genes

MDPI

Article

DelInsCaller: An Efficient Algorithm for Identifying Delins and Estimating Haplotypes from Long Reads with High Level of Sequencing Errors

Shenjie Wang [1,2], Xuanping Zhang [1,2], Geng Qiang [1,2] and Jiayin Wang [1,2,*]

[1] School of Computer Science and Technology, Xi'an Jiaotong University, Xi'an 710049, China
[2] Shaanxi Engineering Research Center of Medical and Health Big Data, Xi'an Jiaotong University, Xi'an 710049, China
[*] Correspondence: wangjiayin@mail.xjtu.edu.cn

Abstract: Delins, as known as complex indel, is a combined genomic structural variation formed by deleting and inserting DNA fragments at a common genomic location. Recent studies emphasized the importance of delins in cancer diagnosis and treatment. Although the long reads from PacBio CLR sequencing significantly facilitate delins calling, the existing approaches still encounter computational challenges from the high level of sequencing errors, and often introduce errors in genotyping and phasing delins. In this paper, we propose an efficient algorithmic pipeline, named delInsCaller, to identify delins on haplotype resolution from the PacBio CLR sequencing data. delInsCaller design a fault-tolerant method by calculating a variation density score, which helps to locate the candidate mutational regions under a high-level of sequencing errors. It adopts a base association-based contig splicing method, which facilitates contig splicing in the presence of false-positive interference. We conducted a series of experiments on simulated datasets, and the results showed that delInsCaller outperformed several state-of-the-art approaches, e.g., SVseq3, across a wide range of parameter settings, such as read depth, sequencing error rates, etc. delInsCaller often obtained higher f-measures than other approaches; specifically, it was able to maintain advantages at ~15% sequencing errors. delInsCaller was able to significantly improve the N50 values with almost no loss of haplotype accuracy compared with the existing approach as well.

Keywords: sequencing data analysis; variant calling; variant phasing; delins; complex indel

check for updates

Citation: Wang, S.; Zhang, X.; Qiang, G.; Wang, J. DelInsCaller: An Efficient Algorithm for Identifying Delins and Estimating Haplotypes from Long Reads with High Level of Sequencing Errors. *Genes* **2023**, *14*, 4. https://doi.org/10.3390/genes14010004

Academic Editors: Yan Guo and Jinchuan Xing

Received: 14 September 2022
Revised: 27 October 2022
Accepted: 28 October 2022
Published: 20 December 2022

1. Introduction

Genomic structural variations (SVs) generally include deletions, tandem duplications, insertions, inversions, translocations, and their combinations [1,2]. Delins, as known as complex indel, is one of those combinations formed by simultaneously deleting and inserting DNA fragments of different sizes at a common genomic location [3,4]. Delins are widely reported in different researches, some of which are considered functional or susceptible [5,6]. For example, some delins may be inherited which are identified and validated from familial research [7,8]. While the germline delins are observed in populations, some somatic delins are reported potentially druggable [9,10]. Thus, identifying delins from sequencing data is a necessary task for data analysis.

Several approaches, such as Pindel-C [3], SV-Bay [11], INDELseek [12], have been developed for identifying delins from the second generation sequencing data. However, the relatively short read length limits the alignment and assembly, thus affecting the performance on variant calling, genotyping and phasing. With the rapid development of sequencing technologies, long read enables high-confidence mapping across a greater percentage of the genome [13,14]. The PacBio SMRT sequencing technology, as a representative of the third generation sequencing technology, has been attracting more and more attention since its commercial release in 2010 [15]. With long reads, structural variations that are previously undetectable in the second generation data can be accurately detected [16]. Currently, the detection algorithm of delins for long reads is SVseq3 [17]. SVseq3 identifies a series of suspicious regions from the aligned reads mapped by BLASR [18]. Although the SVseq3 can process the third generation sequencing data with better f-measure, it can only tolerate up to 3% of sequencing errors. However, PacBio SMRT sequencing technology produces two types of reads: (i) continuous long reads (CLR) (long reads with high error rates) and (ii) circular consensus sequencing (CCS) reads (short reads with low error rates) [19]. For CLR reads, the sequencing error rate is up to 15%, which would result in a quite mix of true and false positive variations. When SVseq3 is processing such data, it always fails due to its inability to distinguish between true and false positive variations.

In addition to variant calling, estimating haplotypes is another important task [20]. For example, recent studies suggest that the patients with the same level of mutation loads may lead to different clinical manifestation when the loads on haplotypes are significantly different. Benefiting from the CLR reads with the length ranging from 15,000 to 40,000 bps, we are empowered to estimate each haplotype by identifying these variations that are co-located on the same read/contig. Currently, the tools for haplotype estimation based on variant detection results include hapcut2, whatshap, etc. However, these tools are prone to incorrect splicing due to the false positives introduced by high sequencing errors.

To summarize above, the existing methods often encounter the following two computational problems. delins is difficult to detect accurately due to the interference of sequencing errors. In addition, the false positives introduced by sequencing errors often mislead contig splicing. But we do need the read length advantage of CLR sequencing in some scenarios. Motivated by this, we propose an efficient algorithmic pipeline, named delInsCaller, to identify the delins on haplotype resolution from the PacBio CLR sequencing data with high sequencing errors. delInsCaller design a fault-tolerant method by calculating a variation density score, which helps to locate the candidate mutational regions under a high-level of sequencing errors. It adopts a base association-based contig splicing method, which facilitates contig splicing in the presence of false-positive interference. The experiments showed that delinsCaller was effective in identifying delins on haplotype resolution, and outperformed several state-of-the-art approaches.

2. Materials and Methods

The proposed analysis pipeline consists of the following components. First, it determines the approximate region of a variant by calculating a series of variation density fractions. Then, the candidate variant is classified by multiple machine learning models. A local re-alignment component then locates the exact breakpoints according to the soft clip alignments and estimates the possible source of the inserted fragments. Finally, it estimates the haplotypes for the identified variants.

2.1. Identifying Regions of Variations

We extract variant signatures from the SAM file to identify the genomic structural variation region. Due to the sequencing errors, these variant signatures imply not only the true variants, but the false positives as well. Previous studies report that the sequencing errors of CLR reads are almost uniformly distributed, dominated by the fake insertions and deletions [21]. Therefore, the unmatched base density on the reads mapped to a normal region usually locates in a low level. In contrast, the reads mapped to the region with variations may have a relatively higher density. According to this experience, we propose to calculate the variation fraction, which measures the unmatched base density in different regions.

The core idea is as follows: (1) we define a value called variation aggregation degree, which measures the mutation load on a specific region. According to the error model, for any site, we are not sure whether it is a variant or not, but the surrounding sites provide more information. When there are many unmatched sites around, it indicates that this region has a higher variation aggregation degree, and then it has higher probability of being a variation region. (2) In addition, we borrow the idea that the loci in close proximity may have a stronger linkage. The linkage strength sometimes decays exponentially as the physical distance becomes larger [22]. Thus, the mapping status of the surrounding sites are assigned different weights according to the distance from the central site. The closer the site locates to the central, the higher weight is assigned. Now, we calculate a variation score for each site.

Specifically, when traversing sites, we set a sliding window with the size of windows_len and calculate the proportion of unmatched bases at any site k in the window, that is, the proportion of the number of reads harboring unmatched bases to the number of reads covered this site. The formula is as follow:

$$p_c(k) = \frac{u}{d} \tag{1}$$

where c represents the central site of this window, u represents the number of reads containing unmatched bases, and d represents the total number of reads covering this site. For any site k in the window, we define the weight coefficient to measure its influence on the central site. Two conditions are satisfied: (1) The value of k is inversely proportional to the distance between k and c; (2) When the unmatched rate of all sites in the window (c − windows_len, c + windows_len) is 1, the variation score is defined as 1. The formulas are as follows:

$$w_c(k) \propto \frac{1}{l_{k-c}} \tag{2}$$

$$\sum_{k=c-\text{windows\_len}}^{c+\text{windows\_len}} w_c(k) = 1 \tag{3}$$

where l_{k-c} represents the distance between k and c. Since standard normal distribution meets the formula $\int_{-3}^{3} f(x)dx$ = 0.9974 (approximate to 1), we simply adopt the standard normal distribution as the weight coefficient function here. The calculation of the weight coefficient $w_c(k)$ is:

$$\begin{cases} f(x) = \frac{1}{\sqrt{2\pi}} e^{\left(-\frac{x^2}{2}\right)} \\ w_c(k) = \frac{3}{\text{windows\_len}} f\left(\frac{3|k-c|}{\text{windows\_len}}\right) \end{cases} \tag{4}$$

Now, we obtain the calculation formula of the variation score as:

$$Score_c = \sum_{k=c-w\text{indows\_len}}^{c+\text{windows\_len}} w_c(k) p_c(k) \tag{5}$$

This variation score has the following features: (1) For any site, the value of variation score is not only related to itself, but related to the surrounding sites as well; (2) If all the sites in the window are mutations, the variation score is approximately equal to 1; while if all the sites are matched to the reference, the variation score is equal to 0; (3) The sites around site c have different effects on the variation score, which depends on the distance to site c. Figure 1 shows an example of the variation scores on a region, when the sequencing error rate reaches 15%.

According to Figure 1, the portion where the variation score is significantly higher than the surrounding area is the approximate region where we have tentatively identified possible variations. To refine the range of the variation interval, we traverse site i in turn and smooth the curve to eliminate some noise:

$$smoothScore_i = \frac{1}{2 * \text{windows\_len} + 1} \sum_{j=i-win}^{i+win} Score_i \tag{6}$$

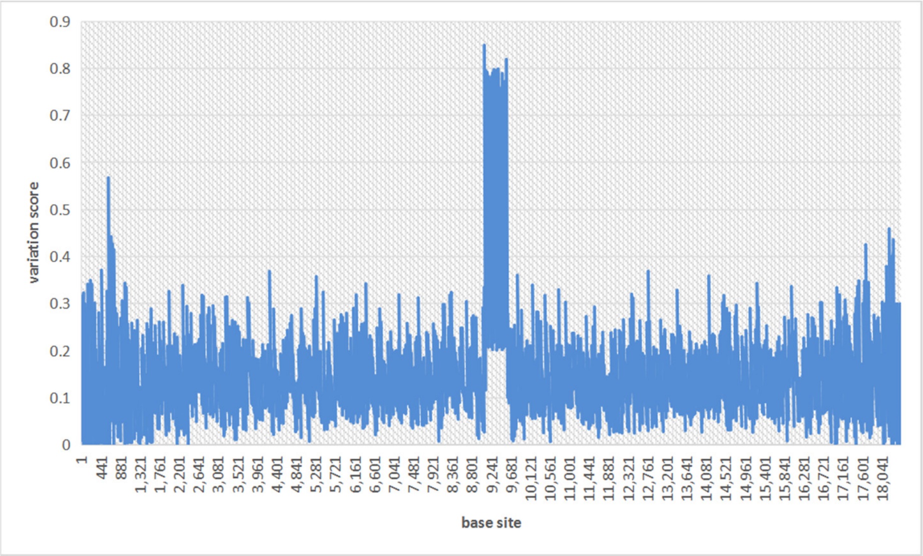


Figure 1. The variation scores for each site on a region under a 15% sequencing error rate.

Figure 2 shows the smoothed variation score graph. Obviously, we are able to obtain a clear candidate interval of a structural variation.

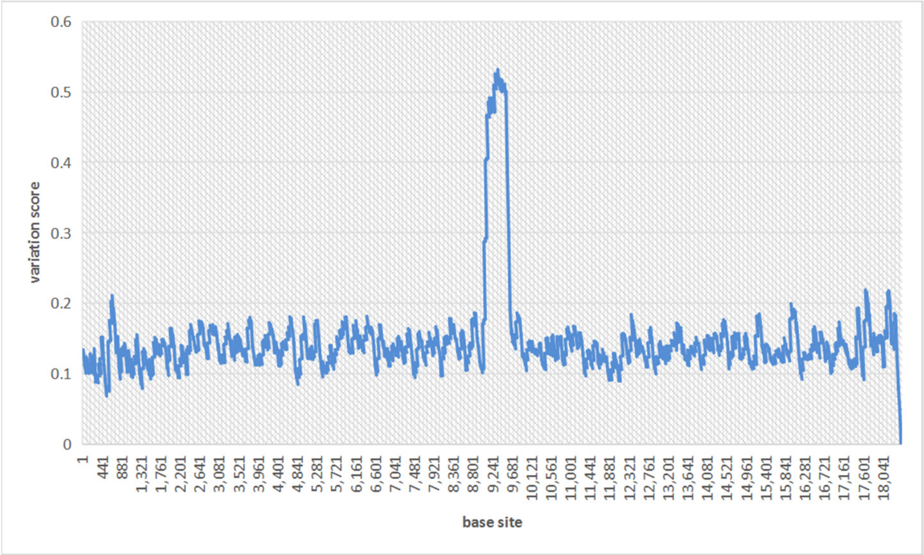


Figure 2. The smoothed variation scores on the same region under a 15% sequencing error rate.

We further set a threshold T for variation scores to determine whether a region, after smoothing, is a candidate region or not. For interval $(l\,,r)$, when it satisfies:

$$\begin{cases} smoothScore_i > T \\ smoothScore_{l-1} < T \\ smoothScore_{r+1} < T \end{cases} \qquad i \in (l\,,r) \tag{7}$$

We select $(l\,,r)$ as the candidate region for a structural variation. According to this pipeline, the approximate ranges of variations can be located quickly, in the case of high sequencing errors.

2.2. Classification of Variations

When we have the candidate regions, we have to further identify the type of structural variations. The variation could be delins, insertion, deletion, or some other type of structural variation. Due to the sequencing errors, we cannot use the hard-filters to identify the types of variations, because the differences among these variations in the space of features are confused. Thus, we trained multiple SVMs to implement a multiple classifications. An SVM classifier is trained between each of the two variation types. For m types of variations, a total of $\frac{m(m-1)}{2}$ SVM classifiers were trained. We take the variation type with the largest count according to the results of all classifiers.

To obtain a better classification performance, we study the features of various regions harboring different variations. Since the insertions and deletions often have higher insertion rates and deletion rates, respectively, while delins have higher transition ratios, these features are selected as a group. Since the unmatched sites in insertions and deletions are generally continuous, while the continuity of the unmatched sites in delins is poor, we select the maximum ratio of consecutive unmatched bases and the corresponding ratios as features.

2.3. Locating the Start and End Sites of the Variation

In this step, we are trying to locate the specific start and end sites of a variation. First, we locate the start site of a variation: (1) Cluster all reads with soft clips at the right end and the length of the soft clip greater than 50 bp in the variation interval defined in the first step. (2) Calculate the starting site of each read in the cluster according to the cigar value of each read to obtain a candidate set of the starting site. (3) Outliers in the candidate set are eliminated by the box diagram method. (4) The mean value of the starting site candidate set is calculated as the starting site for the variation. Next, we locate the end site of the variation: (1) Cluster all reads with soft clips at the left end and the length of the soft clip greater than 50 bp in the variation interval defined in the first step. (2) Calculate the ending site of each read in the cluster according to the cigar value of each read to obtain a candidate set of the ending site. (3) Outliers in the candidate set are eliminated by the box diagram method. (4) The mean value of the ending site candidate set is calculated as the ending site for the variation. A simple flow chart explains this algorithm is shown in Figure 3.

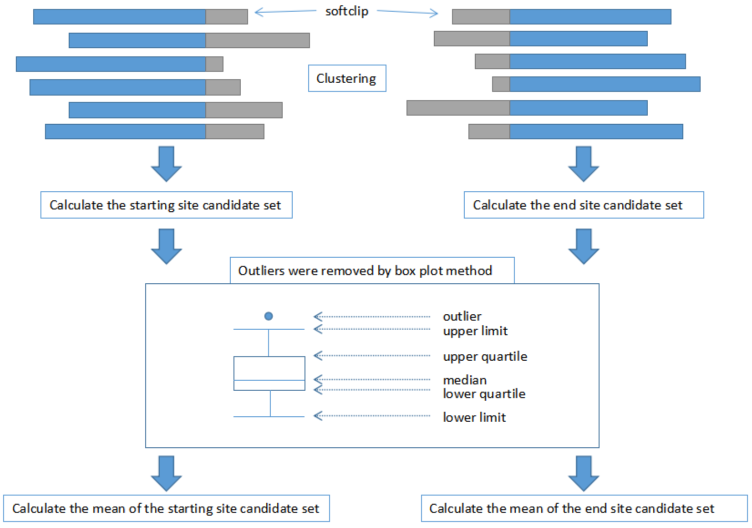

softclip
Clustering
Calculate the starting site candidate set
Calculate the end site candidate set
Outliers were removed by box plot method
outlier
upper limit
upper quartile
median
lower quartile
lower limit
Calculate the mean of the starting site candidate set
Calculate the mean of the end site candidate set

Figure 3. The flow chart of the key steps for locating the start and end sites of a variation.

2.4. Finding the Source of the Inserted Fragment

After obtaining the start and end location of the mutation and the type of the mutation, the algorithm traces the source of inserted fragments in the delins variant and insertion variant. The approximate steps, shown in Figure 4, include: (1) The reads containing soft clips in the interval of the variation are cut to obtain the base sequence of the soft clip part, and the base sequence is written into a new FQ file as artificial reads. (2) The new FQ file is matched back to the reference genome, and the BAM file is generated. (3) The longest continuous matching fragment is selected from the sequence mapping results to determine its mapping site. If soft clips are still included, back to step (1). Otherwise, the algorithm is terminated.

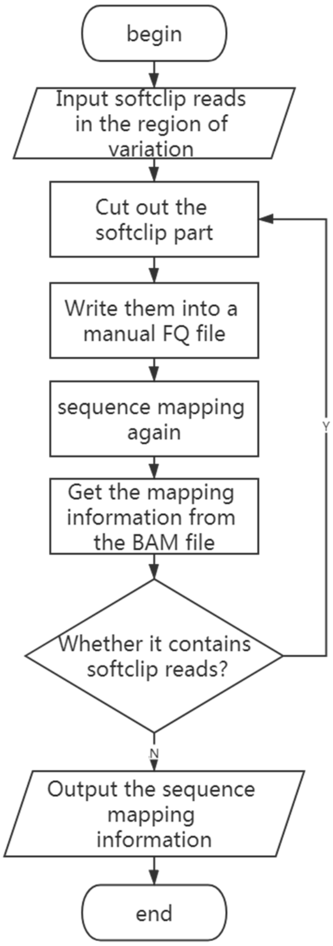

begin
Input softclip reads in the region of variation
Cut out the softclip part
Write them into a manual FQ file
sequence mapping again
Get the mapping information from the BAM file
Whether it contains softclip reads?
Y
N
Output the sequence mapping information
end

Figure 4. The flow chart for finding the sources of the inserted fragment.

2.5. Estimating Haplotypes

We have obtained the identification results of insertions, deletions, and delins through the previous steps. To further estimate haplotype, we introduce information about single point variation in the sequenced samples (the information about single point variation can be obtained with software such as deepvariant, clairvoyante, NanoCaller [23], and so on) and split the variation into two sets by splicing reads containing the same variations (the HapCUT2 can extract variation information from VCF file and splice them into contigs). For intervals that could not be spliced, we estimate their correlations using linkage disequilibrium value. The formula is as follows:

$$L = \sum_{i=1}^{m} \sum_{j=1}^{n} lq * p \tag{8}$$

where L represents the correlation of two contigs, lq represents the linkage strength of two variants, p represents the frequency of variants, and m and n represent the number of variants in the two contigs, respectively. For two contigs that need to be spliced, first, we calculate the correlation strength of the first polymorphic site base in the posterior contig based on the bases of each polymorphic site in the anterior contig. Then, the same operation is repeated for each posterior base until the correlation calculation is completed for each polymorphic site of the posterior base, and finally, the correlation strengths of the two contigs are accumulated. We select the posterior contig with the higher correlation strength to splice with the anterior contig. The splicing schematic is shown in Figure 5.

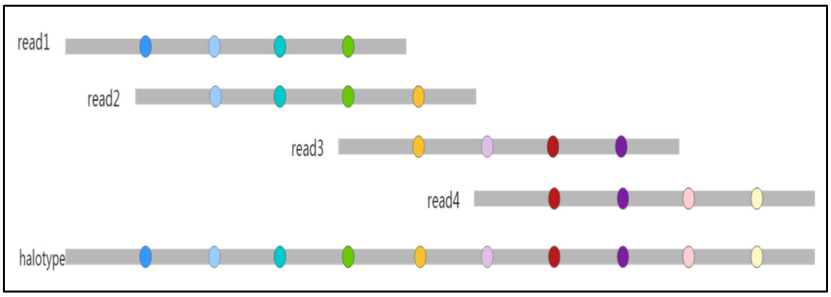


Figure 5. The schematic of estimating a haplotype. The different colored circles represent different variants.

The steps of the algorithm are as follows: (1) Introduce the information of variants and incorporate them with insertions, deletions, and delins identification results. (2) Find the source of each variation by walking through the cigar value of each read recorded in the same file. (3) Sequentially splice the reads containing the same variations to obtain a splicing sequence as long as possible. (4) Divide the variations into different sets according to the sequence obtained by splicing. (5) Perform secondary splicing based on the correlation strength between contigs. The program flow chart is shown in Figure 6.

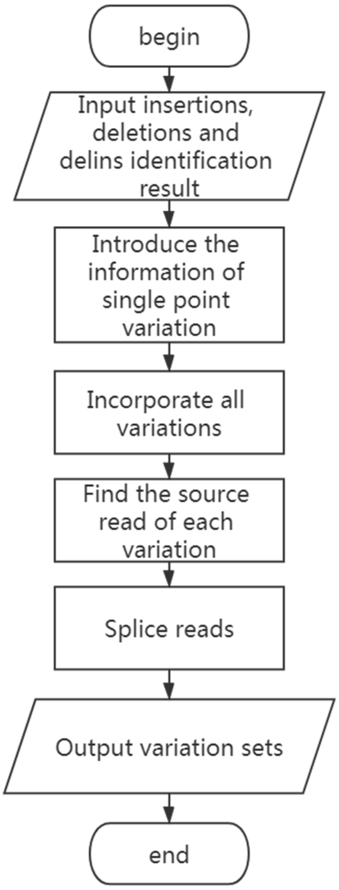

begin
Input insertions, deletions and delins identification result
Introduce the information of single point variation
Incorporate all variations
Find the source read of each variation
Splice reads
Output variation sets
end

Figure 6. The flow chart of haplotypes estimating process.

2.6. Experimental Methods

The performance of the proposed algorithm was validated with artificial data sets and an independent validation data sets. Since the SVseq3 algorithm can be used for PacBio sequencing data and can detect delins and the position information of the inserted segment, we compared the delInsCaller and SVseq3 under different sequencing errors. In the experiments, precision, recall, and f-measure were used to measure the performance of the algorithm. What's more, we compared it with the state-of-the-art haplotype estimation tool, HapCUT2. To measure the experimental performances, we selected two widely used haplotype estimating metrics, haplotype accuracy (including switch and mismatch rates), and N50, where switch rate means switch errors as a fraction of possible positions for switch errors, mismatch rate means mismatch errors as a fraction of possible positions for mismatch errors, and N50 means the N50 metric of haplotype completeness.

2.6.1. Simulation Data Sets Experiments

To generate the simulation data sets, a 10 Mbp region was randomly sampled from chromosome 1 of the human reference genome hg19. Then we randomly selected the variation sites on the reference, delete and insert approximately equal-length fragments to simulate the delins variations, and obtain the variant DNA sequence. In the process of simulating the delins, we randomly chose the length of the deletion and the position of the insertion on the reference. To simulate the reads, we used the commonly used simulation tool for the third-generation sequencing data, the PBsim simulator, which can simulate the sequencing process on a DNA sequence file (.fasta) to obtain the read file (.fastq). PBsim simulator can specify the sequencing depth, the mean, and the variance of the read length, the sequencing error rate, and other data characteristics.

2.6.2. Independent Validation Data Sets Experiments

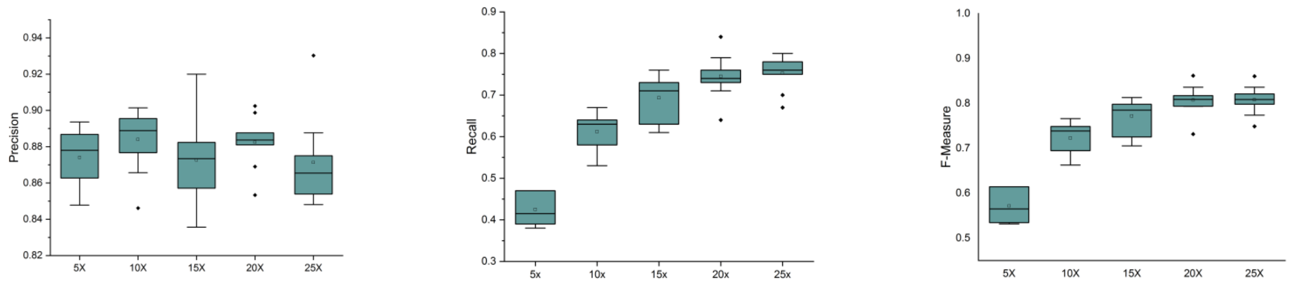
To further verify the results and prove the advantages of the method, we conducted independent validation data sets experiments. We used the sample-fastq function of PBsim to sample data from the real subreads data sets of the NA12878 individual (HG001). The real data were downloaded from GIAB (https://ftp-trace.ncbi.nlm.nih.gov/giab/ftp/data/, accessed on 28 January 2022). Here, we designed two groups of experiments. First, we generated independent validation data sets and performed experiments at different sequencing depths. We set the read length as N (15,000, 1500), the sequencing error rate as 15%, and varied the sequencing depth from 5× to 25×. Again, haplotype estimating metrics, haplotype accuracy (including switch and mismatch rates), and N50 were selected and compared to the state-of-the-art haplotype estimation tool, HapCUT2. Then we compared the precision, recall, and f-measure of delInsCaller on the previous validation data sets and the independent validation data sets, considering that there are too many combinations of read lengths, sequencing depth, delins length, etc. Here, we set the length of delins as 500 bp, the length of read as N (20,000, 2300), the sequencing error rate as 15%, the sequencing depth as 25×, and the number of delins as 100, as an example to generate independent validation data sets.

3. Results and Discussion

3.1. Experiments under Different Data Characteristics

3.1.1. Experimental Results under Different Sequencing Depths

We varied the sequencing depth from 5× to 25× and set the read length as N (20,000, 2300), the sequencing error rate as 15%, the length of delins as 500 bp, and the number of delins as 100. For each value of the sequencing depth, we performed ten repeated experiments and drew box diagrams, as shown in Figure 7. From Figure 7, we can conclude that with the increase in the sequencing depth, the recall and f-measure also rise. The reason for this may be that at lower sequencing depth, the reads may not contain enough delins variation signals. However, as the depth deepens, more and more variation signals can be collected.

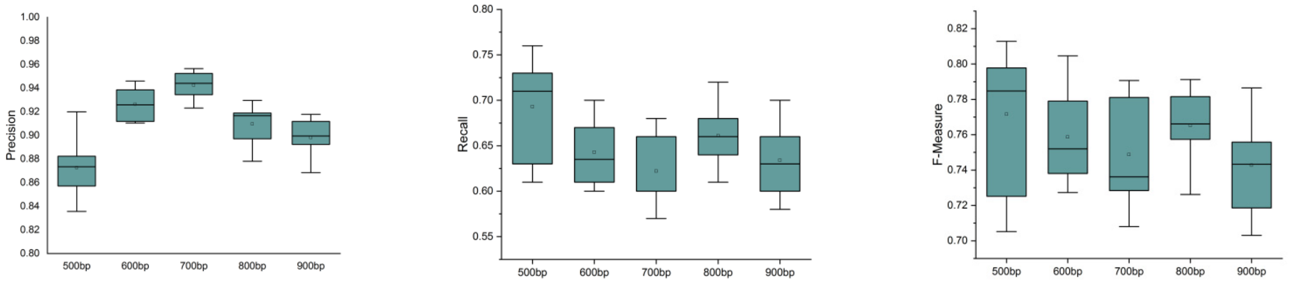


(**a**) (**b**) (**c**)

Figure 7. Precision, recall, and f-measure under different sequencing depths: (**a**) description of precision value; (**b**) description of recall value; (**c**) description of the f-measure value. The horizontal lines above and below the boxes represent the maximum and minimum values of the data, respectively. The discrete black dots represent outliers.

3.1.2. Experimental Results under Different Lengths of Delins

We altered the length of delins from 500 bp to1000 bp and set the read length as N (20,000, 2300), the sequencing error rate as 15%, the sequencing depth as 15×, and the number of delins as 100. For each value of the delins length, we performed ten repeated experiments and drew box diagrams, as shown in Figure 8. From Figure 8, we can see that the precision, recall, and f-measure of the delins detection remain almost unchanged as the length of the delins increases. It indicates that the length of delins has little influence on the precision, recall, and f-measure.

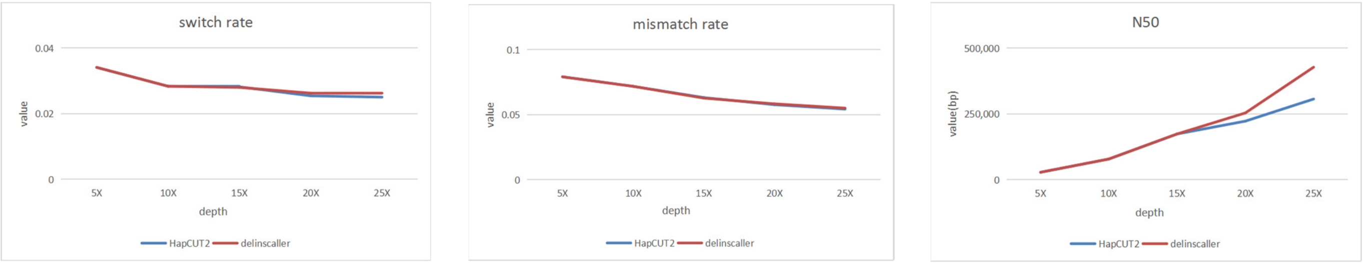


(**a**) (**b**) (**c**)

Figure 8. Precision, recall, and f-measure under different lengths of delins: (**a**) description of precision value; (**b**) description of recall value; (**c**) description of the f-measure value. The horizontal lines above and below the boxes represent the maximum and minimum values of the data, respectively. The discrete black dots represent outliers.

3.2. Haplotype Phasing Experiments

In order to test the performance of haplotype estimation, we performed experiments under different sequencing depths and set the read length as N (15,000, 1500) and the sequencing error rate as 15%. We varied the sequencing depth from 5× to 25×. Then a series of experiments were performed for the sequencing data at each depth, and the experimental results are shown in Figure 9. From Figure 9, we can see that at different sequencing depths, our proposed algorithm can be able to improve the N50 values with almost no loss of haplotype accuracy. Moreover, as the depth increases, the advantages become more and more obvious.

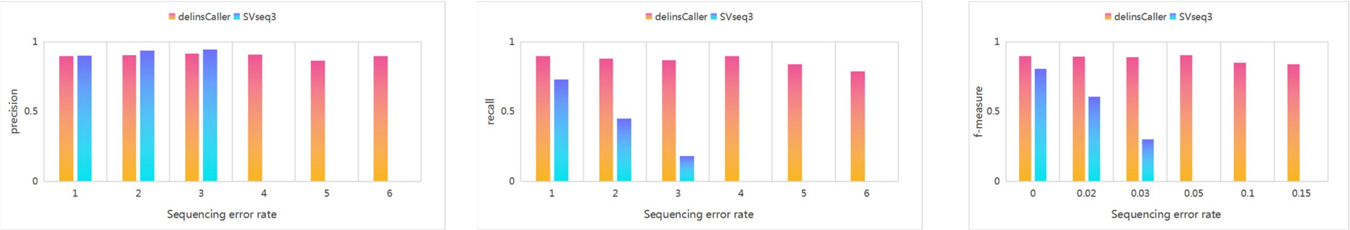


(**a**) (**b**) (**c**)

Figure 9. Haplotype estimating under different sequencing depths; (**a**) switch rate; (**b**) mismatch rate; (**c**) N50. The blue line represents the HapCUT2, and the red line represents the dellinsCaller.

3.3. Comparison Experiment with SVseq3

We set the sequencing errors in the experiment as 15%, 10%, 5%, 3%, 2%, and 0%, the read length as N (15,000, 1500), the length of the delins as 500~1000 bp, and the number of delins as 100. The experimental results are shown in Figure 10. As can be seen from the figure, the detection performance of delInsCaller for delins without sequencing errors is similar to that of SVseq3, but delInsCaller is significantly better than SVseq3 if there exist sequencing errors, especially in the case of high sequencing errors. As the sequencing error rate increases, the detection performance of SVseq3 decreases rapidly. However, the increase in sequencing error rate has less effect on delInsCaller, whose f-measure is stable, and the f-measure value of delInsCaller can be reached above 80% in some tests, even when the sequencing error rate reaches 15%. Therefore, the algorithm has a higher tolerance for sequencing errors.



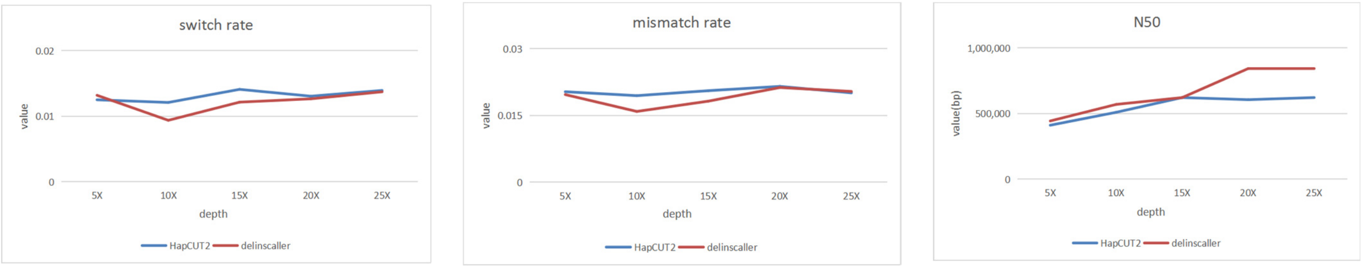
(**a**) (**b**) (**c**)

Figure 10. Comparison of delInsCaller and SVseq3 under different sequencing errors: (**a**) precision value; (**b**) recall value; (**c**) f-measure value.

3.4. Independent Validation Data Sets Experiments

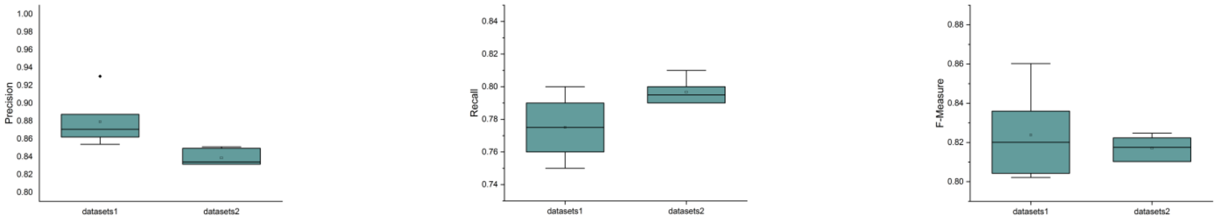
First, we set the read length as N (15,000, 1500), the sequencing error rate as 15%, and varied the sequencing depth from 5× to 25×. The experimental results are shown in Figure 11. From Figure 11, we can also see that at different sequencing depths, our proposed algorithm can be able to improve the N50 values with almost no loss of haplotype accuracy. Moreover, as the depth increases, the advantages become more and more obvious. The result is consistent with the previous experimental results on the validation data sets.

We set the length of delins as 500 bp, the length of read as N (20,000, 2300), the sequencing error rate as 15%, the sequencing depth as 25×, and the number of delins as 100 to generate independent validation data sets. Then, we compared the precision, recall, and f-measure of delInsCaller with the previous validation data sets and the independent validation data sets. The experimental results are shown in Figure 12. From Figure 12, we can see that there is little difference in the performance of delInsCaller on the previous validation data sets and the independent validation data sets. In particular, the validation results on both data sets are largely consistent under the f-measure metric.



(**a**) (**b**) (**c**)

Figure 11. Haplotype estimating under different sequencing depth: (**a**) switch rate; (**b**) mismatch rate; (**c**) N50. The blue line represents the HapCUT2, and the red line represents the dellinsCaller.



(**a**) (**b**) (**c**)

Figure 12. Comparison of detection performance on the previous validation data sets and the independent validation data sets: (**a**) precision value, (**b**) recall value, (**c**) f-measure value. Data set 1 refers to the previous validation data sets, and data set 2 refers to the independent validation data sets. The horizontal lines above and below the boxes represent the maximum and minimum values of the data, respectively. The discrete black dots represent outliers.

4. Conclusions

In this paper, we focus on identifying delins on haplotype resolution from the PacBio CLR sequencing data. Recent studies emphasized the importance of delins in cancer diagnosis and treatment. Although the CLR reads significantly facilitate delins calling, the existing approaches still encounter computational challenges from the high level of sequencing errors, and often introduce errors in genotyping and phasing delins. So, we proposed an efficient algorithmic pipeline, named dellnsCaller, to identify the delins on haplotype resolution from the PacBio CLR sequencing data. dellnsCaller has a good tolerance for sequencing errors and can still maintain a high f-measure under a 15% sequencing error rate. It uses a fault-tolerant method by calculating a variation density score, which helps to locate the candidate mutational regions under a high-level of sequencing errors. And it adopts a base association-based contig splicing method, which facilitates contig splicing in the presence of false-positive interference.. We carried out a set of experiments to prove that dellnsCaller has a good performance by changing the delins length, sequencing depth, and other data features on the simulation data set. Moreover, we also conducted comparative experiments with several state-of-the-art approaches, e.g., SVseq3, HapCUT2. It is proved that dellnsCaller outperformed the existing algorithms. Specifically, it maintained the advantages at ~15% sequencing errors. Therefore, the proposed algorithm is very effective in identifying the delins on haplotype resolution from the PacBio CLR sequencing data with high sequencing errors.

Author Contributions: Conceptualization, S.W.; methodology, S.W. and G.Q.; writing—original draft preparation, S.W.; writing—review and editing, J.W. and X.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Natural Science Basic Research Program of Shaanxi, grant number 2020JC-01. The APC was also funded by the Natural Science Basic Research Program of Shaanxi, grant number 2020JC-01.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: This is research on developing a bioinformatics tool for sequencing data analysis. We do not generate sequencing data.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Quinlan, A.R.; Hall, I.M. Characterizing complex structural variation in germline and somatic genomes. *Trends Genet.* **2012**, *28*, 43–53. [CrossRef] [PubMed]
2. Collins, R.L.; Brand, H.; Redin, C.E.; Hanscom, C.; Antolik, C.; Stone, M.R.; Glessner, J.T.; Mason, T.; Pregno, G.; Dorrani, N.; et al. Defining the diverse spectrum of inversions, complex structural variation, and chromothripsis in the morbid human genome. *Genome Biol.* **2017**, *18*, 36. [CrossRef] [PubMed]
3. Ye, K.; Wang, J.; Jayasinghe, R.; Lameijer, E.-W.; McMichael, J.F.; Ning, J.; McLellan, M.D.; Xie, M.; Cao, S.; Yellapantula, V.; et al. Systematic discovery of complex insertions and deletions in human cancers. *Nat. Med.* **2016**, *22*, 97–104. [CrossRef] [PubMed]
4. Carvalho, C.M.; Lupski, J.R. Mechanisms underlying structural variant formation in genomic disorders. *Nat. Rev. Genet.* **2016**, *17*, 224–238. [CrossRef]
5. Roerink, S.F.; van Schendel, R.; Tijsterman, M. Polymerase theta-mediated end joining of replication-associated DNA breaks in *C. elegans*. *Genome Res.* **2014**, *24*, 954–962. [CrossRef]
6. Koole, W.; Van, S.R.; Karambelas, A.E.; van Heteren, J.T.; Okihara, K.L.; Tijsterman, M. A polymerase theta-dependent repair pathway suppresses extensive genomic instability at endogenous g4 DNA sites. *Nat. Commun.* **2014**, *5*, 3216. [CrossRef]
7. Kwong, A.; Shin, V.Y.; Au, C.H.; Law, F.B.; Ho, D.N.; Ip, B.K.; Wong, A.T.; Lau, S.S.; To, R.M.; Choy, G.; et al. Detection of Germline Mutation in Hereditary Breast and/or Ovarian Cancers by Next-Generation Sequencing on a Four-Gene Panel. *J. Mol. Diagn.* **2016**, *18*, 580–594. [CrossRef]
8. Garcia, C.; Lyon, L.; Littell, R.D.; Powell, C.B. Comparison of risk management strategies between women testing positive for a BRCA variant of unknown significance and women with known BRCA deleterious mutations. *Genet. Med.* **2014**, *16*, 896–902. [CrossRef]
9. Kloosterman, W.P.; Francioli, L.C.; Hormozdiari, F.; Marschall, T.; Hehirkwa, J.Y.; Abdellaoui, A.; Lameijer, E.-W.; Moed, M.H.; Koval, V.; Renkens, I.; et al. Characteristics of de novo structural changes in the human genome. *Genome Res.* **2015**, *25*, 792. [CrossRef]
10. Zheng, T.; Li, Y.; Geng, Y.; Zhao, Z.; Zhang, X.; Xiao, X.; Wang, J. CIGenotyper: A Machine Learning Approach for Genotyping Complex Indel Calls. *Bioinform. Biomed. Eng.* **2018**, *10813*, 473–485.
11. Iakovishina, D.; Janoueix-Lerosey, I.; Barillot, E.; Regnier, M.; Boeva, V. SV-Bay: Structural variant detection in cancer genomes using a Bayesian approach with correction for GC-content and read mappability. *Bioinformatics* **2016**, *32*, 984–992. [CrossRef] [PubMed]
12. Au, C.H.; Leung, A.Y.H.; Kwong, A.; Chan, T.L.; Ma, E.S.K. INDELseek: Detection of complex insertions and deletions from next-generation sequencing data. *BMC Genom.* **2017**, *18*, 16. [CrossRef] [PubMed]
13. Chaisson, M.J.P.; Huddleston, J.; Dennis, M.Y.; Sudmant, P.H.; Malig, M.; Hormozdiari, F.; Antonacci, F.; Surti, U.; Sandstrom, R.; Boitano, M.; et al. Resolving the complexity of the human genome using single-molecule sequencing. *Nature* **2014**, *517*, 608–611. [CrossRef] [PubMed]
14. Lee, H.; Schatz, M.C. Genomic dark matter: The reliability of short read mapping illustrated by the genome mappability score. *Bioinformatics* **2012**, *28*, 2097–2105. [CrossRef] [PubMed]
15. Eid, J.; Fehr, A.; Gray, J.; Luong, K.; Lyle, J.; Otto, G.; Peluso, P.; Rank, D.; Baybayan, P.; Bettman, B.; et al. Real-time DNA sequencing from single polymerase molecules. *Science* **2010**, *472*, 431–455. [CrossRef]
16. John, H.; Chaisson, M.J.P.; Steinberg, K.M. Corrigendum Discovery and genotyping of structural variation from long-read haploid genome sequence data. *Genome Res.* **2017**, *27*, 677–685.
17. Zhang, X.; Chen, H.; Zhang, R.; Pei, J.; Wang, Y.; Zhao, Z.; Huang, Y.; Wang, J. Detecting complex indels with wide length-spectrum from the third generation sequencing data. In Proceedings of the IEEE International Conference on Bioinformatics and Biomedicine, Kansas City, MO, USA, 13–16 November 2017; IEEE: Piscataway Township, NJ, USA, 2017; pp. 1980–1987.
18. Chaisson, M.; Tesler, G. Mapping single molecule sequencing reads using basic local alignment with successive refinement (BLASR): Application and theory. *BMC Bioinform.* **2012**, *13*, 238. [CrossRef]

19. Yukiteru, O.; Kiyoshi, A.; Michiaki, H. PBSIM: PacBio reads simulator—Toward accurate genome assembly. *Bioinformatics* **2013**, *29*, 119–121.
20. Jiao, X.D.; He, X.; Qin, B.D.; Liu, K.; Wu, Y.; Liu, J.; Hou, T.; Zang, Y.S. The prognostic value of tumor mutation burden in EGFR-mutant advanced lung adenocarcinoma, an analysis based on cBioPortal data base. *J. Thorac. Dis.* **2019**, *11*, 4507–4515. [CrossRef]
21. Koren, S.; Schatz, M.C.; Walenz, B.P.; Martin, J.; Howard, J.T.; Ganapathy, G.; Wang, Z.; Rasko, D.A.; McCombie, W.R.; Jarvis, E.D.; et al. Hybrid error correction and de novo assembly of single-molecule sequencing reads. *Nat. Biotechnol.* **2012**, *30*, 693–700. [CrossRef]
22. Morton, N.E.; Zhang, W.; Taillon-Miller, P.; Ennis, S.; Kwok, P.Y.; Collins, A. The optimal measure of allelic association. *Proc. Natl. Acad. Sci. USA* **2001**, *98*, 5217–5221. [CrossRef] [PubMed]
23. Ahsan, M.U.; Liu, Q.; Fang, L.; Wang, K. NanoCaller for accurate detection of SNPs and indels in difficult-to-map regions from long-read sequencing by haplotype-aware deep neural networks. *Genome Biol.* **2021**, *22*, 261. [CrossRef] [PubMed]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.