







TECHNICAL NOTE

Vulcan: Improved long-read mapping and structural variant calling via dual-mode alignment

Yilei Fu ¹, Medhat Mahmoud ^{2,3}, Vignesh Vaibhav Muraliraman¹, Fritz J. Sedlazeck ^{2,*},[†] and Todd J. Treangen ^{1,*},[†]

¹Department of Computer Science, Rice University, Houston, TX 77251-1892, USA; ²Human Genome Sequencing Center, Baylor College of Medicine, Houston, TX 77030, USA and ³Department of Molecular and Human Genetics, Baylor College of Medicine, Houston, TX 77030, USA

*Correspondence address: Fritz J. Sedlazeck, Human Genome Sequencing Center, Baylor College of Medicine, Houston, TX 77030, USA.

Fritz.Sedlazeck@bcm.edu  <https://orcid.org/0000-0001-6040-2691>; Todd J. Treangen, Department of Computer Science, Rice University, Houston, TX 77005, USA. treangen@rice.edu  <https://orcid.org/0000-0002-3760-564X>

[†]These authors share senior authorship.

Abstract

Background: Long-read sequencing has enabled unprecedented surveys of structural variation across the entire human genome. To maximize the potential of long-read sequencing in this context, novel mapping methods have emerged that have primarily focused on either speed or accuracy. Various heuristics and scoring schemas have been implemented in widely used read mappers (minimap2 and NGMLR) to optimize for speed or accuracy, which have variable performance across different genomic regions and for specific structural variants. Our hypothesis is that constraining read mapping to the use of a single gap penalty across distinct mutational hot spots reduces read alignment accuracy and impedes structural variant detection. **Findings:** We tested our hypothesis by implementing a read-mapping pipeline called Vulcan that uses two distinct gap penalty modes, which we refer to as dual-mode alignment. The high-level idea is that Vulcan leverages the computed normalized edit distance of the mapped reads via minimap2 to identify poorly aligned reads and realigns them using the more accurate yet computationally more expensive long-read mapper (NGMLR). In support of our hypothesis, we show that Vulcan improves the alignments for Oxford Nanopore Technology long reads for both simulated and real datasets. These improvements, in turn, lead to improved accuracy for structural variant calling performance on human genome datasets compared to either of the read-mapping methods alone. **Conclusions:** Vulcan is the first long-read mapping framework that combines two distinct gap penalty modes for improved structural variant recall and precision. Vulcan is open-source and available under the MIT License at <https://gitlab.com/treangenlab/vulcan>.

Keywords: long-read; read mapping; gap penalty; structural variation

Background

The advent of long-read DNA sequencing over the past decade has led to many new insights in genomics and genetics [1–3]. One of the main advantages of long-read sequencing is for human research given the size and complexity of the human genome, and specifically for the detection of structural variation (SV) [1, 2, 4, 5]. SVs are often defined as 50 bp or larger ge-

nomical alterations that can be categorized into five types: deletions (DEL), duplications (DUP), insertions (INS), inversions (INV), and translocations (TRA) [6, 7]. Owing to higher false-positive and false-negative rates in SV detection with short reads, long reads are preferred to accurately detect and fully resolve SVs [6].

In recent years, three types of single-molecule long reads have been established, produced by two sequencing platforms: Pacific Biosciences (PacBio) and Oxford Nanopore Technology

Received: 2 June 2021; Revised: 22 July 2021; Accepted: 29 August 2021

© The Author(s) 2021. Published by Oxford University Press GigaScience. This is an Open Access article distributed under the terms of the Creative Commons Attribution License (<https://creativecommons.org/licenses/by/4.0/>), which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.

(ONT) [3]. The latest PacBio device (Sequel II) [3] sequences not only continuous long reads (CLR) that have error rates of $\leq 10\%$ but also longer average length; it can also produce HiFi reads [8]. The latter is produced by repeatedly sequencing the same molecule multiple times (10–20 kb long), producing a consensus read that lowers the sequencing error down to 1% or even lower [8]. ONT is the other long-read sequencing platform. ONT also offers single-molecule sequencing and can produce ultra-long reads (> 100 kb and ≤ 2 Mb) [9] with drastically reduced cost with respect to HiFi reads but at a higher error rate (3–10%) [10]. In recent years, SVs have been shown as an important type of genomic alteration often leading to more modified base pairs than single-nucleotide variants (SNVs) on their own [6, 8]. Furthermore, SVs have been shown to have an effect on many human diseases and other phenotypes across multiple species [6, 11–13]. Most of the existing SV detection approaches depend on long reads to facilitate the mapping of these reads to a known reference genome.

We define read mapping as the process of performing a pairwise alignment between a read and a reference genome to identify the region of origin for this DNA molecule [14, 15]. Early on BLASR [16] was the method of choice for high-error long-read mapping. Given its advantageous speed, BWA-MEM [17] later emerged as the method of choice to align single-molecule sequencing reads. We have previously shown that while BWA-MEM performs well in aligning these long reads, it produces less optimal alignments in the presence of structural variants (SVs) [2, 18]. This is mainly due to sequencing errors coupled with SV signals in repetitive regions being mixed and causing sub-optimal pairwise alignments, hindering an accurate detection of SV. To circumvent this issue we introduced NGMLR [2], which made use of a convex scoring matrix to better distinguish between read error and SV signal. Using this approach, we were able to achieve high-accuracy SV detection and at a similar speed compared to BWA-MEM. However, as sequencing throughput increased, NGMLR was not fast enough to keep up with the sheer volume of data, thus becoming a bottleneck in the analysis of larger datasets. Minimapp2 [18] has since emerged as a highly efficient long-read mapper, implementing a much faster alignment approach involving extending the traditional affine gap cost model to a two-piece affine gap model [19] and implementing an efficient chaining process. Thanks to these important algorithmic enhancements, minimapp2 achieved a faster runtime at a similar accuracy to state-of-the-art long-read mappers [18]. There exist several other long-read aligners that have prioritized accuracy, sensitivity, or speed, such as MashMap [20], LAST [21], GraphMap [22], and LRA [23]. However, despite promising recent progress exemplified by these methods, there is still room for improvement in long-read mapping [14].

We posit that a single strategy may not be sufficient for those variable regions; we explore in this study whether distinct heuristics implemented in the different mappers perform better or worse in certain organisms or even regions of the genome (e.g., human). The latter is especially relevant if one considers the different mutational rates per specific genomic region due to recombination [24], housekeeping genes [25], and orphan genes [26]. For example, a conserved housekeeping gene will have a very different mutational landscape compared to genes involved in immune responses (e.g., *HLA* [26], *KIR*) or compared to other highly variable genes among the human population (e.g., *LPA* [27], *CYP2D6*).

To cope with these challenges, in this work we describe a unified long-read mapping framework called Vulcan that melds alignment strategies from two different long-read mappers, here

NGMLR and minimapp2. At its core, Vulcan is based on the following straightforward idea: use distinct gap penalties for different mappings between long reads and a reference genome. Notably, Vulcan is the first long-read mapping framework that combines two gap penalty models, as shown in Fig. 1. Vulcan first maps reads starting with the fastest long-read mapper (minimapp2 by default). The key idea behind Vulcan is to identify reads that are sub-optimally aligned on the basis of edit distance (i.e., number of differences between a read and the reference) and then realign them with a more sensitive gap penalty (NGMLR by default). Previous works have shown that edit distance-based approaches may have an effect on effective detection of SVs [28, 29–31]. Here we show that edit distance can be used as a prior for sub-optimally aligned reads, highlighting the utility and accuracy of Vulcan based on NGMLR and minimapp2. We apply Vulcan on simulated and real datasets (HG002) to measure the improvements of our dual-mode alignment approach in both the number of correctly aligned reads and runtime. Furthermore, to showcase the benefit of improved read mappings, we compared SV calling from Vulcan mapped reads to both NGMLR and minimapp2 mapped reads on simulated ONT reads and human ONT and PacBio CLR and HiFi reads.

Data Description

To evaluate Vulcan’s ability to improve structural variant calling, we simulated five types of structural variant in the reference genome (*Saccharomyces cerevisiae* S288C). Specifically, we selected *S. cerevisiae* S288C genome as the reference and added SVs into the genome with SURVIVOR (1.0.7) and simSV [11]; later, we used Nanosim-h (1.1.0.4) [32] to simulate a $10\times$ coverage reads set. We ran NGMLR, minimapp2, and Vulcan on the dataset and used Sniffles (version 1.12) to identify SV. In this experiment we also included other SV types such as DUP, TRA, and INV.

Additionally, we used real data to show the improvements over HG002, a benchmark sample well studied by Genome in a Bottle (GIAB NIST). Here we downloaded ONT, PacBio HiFi, and PacBio CLR datasets for the same sample. The data are available at [33] and have been described in multiple publications [34, 35]. The subsample of coverages (Nanopore $10\times$, $20\times$, $30\times$; PacBio CLR $10\times$, $20\times$; PacBio HiFi $10\times$) was performed with seqtk [36].

Analyses

To demonstrate the ability of Vulcan to improve the overall mapping of long reads and thus to improve the SV detection across organisms we used simulated (*S. cerevisiae* S288C) and real data (human hg19) datasets. For the real datasets we used three distinct long-read technologies (PacBio HiFi and CLR, ONT) [32, 35]. Using these datasets, we evaluated the edit distance improvement after Vulcan’s refinement and SV calling performance (recall, precision, and F1 score). Also, we show that Vulcan reduces computational time against the methods that use convex gap penalty (NGMLR).

Vulcan improves long-read mapping over minimizing read-to-reference edit distance

First, we investigated Vulcan’s ability to identify reads that would benefit from convex gap penalty vs two-piece affine gap penalty by thresholding the reported edit distance from the mappers (see Methods section) and thus minimize the edit distance between the read and mapped location on the reference genome. To accomplish this, we mapped the GIAB HG002 ONT

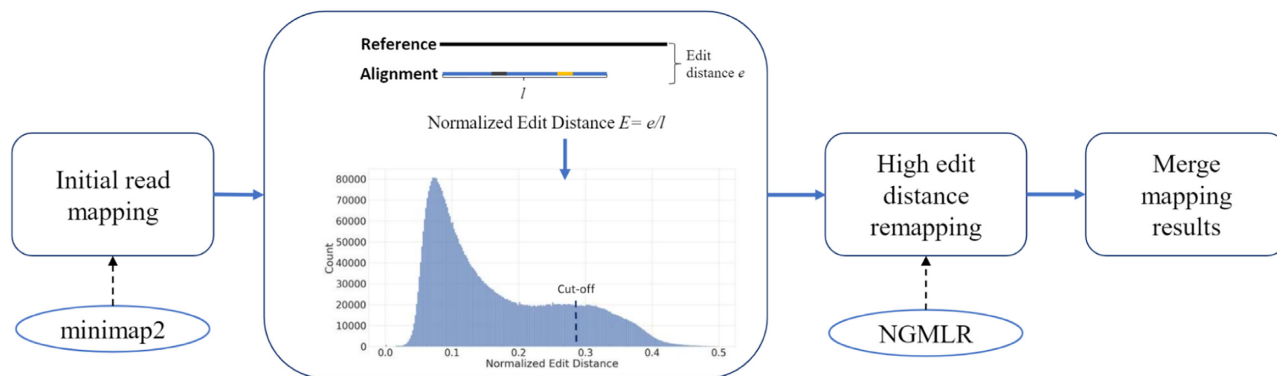


Figure 1: Overview of Vulcan: As step 1, Vulcan takes raw ONT or PacBio reads as input, then uses minimap2 to map them to the provided reference genome. Subsequently, in step 2, Vulcan performs a normalized edit distance calculation (see Methods) to identify the reads with the highest normalized edit distances. In step 3, Vulcan realigns the high edit distance reads with NGMLR. Finally, in step 4 Vulcan merges the minimap2 and NGMLR remapped reads to create a new bam file.

Ultra-long UCSC dataset using minimap2 and investigated the alignments from the reads given their reported edit distance (NM tag).

We benchmarked Vulcan genome-wide to see whether it would improve the overall edit distance compared to minimap2 alone. Figure 2A shows this trend, as Vulcan on the median has a lower normalized edit distance than minimap2 alone. Notably, Vulcan does not recapitulate the overall distribution of edit distance from NGMLR because it only realigns 10% of the reads in this example. Thus, by automatically realigning only 10% of the reads Vulcan significantly improves the alignments in certain regions of the human genome compared to minimap2. These results provide support for our dual-mode alignment strategy implemented in Vulcan to select reads on the basis of their normalized edit distance and then realign these using NGMLR. This strategy seemed to work and indeed improve the representation of SV (Tables 1 and 2).

Vulcan accelerates long-read mapping for SV calling

Next, we evaluated the speed-up of Vulcan compared to minimap2 and NGMLR. As shown in Fig. 3, Vulcan is able to achieve $\leq 2.5\times$ speed-up over NGMLR, from 6.5 CPU hours down to 2.5 CPU hours for the 90% cut-off (default parameter setting for human genome mapping). When increasing the edit distance cut-off percentile, Vulcan CPU time decreases linearly. When comparing minimap2's CPU time we see that Vulcan's default setting only requires ~ 3 times more CPU time compared to >10 times more CPU time required for NGMLR. This highlights Vulcan's ability to drastically reduce NGMLR CPU time and maintain comparable CPU time to minimap2, one of the most efficient long-read aligners that currently exists. The RAM usage of Vulcan 90% cut-off with Nanopore $10\times$ reads is 29.7 GB.

We also show the relative contribution to CPU time for each component in Vulcan (Fig. 3B and C): minimap2, samtools, file parsing, and edit distance calculation with Python, and NGMLR. As expected, NGMLR dominates this breakdown when mapping the reads that are above the Vulcan cut-off (60% in this experiment); the remaining components represent minor contributions to Vulcan's execution time.

SV calling benchmarking

Next, we highlight the finding that NGMLR's SV-aware mappings enable the improved detection of SV (here deletion indicated by

black lines in Integrative Genomics Viewer [37]) compared to the mapping results from minimap2 (Fig. 4A). We see in this example that minimap2 demonstrates a more scattered pattern of the deletion signal across all three regions (Fig. 4A–C). These regions include an INS and a DEL, which induce noisy alignments from minimap2. In contrast, automatically realigning the reads with Vulcan using NGMLR shows a more consistent mapping pattern (Fig. 4B and C). Notably, Vulcan is able to eliminate a false-positive SV call by preferentially selecting a convex gap penalty over the two-piece affine gap penalty (Fig. 4C), highlighting the benefit of trading off increased CPU time (measured in CPU hours) for increased accuracy (measured as fewer false-positive SV calls).

Benchmarking SV calling with Vulcan's mappings on simulated ONT data

To follow up on the previous result, we next benchmarked SV calls based on each of the three mapping strategies: minimap2, NGMLR, and Vulcan. To perform this evaluation, we simulated Nanopore reads from the *S. cerevisiae* S288C genome. As we see in Fig. 5, Vulcan combined with Sniffles offers the highest recall and lowest false discovery rate (FDR) of all three mapping approaches. Next, Fig. 5B highlights that Vulcan has the highest recall for all five SV types. We see that minimap2 has the lowest recall for DUPs on this low-coverage simulated long-read dataset. However, both NGMLR and Vulcan are able to capture the DUP with $>99\%$ recall. We also see that while TRA and INS SV recall is identical for all three mapping approaches, Vulcan mappings help to improve both INV and DEL detection. With respect to precision (Fig. 5C), Vulcan once again performs best across all five SV categories, with NGMLR mirroring Vulcan performance in all cases.

Benchmarking SV calling with Vulcan's mappings on GIAB human data

Given the promising SV calling results based on Vulcan mappings that we discovered in the simulated data, we next evaluated SV calling using Vulcan on real human (hg19) read samples from the GIAB project [35]. Similar to the SV experiment with simulated data, we used Sniffles to call SVs called from human (hg19) reads mapped from each of the three methods: minimap2, NGMLR, and Vulcan. This GIAB dataset allowed us to evaluate against an established ground truth on real hg19 long-read sequencing data. We next describe SV performance for var-

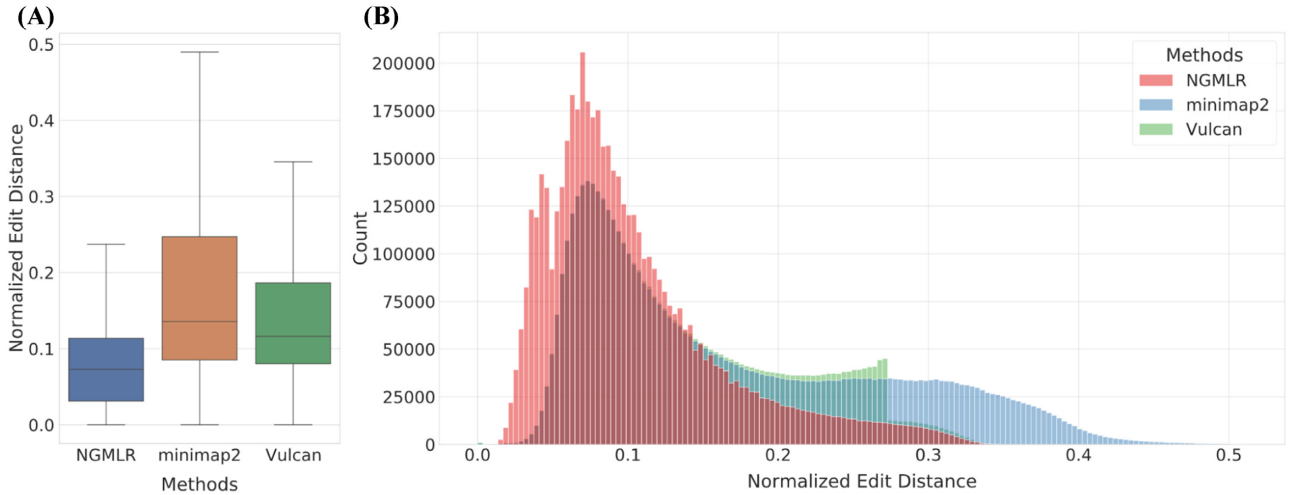


Figure 2: Overall edit distance improvements. A: Normalized edit distance comparison of Vulcan's (green) 90% percentile cut-off, NGMLR (red) and minimap2's (blue) mapping result with human ONT 30X reads. We can see clear evidence that the realignment of only 10% of the reads lead to an improvement in edit distance and thus of the variant calling. B: Distribution of mappings' normalized edit distance from Vulcan (green), NGMLR (red) and minimap2 (blue). Vulcan has a lower edit distance mapping than minimap2 with NGMLR's refinement.

Table 1: Benchmarking SV recall, precision, and F1 on HG002 Human (hg19) ONT reads at varying coverages (10×, 20×, 30×)

Method	Recall, %	Precision, %	F1, %
ONT 10×			
minimap2	78.31	75.59	76.93
NGMLR	77.40	76.65	77.02
Vulcan			
60%	74.64	88.69	81.06
70%	76.66	87.87	81.88
80%	77.66	85.55	81.42
90%	78.29	83.31	80.72
ONT 20×			
minimap2	83.55	76.13	79.67
NGMLR	83.39	76.24	79.66
Vulcan			
60%	83.78	77.71	80.63
70%	83.91	78.53	81.13
80%	83.50	79.57	81.49
90%	83.55	80.65	82.08
ONT 30×			
minimap2	88.74	77.37	82.66
NGMLR	88.47	77.79	82.79
Vulcan			
60%	89.37	79.11	83.93
70%	89.36	79.87	84.35
80%	89.24	80.71	84.76
90%	88.81	81.40	84.94

Various percentile cut-offs for Vulcan were used, including 60%, 70%, 80%, 90%. SV calls based on Vulcan mappings achieve the highest F1 score for various cut-off values.

ious Nanopore coverages (10×, 20×, 30×), PacBio CLR (10×, 20×), and PacBio HiFi (10×) datasets.

Specifically, we tested Vulcan on three different coverages across ONT and PacBio datasets with respect to improving the SV calling ability based on the GIAB SV call sets. Table 1 shows the performance for Vulcan, NGMLR, and minimap2 together with Sniffles to identify SV across the dataset. Similar to the simulated data, we achieve the best SV calling results using Vulcan together with Sniffles. Vulcan provides the most improvement on lower coverage datasets. For the Nanopore 20× cover-

age, which is roughly equivalent to one ONT PromethION Flow cell of a human genome, Vulcan improves F1 score by 3.13% compared to minimap2-based alignments.

We then benchmarked the impact of the normalized edit distance thresholds for the ONT 30× dataset (Table 1). We show that by increasing the cut-off percentile, we realign fewer reads and thus Vulcan exhibits lower overall CPU time. However, this subsequently results in lower SV recall but higher precision. We observed the highest SV recall for Vulcan with a 60% cut-off when realigning the top 40% edit distance reads. SV precision

Table 2: Benchmarking SV recall, precision, and F1 on HG002 Human (hg19) PacBio reads (CLR and HiFi) at varying coverages (CLR 10×, 20×, 30×; HiFi 10×)

Method	Recall, %	Precision, %	F1, %
PacBio CLR 10×			
minimap2	62.85	88.88	73.63
NGMLR	60.11	86.44	70.91
Vulcan			
60%	60.13	89.41	71.90
70%	60.79	90.12	72.61
80%	60.97	90.13	72.73
90%	61.85	89.93	73.29
PacBio CLR 20×			
minimap2	77.76	71.85	74.69
NGMLR	75.74	68.36	71.86
Vulcan			
60%	75.74	74.69	75.21
70%	75.98	75.32	75.65
80%	76.22	75.65	75.93
90%	76.90	75.08	75.98
PacBio CLR 30×			
minimap2	83.71	86.25	84.96
NGMLR	81.79	82.41	82.10
Vulcan			
60%	82.05	86.31	84.12
70%	82.33	87.12	84.66
80%	82.47	87.60	84.96
90%	82.75	87.49	85.05
PacBio HiFi 10×			
minimap2	81.50	90.70	85.85
NGMLR	78.22	86.26	82.04
Vulcan			
60%	77.73	86.04	81.68
70%	77.74	86.19	81.75
80%	76.40	85.75	80.81
90%	76.26	85.73	80.72

Various percentile cut-offs for Vulcan were used, including 60%, 70%, 80%, 90%. Vulcan achieves the highest F1 score on PacBio CLR 20× and 30× reads, with minimap2 achieving the highest F1 score on PacBio CLR 10× and PacBio HiFi 10× reads.

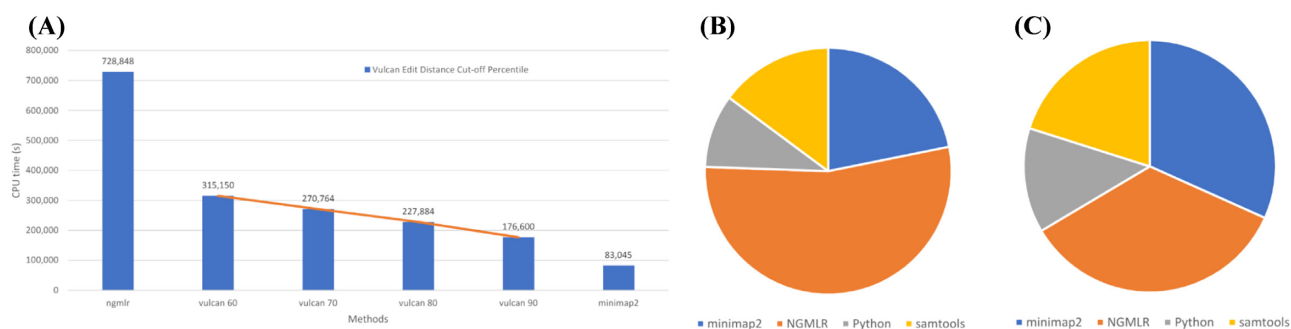


Figure 3: Comparing runtime for Vulcan, NGMLR, and minimap2. The time was measured in terms of CPU time for all programs. **A:** Vulcan achieves an approximately linear acceleration with the increase of the cut-off percentile. With a 90% percentile cut-off, Vulcan only takes approximately one-fourth of NGMLR's CPU time. **B:** The majority of Vulcan's CPU time is spent in running NGMLR on the subset of reads, leading to an improvement of their alignments. **C:** In 90% percentile cut-off, NGMLR only re-aligns 10% of the reads, leading to time usage similar to that of minimap2.

was the highest at a 90% threshold where only the top 10% of the reads are realigned. Notably, across all thresholds, Vulcan performs the best in terms of F1 score. Vulcan by default uses a 90% threshold, yielding $\leq 3.79\%$ improvement in F1 score on low-coverage (10×) ONT data. However, SV calls based on minimap2 mappings achieved the highest recall on 10× coverage (0.02% improvement over Vulcan mappings).

Finally, we investigated Vulcan's performance with respect to Sniffles SV calls on PacBio CLR and HiFi human datasets (Table 2). PacBio CLR and HiFi reads offer a different error profile compared to ONT reads, with PacBio HiFi representing the lowest error rate long reads available to date. As we see in Table 2, SV calls from Vulcan mappings offer the best recall, precision, and F1 score for 20× coverage PacBio CLR data, improving on both

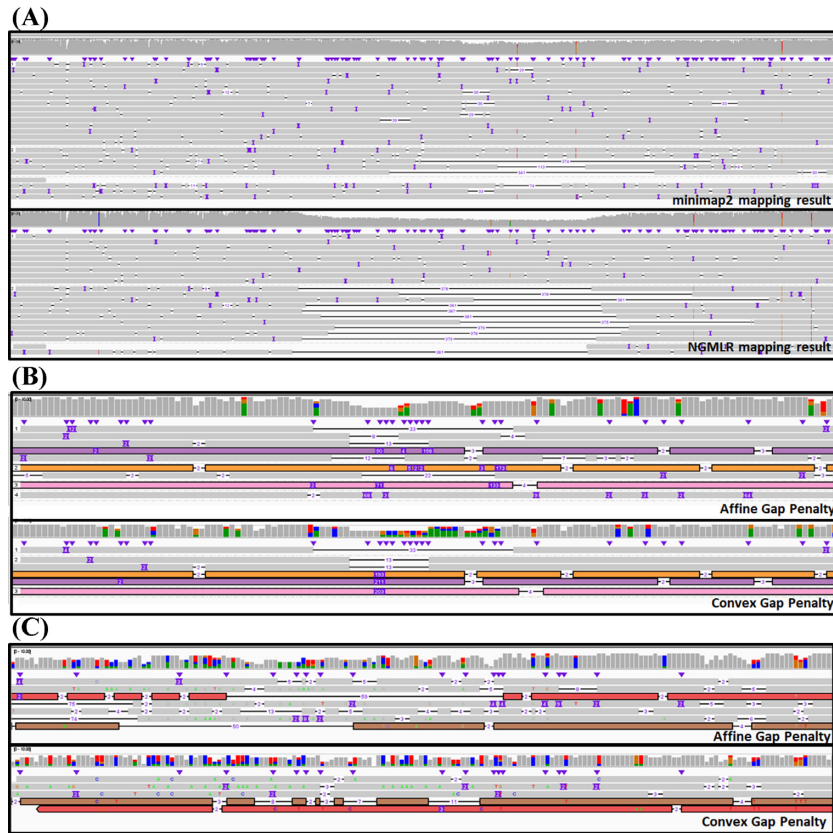


Figure 4: Comparison of the two read mappers used in Vulcan based on 30× ONT data. **A:** An example at chr2: 112,870,823–112,871,894 of reads that show a higher normalized edit distance and thus were automatically realigned with NGMLR. The overall alignments of these reads improved, clearly highlighting a larger deletion at this location compared to the minimap2 alignments. **B:** Another example at chr1: 108,567,498–108,567,633 of automatically aligned reads with Vulcan. The colored reads indicate the same read aligned by the two different methods. The realignment with NGMLR clearly shows a deletion and insertion to be present likely on the two different haplotypes. **C:** Example false-positive SV call improved by Vulcan mapping. This is an example of a false-positive SV call based on minimap2 that would later be resolved with Vulcan's alignment. The region of the genome is on chr1 at 167,9787,40.

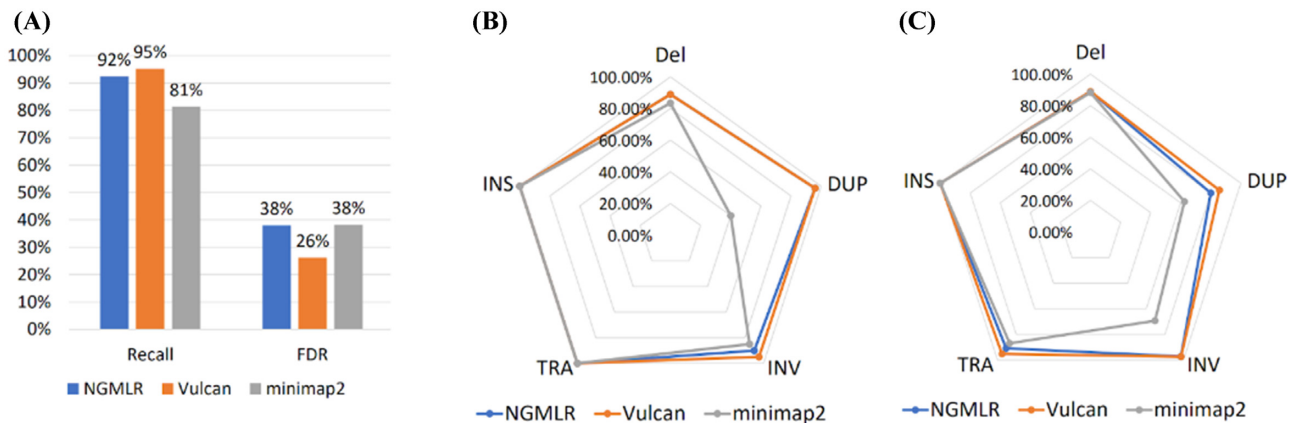


Figure 5: Benchmarking SV calls on simulated structural variants (INS: insertions; Del: deletions; TRA: translocations; DUP: duplications; INV: inversions) with ONT reads simulated from *Saccharomyces cerevisiae*. **A:** Recall and false discovery rate (FDR) of Sniffles' SV calling on simulated Nanopore reads with three different mappers. SV calls on Vulcan mappings offer the highest recall (95%) and lowest FDR (26%). **B:** Recall of different SV types from minimap2, NGMLR, and Vulcan mappings with Sniffles' SV calling on simulated Nanopore reads. **C:** Precision of different SV types from NGMLR, Vulcan, and minimap2's mappings with Sniffles' SV calling on simulated Nanopore reads. NGMLR has similar performance across all SV types, while minimap2 has a lower precision on INVs and DUPs.

NGMLR- and minimap2-based SV calls by >2% in F1 score and nearly a 4% improvement over minimap2 and NGMLR precision. The F1 score improvement is due to the SV calls based on Vulcan offering similar recall to existing approaches but improved pre-

cision. However, when comparing SV recall, we see that Vulcan mappings offer slightly lower performance compared to minimap2, while meeting or exceeding NGMLR recall. We also observed that SV calls based on Vulcan mappings offer a slightly

increased recall rate when the normalized edit distance cut-off increases in the PacBio CLR read dataset, different from the ONT dataset results. One difference between these two datasets is that the coverage of the PacBio CLR dataset is lower, and so the Sniffles minimum read support is set lower. Then when increasing the cut-off percentiles for Vulcan, there remain enough NGMLR mappings to meet or exceed the minimum read number support for SV calling.

Discussion

In this article we introduce Vulcan, a novel long-read mapping tool that leverages dual-mode long-read alignment, which we have shown improves SV calling. Vulcan uses the edit distance information across the mapped reads to rapidly identify regions that are better suited for a convex gap penalty vs two-piece affine gap penalty. The key idea behind Vulcan is that different regions of the genome can benefit from distinct alignment methods (e.g., due to differences in mutation rate), leading to, e.g., improved SV detection. The latter is often highlighted over mismatched reads, indicated by a higher per read substitution and Indels rate [2, 35]. Throughout the Results section we have highlighted the benefits of using a dual-mode alignment approach compared to minimap2 and NGMLR alone; Vulcan not only results in long reads mapped at smaller edit distances, it also improves the recall and precision of SV calling on ONT data.

Our results show that Vulcan runs up to 4 times faster than NGMLR alone and produces lower edit distance alignments than minimap2, on both simulated and real datasets. In addition to improved alignments (Fig. 3), we also show that using Vulcan improves the precision and recall of SV calls for both PacBio CLR and ONT datasets (Tables 1 and 2). Specifically on ONT, Vulcan is able to achieve up to a 5% improvement in F1 score for SV calls (harmonic mean of recall and precision) over the other two mappers, minimap2 and NGMLR. This result not only highlights the benefit of dual-mode alignment, it supports our hypothesis that Vulcan can improve SV calling in human genome samples. We further speculate that Vulcan could improve SNV calling for complex regions. However, the edit distance selection of the reads would need to be adapted for this task and thus the signal would not be that clear. Therefore, we abandoned this benchmarking. Nevertheless, SNV detection around breakpoints or within SV will obviously be improved.

When designing Vulcan, we opted to focus on precision and computational efficiency. The NM tag is required according to SAMtools specifications and contains all the information needed to evaluate alignment quality. Future improvements to this approach may include not counting every difference on the read (i.e., edit distance) but instead only the start of each edit. The latter would count a longer deletion as 1 and not by the length of the event as in the current implementation. Therefore, misalignments that often introduce many smaller events and or substitutions would be more penalized than larger INS or DEL. This could slightly improve the selection process of Vulcan but will lead to longer runtimes because the entire alignment would need to be reconstructed per read. This approach would also consume the majority of the time of our parsing method and thus significantly alter the runtime. Thus, we did not implement this in the current version of Vulcan but will continue to investigate other filtering schemas. The soft clipping also takes place at split reads that are indicative for SV and thus often form at breakpoints of SV. The focus here is on reads that did not get split due to an SV in this region but rather are forced into a continuous alignment.

Such reads will benefit from a realignment step as is done here with Vulcan. We currently do not use MAPQ as a filtering criterion because MAPQ reports the confidence of a read in a region (weighted distance of best vs other potential alignments) [38]. This is indicative for repeats or other regional properties but not for misalignments, or misrepresentation of variants. The issue with correct or wrong representation of SV is more related to the alignment score or chosen alignment algorithm rather than the region. Most of the time NGMLR will not change the location of the read compared to minimap2 but the alignment itself. For example, Fig. 4 shows the same reads in the same region but with a better variant representation. Thus, using the normalized edit distance has been shown to be a robust and rapid approximation to detect these alignment artifacts.

Finally, we note that Vulcan could be used for any combination of long-read mappers that output the edit distance (NM tag) directly within sam/bam file output. Thus, allowing the inclusion of WinnowMap [39], LAST [21], LRA [23], or Duplomap [40] may further exploit our observation that variable gap costs for different read-to-reference mappings provide improved SV calling while offering improved runtimes compared to the more computationally expensive long-read mapping approaches.

Potential Implications

A key finding of this article is that the utilization of dual-mode alignment, combining convex gap costs with two-piece affine gap costs, leads to improvements in alignment edit distance and subsequently SV identification. Notably, we see that SV calling based on minimap2 mappings has low recall for DUPs, compared to near perfect recovery of DUPs with NGMLR and Vulcan. Recently, Jain et al. [39] discuss that the minimizer selection strategy in minimap2 may lead to a degradation in repeat detection. Improved SV calling based on Vulcan's results can be attributed to leveraging the strengths of the long-read alignment strategies found in minimap2 and NGMLR. Vulcan provides the first approach for long-read mapping able to track variable mutation rates and predominant mutation types at certain regions or SV hot spots. The straightforward idea behind Vulcan of adapting alignment gap costs to specific regions of the genome may be found useful for compensating for highly polymorphic regions such as HLA, a 14-Mb section of the human genome that has been at the center of several recent studies [24, 25, 28]. Vulcan takes the first step in leveraging this observation, and we anticipate other mappers for long reads to follow up on this observation. In conclusion, in this study we have shown that combining different long-read alignment strategies improves SV recall and precision of human SV detection and have provided a new open-source software tool (Vulcan) that encapsulates these benefits.

Methods

The main idea behind Vulcan is that we combine the benefits of two popular long-read mapping tools (here NGMLR and minimap2) for improved SV calling. To accomplish this, we first map the reads (sequenced on the ONT or PacBio platform) to a reference genome with minimap2 (2.17-r941), then identify the large edit distance alignments taken from minimap2 mapping results and flag them for realignment with NGMLR (0.2.7). As shown in Fig. 1, Vulcan is composed of 4 main steps: (i) initial read mapping, (ii) normalized edit distance calculation, (iii) high edit distance remapping, and (iv) merging mapping results for downstream SV calling. The first step of the pipeline is to map all the reads against the reference using minimap2 and its preset pa-

parameters suitable for either PacBio or ONT long-read sequences. Subsequently, Vulcan uses the edit distance and scans the reads. The edit distance is the number of substitutions, insertions, or deletions that are different between the read and its region of the reference [39, 41]. The edit distance is captured by the “NM” tag (mandatory tag in sam format) in read mappers. We normalize the edit distance with the overall read length to obtain a ratio that represents the alignment of a given read. By dividing the edit distance by the alignment length of a read, we can normalize it to calculate the number of mismatches given an alignment length; i.e., with longer alignments, we tolerate larger edit distances. And normalized edit distance can be expressed as $E = e/l$, where e is the edit distance and l is the alignment length. We only keep all the primary mappings and gather the normalized edit distances with SAMtools and pysam [42]. Note, the secondary mappings typically have larger edit distances because they have a lower mapping quality than the primary mapping, which may lead to the increase of high edit distance mappings in the distance profile that we generated. With the knowledge of all the normalized edit distances calculated from minimap2’s mapping result, we can now set a percentile cut-off in agreement with the user’s preference (90% as the default, based on experimental results). With the selected percentile cut-off, we can separate reads mapped with minimap2 into two sets: reads that are mapped below the cut-off and reads that are mapped above the cut-off. If we only use raw edit distance, bamtools [42, 43] supports splitting mapped reads via specific tags. However, with normalized edit distances, we instead use pysam, a wrapped Python interface for htslib [42] to calculate the normalized edit distance and split the mapping result. We then extract all the reads above the cut-off and re-map them with NGMLR. Thanks to NGMLR’s ability to accurately remap large edit distance reads, Vulcan is able to improve minimap2’s high edit distance results (in some cases) into read mappings with small edit distances. Finally, we combine the mapping results—specifically, the mapped reads from minimap2 below the cut-off and the remapped reads from NGMLR—into a final merged and sorted BAM file. Vulcan was written in Python 3.8 using the multiprocessing module for multicore support. All versions of software and parameters used in this study are provided in Supplementary Table S1.

Computational benchmarking

To evaluate Vulcan’s computational performance, we assessed the fold speed-up vs NGMLR and compared Vulcan to minimap2. We chose subsampled ONT real data with 10× coverage as test data. In this experiment, we assessed our speed-up under different edit distance cut-offs in Vulcan and compared them with NGMLR and minimap2. We used the `/usr/bin/time` command in Linux to record the program’s wall clock and CPU time. Furthermore, to profile the individual steps of Vulcan, we also counted the time usage per step on the ONT 10× coverage dataset with 90% and 60% percentile normalized edit distance cut-off. In the time benchmarking experiment, the read dataset size is a 62.6 GB fastq file and contains 6,190,519 reads.

Human read dataset structural variant calling evaluation

We used Vulcan on three long-read human genome datasets: ONT Ultra Long reads, PacBio HiFi reads, and PacBio CLR reads [35]. We downloaded these three long-read types from GIAB [34] and mapped them to the human reference genome hg19. Fur-

thermore, we used Sniffles to call SVs from our mapping result, then compared with the ground truth that GIAB provided through truvari (v2.0.0-dev) [35].

Sniffles [2] allows users to define the minimum number of reads supported for the SV calling; we set that parameter as 2 and then use bcftools [44] to further filter the minimum supported read number to achieve the optimal F1 score. We set the minimum read support to be the same for all three methods when the coverage and read type is the same, and the optimal F1 score was preferentially selected for both minimap2 and NGMLR.

The experiment was performed on an Intel® Xeon® Gold 5218 CPU at 2.00 GHz with 64 threads with Ubuntu 18.04 LTS. Total RAM was 300 GB.

Availability of Source Code and Requirements

- Project name: Vulcan
- Project home page: <https://gitlab.com/treangenlab/vulcan>
- Operating system(s): Unix
- Programming language: Python
- Other requirements: Python 3.8 or 3.9
- License: MIT
- RRID:SCR_021657
- biotools:vulcan_mapper

Data Availability

- The *Saccharomyces cerevisiae* 288C reference genome for reads and SV simulation, NCBI:txid559292, is available at [45].
- The Ashkenazim Trio HG002 raw sequence data, and ground truth sets of SVs are available at [33].
- Simulated reads, supporting data and an archival copy of the code is also available via the *GigaScience database*, GigaDB [46].

Additional Files

Supplementary Figure S1: Wall clock time benchmarking of Vulcan, NGMLR, and minimap2 on ONT 10× datasets. A wall clock time benchmarking has been performed to compare the performance of three different methods. From the chart we can infer that Vulcan takes less than two-fifths the time of NGMLR. The experiment was performed on a Nanopore 10× subsample real dataset from the GIAB project.

Supplementary Figure S2: CPU time benchmarking of Vulcan, NGMLR, and minimap2 on PacBio 20× datasets. Vulcan achieves an approximately linear acceleration with the increase of the cut-off percentile. With a 90% percentile cut-off, Vulcan only takes roughly one-fourth of NGMLR’s runtime. The experiment was performed on a PacBio CLR 20× subsampled real dataset from the GIAB project.

Supplementary Table S1: Programs, program versions, and parameters used in this study.

Supplementary Table S2: Accuracy, precision, and F1 score of simulated data.

Abbreviations

bp: base pairs; BWA: Burrows-Wheeler Aligner; CLR: continuous long read; CPU: central processing unit; DEL: deletion; DUP: duplication; FDR: false discovery rate; GIAB: Genome in a Bottle; INS: insertion; INV: inversion; kb: kilobase pairs; Mb: megabase pairs; NCBI: National Center for Biotechnology Information; NIH:

National Institutes of Health; ONT: Oxford Nanopore Technologies; PacBio: Pacific Biosciences; PacBio HiFi: PacBio circular consensus sequencing; RAM: random access memory; SNV: single-nucleotide variation; SV: structural variant; TRA: translocation; UCSC: University of California Santa Cruz.

Competing Interests

The authors declare that they have no competing interests.

Funding

Y.F. is supported in part by funds from Rice University and Ken Kennedy Institute Computer Science Engineering Enhancement Fellowship, funded by the Rice Oil Gas HPC Conference. T.J.T. is supported in part by NIH (1P01AI152999-01). M.M. and F.J.S. are supported by NIH (UM1 HG008898).

Authors' Contributions

All authors conceived the experiment, analyzed the results, and reviewed the manuscript. Y.F. and M.M. conducted the experiment. Y.F. wrote the code. V.V.M. implemented the initial version of the pipeline and conducted experiments. F.J.S. and T.J.T. managed the project.

Conflict of Interest

None declared.

Acknowledgments

The authors thank Dreycey Albin for contributing critical discussion.

References

- Sedlazeck FJ, Lee H, Darby CA, et al. Piercing the dark matter: bioinformatics of long-range sequencing and mapping. *Nat Rev Genet* 2018;**19**(6):329–46.
- 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.
- Goodwin S, McPherson JD, McCombie WR. Coming of age: ten years of next-generation sequencing technologies. *Nat Rev Genet* 2016;**17**(6):333–51.
- Nattestad M, Goodwin S, Ng K, et al. Complex rearrangements and oncogene amplifications revealed by long-read DNA and RNA sequencing of a breast cancer cell line. *Genome Res* 2018;**28**(8):1126–35.
- De Coster W, Weissensteiner MH, Sedlazeck FJ. Towards population-scale long-read sequencing. *Nat Rev Genet* 2021;**22**(9):572–87.
- Mahmoud M, Gobet N, Cruz-Dávalos DI, et al. Structural variant calling: the long and the short of it. *Genome Biol* 2019;**20**(2):246.
- Cameron DL, Di Stefano L, Papenfuss AT. Comprehensive evaluation and characterisation of short read general-purpose structural variant calling software. *Nat Commun* 2019;**10**(1):3240.
- 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.
- Payne A, Holmes N, Rakyan V, et al. Whale watching with BulkVis: A graphical viewer for Oxford Nanopore bulk fast5 files. *Bioinformatics* 2019;**35**(13):2193–8.
- Xiao T, Zhou W. The third generation sequencing: the advanced approach to genetic diseases. *Transl Pediatr* 2020;**9**(2):163–73.
- Jeffares DC, Jolly C, Hoti M, et al. Transient structural variations have strong effects on quantitative traits and reproductive isolation in fission yeast. *Nat Commun* 2017;**8**:14061.
- Beck CR, Carvalho CMB, Akdemir ZC, et al. Megabase length hypermutation accompanies human structural variation at 17p11.2. *Cell* 2019;**176**(6):1310–24.e10.
- Alonge M, Wang X, Benoit M, et al. Major impacts of widespread structural variation on gene expression and crop improvement in tomato. *Cell* 2020;**182**(1):145–61.e23.
- Smolka M, Rescheneder P, Schatz MC, et al. Teaser: Individualized benchmarking and optimization of read mapping results for NGS data. *Genome Biol* 2015;**16**:235.
- Smith TF, Waterman MS. Identification of common molecular subsequences. *J Mol Biol* 1981;**147**(1):195–7.
- Chaisson MJ, Tesler G. Mapping single molecule sequencing reads using basic local alignment with successive refinement (BLASR): application and theory. *BMC Bioinformatics* 2012;**13**:238.
- Li H. Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM. *arXiv* 2013:1303.3997.
- Li H. Minimap2: pairwise alignment for nucleotide sequences. *Bioinformatics* 2018;**34**:3094–100.
- Gotoh O. Optimal sequence alignment allowing for long gaps. *Bull Math Biol* 1990;**52**(3):359–73.
- Jain C, Dilthey A, Koren S, et al. A fast approximate algorithm for mapping long reads to large reference databases. *J Comput Biol* 2018;**25**(7):766–79.
- Kielbasa SM, Wan R, Sato K, et al. Adaptive seeds tame genomic sequence comparison. *Genome Res* 2011;**21**(3):487–93.
- Sović I, Šikić M, Wilm A, et al. Fast and sensitive mapping of nanopore sequencing reads with GraphMap. *Nat Commun* 2016;**7**:11307.
- Ren J, Chaisson MJP. Ira: A long read aligner for sequences and contigs. *PLoS Comput Biol* 2021;**17**:e1009078.
- Duret L, Arndt PF. The impact of recombination on nucleotide substitutions in the human genome. *PLoS Genet* 2008;**4**:e1000071.
- Eisenberg E, Levanon EY. Human housekeeping genes, revisited. *Trends Genet* 2013;**29**(10):569–74.
- Chin C-S, Wagner J, Zeng Q, et al. A diploid assembly-based benchmark for variants in the major histocompatibility complex. *Nat Commun* 2020;**11**(1):4794.
- Wu Z, Sheng H, Chen Y, et al. Copy number variation of the lipoprotein(a) (LPA) gene is associated with coronary artery disease in a southern Han Chinese population. *Int J Clin Exp Med* 2014;**7**:3669.
- Tautz D, Domazet-Lošo T. The evolutionary origin of orphan genes. *Nat Rev Genet* 2011;**12**:692–702.
- Yang R, Van Etten JL, Dehm SM. Indel detection from DNA and RNA sequencing data with transIndel. *BMC Genomics* 2018;**19**:270.
- Sahlin K, Medvedev P. De novo clustering of long-read transcriptome data using a greedy, quality value-based algorithm. *J Comput Biol* 2020;**27**(4):472–84.

31. Jiang T, Liu B, Li J, et al. rMETL: sensitive mobile element insertion detection with long read realignment. *Bioinformatics* 2019;**35**(18):3484–6.
32. Yang C, Chu J, Warren RL, et al. NanoSim: nanopore sequence read simulator based on statistical characterization. *Gigascience* 2017;**6**, doi:10.1093/gigascience/gix010.
33. Index of [/giab/ftp/data/AshkenazimTrio/HG002_NA24385.son](ftp://giab/ftp/data/AshkenazimTrio/HG002_NA24385.son). https://ftp.ncbi.nih.gov/giab/ftp/data/AshkenazimTrio/HG002_NA24385.son/. Accessed 2021 August 23.
34. Zook JM, Catoe D, McDaniel J, et al. Extensive sequencing of seven human genomes to characterize benchmark reference materials. *Sci Data* 2016;**3**:160025.
35. 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.
36. lh: lh3/seqtk. <https://github.com/lh3/seqtk>. Accessed 2021 May 28.
37. Robinson J, Thorvaldsdóttir H, Winckler W, et al. Integrative genomics viewer. *Nat Biotechnol* 2011;**29**:24–6. <https://doi.org/10.1038/nbt.1754>.
38. Li H, Ruan J, Durbin R. Mapping short DNA sequencing reads and calling variants using mapping quality scores. *Genome Res* 2008;**18**(11):1851–8.
39. Jain C, Rhie A, Zhang H, et al. Weighted minimizer sampling improves long read mapping. *Bioinformatics* 2020;**36** (Suppl 1):i1111–8.
40. Prodanov T, Bansal V. Sensitive alignment using paralogous sequence variants improves long-read mapping and variant calling in segmental duplications. *Nucleic Acids Res* 2020;**48**:e114.
41. Langmead B, Salzberg SL. Fast gapped-read alignment with Bowtie 2. *Nat Methods* 2012;**9**(4):357–9.
42. Li H, Handsaker B, Wysoker A, et al. The Sequence Alignment/Map format and SAMtools. *Bioinformatics* 2009;**25**(16):2078–9.
43. Barnett DW, Garrison EK, Quinlan AR, et al. BamTools: a C API and toolkit for analyzing and managing BAM files. *Bioinformatics* 2011;**27**(12):1691–2.
44. Danecek P, McCarthy SA. BCFtools/csq: haplotype-aware variant consequences. *Bioinformatics* 2017;**33**(13):2037–9.
45. NCBI. Taxonomy browser (*Saccharomyces cerevisiae* S288C). <https://www.ncbi.nlm.nih.gov/Taxonomy/Browser/wwwtax.cgi?id=559292>. Accessed 2021 August 23.
46. Fu Y, Mahmoud M, Muraliraman VV, et al. Supporting data for “Vulcan: Improved long-read mapping and structural variant calling via dual-mode alignment.” *GigaScience Database* 2021. <https://dx.doi.org/10.5524/100926>.