result*

We define the SAM* cigar as follows. Denote $p_1 = (x_1, y_1)$ and $p_2 = (x_2, y_2)$, two points in the Cartesian coordinate system of the reference and the read, where $x_1 \leqslant x_2$ and $y_1 \leqslant y_2$. Denote $\overline{p_1p_2}$ the arbitrary alignment from $p_1$ to $p_2$. Let $c_D$, $c_I$ and $c_M$ be the sum of lengths of deletions, insertions and (mis)matches of $\overline{p_1p_2}$, we have expression 9

$$\begin{cases} c_D + c_I = x_2 - x_1 + y_2 - y_1 - 2c_M \\ c_D - c_I = x_2 - x_1 - y_2 + y_1 \\ c_M \leqslant \min(x_2 - x_1, y_2 - y_1). \end{cases} \tag{9}$$

We then define the virtual alignment of $p_1$ and $p_2$ based on the expression. There are two types of deletions and insertions, namely deletions and insertions of sequencing errors or SVs. Thus, $c_D + c_I = l_v + l_e$, where $l_v$ and $l_e$ are estimated lengths of deletions and insertions of SVs and sequencing errors. Since SVs are rare in most sequences ($l_v = 0$). We define the virtual alignment of $\overline{p_1p_2}$ as the one which minimizes $l_v + l_e$. It can be proved that $l_v + l_e$ is minimized when $c_M = \min(x_2 - x_1, y_2 - y_1)$, namely

$$\begin{cases} c_M = x_2 - x_1 \\ c_D = 0 \\ c_I = y_2 - y_1 - c_M \end{cases} \text{or} \begin{cases} c_M = y_2 - y_1 \\ c_D = x_2 - x_1 - c_M \\ c_I = 0. \end{cases} \tag{10}$$

Assuming 2 cigars are used for $\overline{p_1p_2}$, then

$$\overline{p_1p_2} = c_M M c_{D,I} G \quad \text{or} \quad c_{D,I} G c_M M, \tag{11}$$

where $M \in \{'=', 'X', 'M'\}$ and $G \in \{'D', 'I'\}$. We use 2 cigars due to the following considerations.

1. First, virtual alignments of $4, 6, 8\ldots$ cigars are identical to those of 2 cigars.
2. Second, virtual alignment of $3, 5, 7\ldots$ cigars are ambiguous. Using 3 cigars for points $p_1$, $p_2$ and $p_3$ for instance, cigars are $\overline{p_1p_2} = c_1Mc_2Ic_3M$, $\overline{p_2p_3} = c_4Mc_5Ic_6M$. Then, $\overline{p_1p_2p_3} = c_1Mc_2I(c_3 + c_4)Mc_5Ic_6M$, where $p_2$ is incorrectly omitted since $c_3M$ and $c_4M$ are merged.

Supplementary Figures 11 and 12 are two examples of SAM* and cigars defined in the expression 11. And Supplementary Figure 13 further discusses the estimated deviation of the virtual alignment.

**Key Points**

Our main contributions in this paper are as follows:

- We propose the framework grounded in statistics for resolving structural variants (SVs) in long reads to eliminate limitations faced by conventional approaches. To our knowledge, it is the first approach that can resolve structural variants with completely alignment-free models.
- We propose a new extension for SAM/BAM format to connect existing long-read analysis tools to novel statistical models. The new extension is promising to be used by other research, which applies novel methods including machine learning to long-read analysis.
- Our approach has achieved state-of-the-art performance for resolving SVs. It is highly flexible and sensitive for identifying SVs hidden in very noisy sequences.
- Linear is ultrafast. It is orders of magnitude faster than conventional pipelines for the detection of SVs in long reads. It has the potential for wide-ranging applications in population-scale long-read research.

## Supplementary data

Supplementary data are available online at http://bib.oxfordjournals.org/.

## Author contribution statement

K.R. and C.P. designed research; C.P. performed the research; R.R. and D.H. contributed to the source code and data.

## Acknowledgments

We thank the Sequence Analysis library (SeqAn) for providing algorithms and data structures support. We thank the German Network for Bioinformatics Infrastructure (de.NBI) for supporting the development of SeqAn. We thank the China Scholarship Council (CSC) and Intel® Parallel Computing Center (IPCC) for financing the work and providing the hardware support.

## Funding

China Scholarship Council (CSC) and Intel® Parallel Computing Center (IPCC) Program at Freie Universität Berlin.

## Code and data availability

The source code and data for the assessment underlying this article are available at https://github.com/xp3i4/linear and https://ftp.imp.fu-berlin.de/pub/linear/assessment_data_raw.

## References

1. Sirén J, Monlong J, Chang X, *et al.*Pangenomics enables genotyping of known structural variants in 5202 diverse genomes. *Science* 2021; **374**(6574): abg8871.

2. Jakubosky D, Smith EN, D'Antonio M, *et al.*Discovery and quality analysis of a comprehensive set of structural variants and short tandem repeats. *Nat Commun* 2020; **11**(1): 2928.
3. Bianco S, Lupiáñez DG, Chiariello AM, *et al.*Polymer physics predicts the effects of structural variants on chromatin architecture. *Nat Genet* 2018; **50**(5): 662–7.
4. Brandler WM, Antaki D, Gujral M, *et al.*Paternally inherited cis-regulatory structural variants are associated with autism. *Science* 2018; **360**(6386): 327–31.
5. Weischenfeldt J, Dubash T, Drainas AP, *et al.*Pan-cancer analysis of somatic copy-number alterations implicates *IRS4* and *IGF2* in enhancer hijacking. *Nat Genet* 2017; **49**(1): 65–74.
6. Duvdevani MP, *et al.*Whole-genome sequencing reveals complex chromosome rearrangement disrupting NIPBL in infant with Cornelia de Lange syndrome. *Am J Med Genet A* 2020; **182**(5): 1143–51.
7. Bowden R, Davies RW, Heger A, *et al.*Sequencing of human genomes with nanopore technology. *Nat Commun* 2019; **10**(1): 1869.
8. Goto K, Pissaloux D, Durand L, *et al.*Novel three-way complex rearrangement of TRPM1-PUM1-LCK in a case of agminated Spitz nevi arising in a giant congenital hyperpigmented macule. *Pigment Cell Melanoma Res* 2020; **33**(5): 767–72.
9. Bajaj P, Richardson JO, Paesani F. Ion-mediated hydrogen-bond rearrangement through tunnelling in the iodide–dihydrate complex. *Nat Chem* 2019; **11**(4): 367–74.
10. Tattini L, D'Aurizio R, Magi A. Detection of genomic structural variants from next-generation sequencing data. *Front Bioeng Biotechnol* 2015; **3**.
11. Amarasinghe SL, Su S, Dong X, *et al.*Opportunities and challenges in long-read sequencing data analysis. *Genome Biol* 2020; **21**(1): 30.
12. Niedringhaus TP, *et al.*Landscape of next-generation sequencing technologies. *Anal Chem* 2011; **83**(12): 4327–41.
13. Metzker ML. Sequencing technologies–the next generation. *Nat Rev Genet* 2010; **11**(1): 31–46.
14. Weirather JL, de Cesare M, Wang Y, *et al.*Comprehensive comparison of Pacific biosciences and Oxford Nanopore technologies and their applications to transcriptome analysis. *F1000Research* 2017; **6**:100.
15. Wenger AM, Peluso P, Rowell WJ, *et al.*Accurate circular consensus long-read sequencing improves variant detection and assembly of a human genome. *Nat Biotechnol* 2019; **37**(10): 1155–62.
16. Koren S, *et al.*Canu: scalable and accurate long-read assembly via adaptive k-mer weighting and repeat separation. *Genome Res* 2017; **27**(5): 722–36.
17. Mahmoud M, Gobet N, Cruz-Dávalos DI, *et al.*Structural variant calling: the long and the short of it. *Genome Biol* 2019; **20**(1): 246.
18. Mitsuhashi S, Matsumoto N. Long-read sequencing for rare human genetic diseases. *J Hum Genet***65**(1): 11–9. January 2020. Number: 1 Publisher: Nature Publishing Group.
19. Sakamoto Y, Sereewattanawoot S, Suzuki A. A new era of long-read sequencing for cancer genomics. *J Hum Genet***65**(1): 3–10. January 2020. Number: 1 Publisher: Nature Publishing Group.
20. Chaisson MJ, Huddleston J, Dennis MY, *et al.*Resolving the complexity of the human genome using single-molecule sequencing. *Nature* 2015; **517**(7536): 608–11.
21. Valle-Inclan JE, Stangl C, de Jong AC, *et al.*Optimizing Nanopore sequencing-based detection of structural variants enables individualized circulating tumor DNA-based disease monitoring in cancer patients. *Genome Med* 2021; **13**(1): 86.
22. Vollger MR, Dishuck PC, Sorensen M, *et al.*Long-read sequence and assembly of segmental duplications. *Nat Methods* 2019; **16**(1): 88–94.
23. Tian L, Shao Y, Nance S, *et al.*Long-read sequencing unveils IGH-DUX4 translocation into the silenced IGH allele in B-cell acute lymphoblastic leukemia. *Nat Commun* 2019; **10**(1): 2789.
24. Sanchis-Juan A, Stephens J, French CE, *et al.*Complex structural variants in Mendelian disorders: identification and breakpoint resolution using short- and long-read genome sequencing. *Genome Med* 2018; **10**(1): 95.
25. Gong L, Wong C-H, Cheng W-C, *et al.*Picky comprehensively detects high-resolution structural variants in nanopore long reads. *Nat Methods* 2018; **15**(6): 455–60.
26. Huddleston J, Chaisson MJP, Steinberg KM, *et al.*Discovery and genotyping of structural variation from long-read haploid genome sequence data. *Genome Res* 2017; **27**: 677–85.
27. Alser M, Rotman J, Deshpande D, *et al.*Technology dictates algorithms: recent developments in read alignment. *Genome Biol* 2021; **22**(1): 249.
28. Tran Q, Abyzov A. LongAGE: defining breakpoints of genomic structural variants through optimal and memory efficient alignments of long reads. *Bioinformatics* 2021; **37**(7): 1015–7.
29. Berlin K, Koren S, Chin C-S, *et al.*Assembling large genomes with single-molecule sequencing and locality-sensitive hashing. *Nat Biotechnol* 2015; **33**(6): 623–30.
30. Sović I, Šikić M, Wilm A, *et al.*Fast and sensitive mapping of nanopore sequencing reads with GraphMap. *Nat Commun* 2016; **7**:11307–7.
31. Li H. Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM. *arXiv:13033997 [q-bio]* 2013; arXiv: 1303.3997.
32. Marco-Sola S, Moure JC, Moreto M, *et al.*Fast gap-affine pairwise alignment using the wavefront algorithm. *Bioinformatics* 2021; **37**(4): 456–63.
33. Li H. Minimap2: pairwise alignment for nucleotide sequences. *Bioinformatics* 2018; **34**(18): 3094–100.
34. Sedlazeck FJ, Rescheneder P, Smolka M, *et al.*Accurate detection of complex structural variations using single-molecule sequencing. *Nat Methods* 2018; **15**(6): 461–8.
35. Ono Y, Asai K, Hamada M. PBSIM2: a simulator for long-read sequencers with a novel generative model of quality scores. *Bioinformatics* 2021; **37**(5): 589–95.
36. Yang C, *et al.*NanoSim: nanopore sequence read simulator based on statistical characterization. *GigaScience* 2017; **6**(4): gix010.
37. Zook JM, Catoe D, McDaniel J, *et al.*Extensive sequencing of seven human genomes to characterize benchmark reference materials. *Scientific Data* 2016; **3**(1): 160025.
38. Zook JM, Hansen NF, Olson ND, *et al.*A robust benchmark for detection of germline large deletions and insertions. *Nat Biotechnol* 2020; **38**(11): 1347–55.
39. Bartenhagen C, Dugas M. RSVSim: an R/Bioconductor package for the simulation of structural variations. *Bioinformatics* 2013; **29**(13): 1679–81.

40. Liu Y, Jiang T, Su J, *et al*.SKSV: ultrafast structural variation detection from circular consensus sequencing reads. *Bioinformatics* 2021; **37**(20): 3647–9.
41. Jiang T, Liu Y, Jiang Y, *et al*. Long-read-based human genomic structural variation detection with cuteSV. *Genome Biol* 2020; **21**(1): 189.
42. Ho SS, Urban AE, Mills RE. Structural variation in the sequencing era. *Nat Rev Genet* 2020; **21**(3): 171–89.
43. Chaisson MJ, *et al*.Multi-platform discovery of haplotype-resolved structural variation in human genomes. *Nat Commun* 2019; **10**(1): 1784.