ricME

Long-Read Based Mobile Element Variant Detection Using Sequence Realignment and Identity Calculation

Ma, Huidong; Zhong, Cheng; Sun, Hui; Chen, Danyang; Lin, HaiXiang

DOI
10.1007/978-981-99-7074-2_13

Publication date
2023

Document Version
Final published version

Published in
Bioinformatics Research and Applications - 19th International Symposium, ISBRA 2023, Proceedings

ricME: Long-Read Based Mobile Element Variant Detection Using Sequence Realignment and Identity Calculation

Huidong Ma[1,2], Cheng Zhong[1,2](✉), Hui Sun[3], Danyang Chen[1,2], and Haixiang Lin[4]

[1] School of Computer, Electronic and Information, Guangxi University, Nanning, Guangxi, China
chzhong@gxu.edu.cn

[2] Key Laboratory of Parallel, Distributed and Intelligent Computing in Guangxi Universities and Colleges, Nanning, Guangxi, China

[3] College of C.S., ICIC, Nankai-Orange D.T. Joint Lab, TMCC, SysNet, Nankai University, Tianjin, China

[4] Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, Delft, Netherlands

Abstract. The mobile element variant is a very important structural variant, accounting for a quarter of structural variants, and it is closely related to many issues such as genetic diseases and species diversity. However, few detection algorithms of mobile element variants have been developed on third-generation sequencing data. We propose an algorithm ricME that combines sequence realignment and identity calculation for detecting mobile element variants. The ricME first performs an initial detection to obtain the positions of insertions and deletions, and extracts the variant sequences; then applies sequence realignment and identity calculation to obtain the transposon classes related to the variant sequences; finally, adopts a multi-level judgment rule to achieve accurate detection of mobile element variants based on the transposon classes and identities. Compared with a representative long-read based mobile element variant detection algorithm rMETL, the ricME improves the F1-score by 11.5 and 21.7% on simulated datasets and real datasets, respectively.

Keywords: mobile element variants · sequence realignment · identity calculation · third-generation sequencing data

1 Introduction

Transposons are DNA sequences that can move autonomously across the genome. Transposons show an important component of the human genome, which occupy approximately half of the human genome [1]. The transposons that have been verified to remain active in the human genome include three classes, Alu, LINE-1 (L1) and SINE-VNTR-Alu (SVA) [1]. The variants caused by transposon position changes are called mobile

X. Guo et al. (Eds.): ISBRA 2023, LNBI 14248, pp. 165–177, 2023.
https://doi.org/10.1007/978-981-99-7074-2_13

element (ME) variants, which can be divided into mobile element insertion (MEI) variants and mobile element deletion (MED) variants. ME variants have demonstrated to be closely associated with various human genetic diseases such as hemophilia and neurofibromatosis [2, 3]. In addition, ME variants account for about a quarter of the overall structural variants [4]. Therefore, it is of great practical importance to carry out research on the detection algorithm of ME variants.

The representative algorithms for detecting ME variants on next-generation sequencing (NGS) data are Tea [3], MELT [4], Mobster [5], and Tangram [6]. Tea alignments the paired-end reads with the reference genome and the assembled sequences composed of ME sequences respectively, and extracts repeat-anchored mate reads and clipped reads based on the alignment information to determine the insertion mechanism of MEs [3]. MELT uses discordant read pairs from NGS data alignment information to detect MEI variants, filters ME variants based on proximity to known ME variants, sequencing depth, and mapping quality of reads; and finally uses discordant sequence pairs and split reads to determine the type of ME variants and precise breakpoints [4]. Mobster uses the discordant reads in the NGS data as the signals of MEI variants, and then extracts the variant sequences based on the signals and compares the variant sequences with the transposon consensus sequence (TCS) to determine the type of ME variations [5]. Tangram designs different ME variation features extraction methods for the read pair and split read in the NGS data alignment information respectively, and then determines the position and type of MEI variants according to the features [6].

Third-generation sequencing (TGS) data has great potential for structural variant detection. It has been shown that structural variant detection algorithms based on TGS data are better than those based on NGS data [7, 8]. rMETL [9] is a representative long-read based ME variant detection algorithm, which is divided into four steps. The first step extracts the candidate ME variant sequences from the long-read alignment file. The second step clusters the candidate variant sequences to determine the ME variant positions. The third step uses the read alignment tool NGMLR [10] to realign the candidate variant sequences with transposon consistency sequences. And the final step counts the transposon classes of the mapped sequences and selects the transposon class with the highest frequency as the type of the ME variant.

However, the existing long-read based ME variant detection algorithms encounter the following problems:

(1) Some insertions (INSs) or deletions (DELs) were not effectively detected, which will directly affect the recall of the final removable element variant detection;
(2) Some of the variant sequences could not be successfully realigned by NGMLR, which could not provide more judgment basis for the final ME variant detection to improve the recall and accuracy of the final detection results.
(3) If the maximum number of transposon class is zero, i.e., all the variant sequences are not mapped with the TCS, the variant may be missed as a false-negative mobile component variant. And if the transposon class with the highest number of occurrences is not unique, the variant may be misclassified as a false positive ME variant when randomly selecting a transposon as the final ME class.

To address the above-mentioned issues, we propose an ME variants detection algorithm ricME using sequence realignment and identity calculation to effectively improve the performance of detecting ME variants.

The remainder of this paper is organized as follows. Section 2 details the proposed ME variant detection algorithm ricME. Section 3 describes the experimental environment, dataset, results and analysis. Section 4 concludes the paper.

2 Method

Our proposed ricME algorithm comprises the following four steps.

Firstly, the long-read based structural variant detection algorithm cnnLSV [11] is executed on the read alignment file *Dset* to obtain the initial detection result *Cset* containing only INSs and DELs. Secondly, the ricME applies different sequence extraction methods according to the characteristics of INSs and DELs to extract variant sequences within *Cset*, and save the sequences into the set *S*. Thirdly, all variant sequences in the set *S* are realigned with TCS using the tool NGMLR. For the variant sequences that can be successfully aligned, the transposon class *te* mapped by the sequences is stored in the set *Tset*; for the variant sequences that are not aligned, the ricME calculates their identity with TCS, and the potential transposon class *pte* corresponding to the maximum identity is selected and deposited into the set *Tset*. Finally, the ricME uses a multilevel judgment rule to determine the final class of ME based on the distribution of *te*, *pte* and identity in the set *Tset*. Figure 1 shows the process of algorithm ricME.

2.1 Initial Variant Position Detection

Since ME variants are caused by transposons moving autonomously across the genome, ME variants are essentially special INSs and DELs. And the sequences of variants are highly similar to transposon families including Alu, L1 and SVA. Therefore, the first step of algorithm ricME is to initially identify INS and DEL position.

In our previous work, we proposed an algorithm called cnnLSV [11] to detect structural variants by encoding long-read alignment information and modeling convolutional neural network. Experiments have shown that the algorithm cnnLSV has a high overall F1-score compared to other existing algorithms. We use cnnLSV to detect the structural variants on sequence alignment file *Dset*, and save the INSs and DELs to the set *Cset*.

2.2 Variant Sequence Extraction

The algorithm ricME using the following method to extract the variant sequences for INS and DELs in *Cset*.

Variant Sequence Extraction for INSs. As shown in Fig. 2(a), the ricME extracts variant sequences using intra- and inter-alignment signatures.

(1) Extracting variant sequences based on intra-alignment signatures. The flag “I” in the CIGAR strings in the alignment information indicates the INS, while the number preceding the flag represents the length of the variant. The ricME searches all read

alignment information around the INSs in *Dset* to obtain the variant sequences with a length of more than 50 base pairs (bps) within the CIGAR strings, and saves the information about the variant sequences *seq* to the FA format file.

(2) Extracting variant sequences based on the inter-alignment signatures. When long-reads are aligned to the reference genome, long-reads that span structural variants may split into multiple segments. According to the characteristics of INSs, the distance between two segments from the same read will change before and after the alignment, and the redundant segments that cannot be aligned are exactly the INS variant sequence *seq*. In addition to the above case, a read alignment situation around the INS is also mentioned in the algorithm rMETL. When the sequencing reads cannot span the whole INS segment, there will be a fragment that can be aligned successfully, and the adjacent fragment is clipped off because it is located at the boundary of the variant region. The sequence is clipped off, and it is exactly the INS sequence *seq*. The ricME extracts the variant sequences according to the above two cases, and saves the information related to the variants to the FA format file.

Variant Sequence Extraction for DELs. Unlike the existing algorithm rMETL, which searches for variant features before detecting variant position, our algorithm ricME utilizes the detection results *Cset* of the existing structural variant detection algorithm cnnLSV to obtain the initial positions of DELs. Therefore, the ricME can directly intercept the variant sequence *seq* in the corresponding region of the reference genome based on the chromosome number *chr*, variant position *pos*, and variant length *svl* where the DEL occurs in the *Cset*, as shown in Fig. 2(b). And last, the ricME saves the information of the DEL to the FA format file.

2.3 Sequence Realignment and Identity Calculation

In algorithm rMETL, sequence realignment means that the variant sequences extracted from the long-read alignment information are realigned to TCS by NGMLR to obtain the transposon class *te*. The rMETL could judge the ME variant class according the distribution of the *te*.

However, some of the variant sequences were discarded because they could not be successfully aligned to TCS by NGMLR, i.e., they could not provide more basis for ME variant type judgment. In addition, relying only on the distribution of *te* to determine the ME class may lead to accidental errors. As shown in Fig. 3(a), three sequences seq_1, seq_3 and seq_6 are finally judged as the Alu class, and the other three sequences seq_4, seq_7 and seq_8 are aligned to the L1 class, i.e., the number of variant sequences supporting Alu and L1 classes are of equal sizes, which will lead to the difficulty for the algorithm rMETL to determine the final ME variant class. To solve the above problem, the algorithm ricME introduces the sequence identity calculation based on sequence realignment, as shown in Fig. 3(b).

Sequence identity reflects the degree of similarity between two sequences. The premise of sequence identity calculation is to align two sequences [12, 13]. The two-sequence alignment algorithms include global-based and local-based alignment. Due to the large differences between the lengths of the three transposons and the lengths of each variant sequence, if the global based approach is used to calculate the identity,

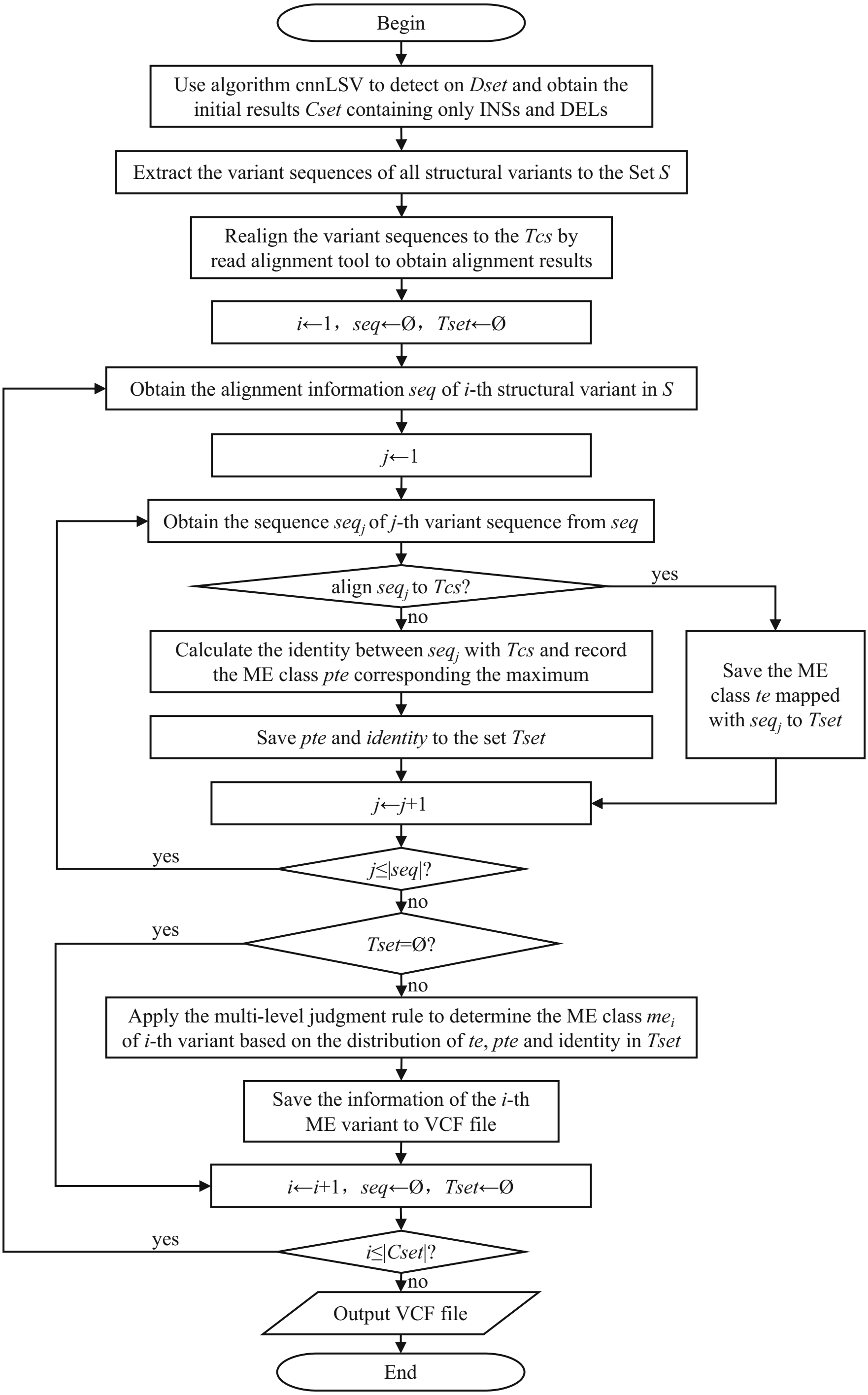
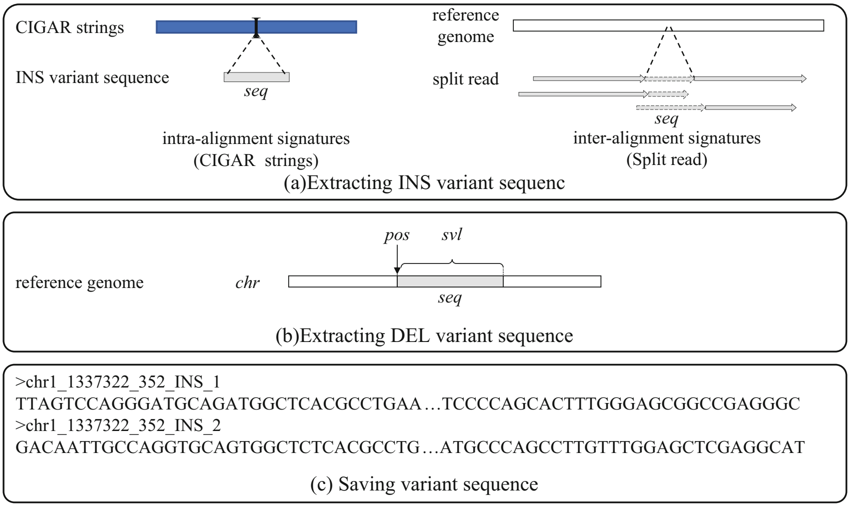

Begin
Use algorithm cnnLSV to detect on *Dset* and obtain the initial results *Cset* containing only INSs and DELs
Extract the variant sequences of all structural variants to the Set *S*
Realign the variant sequences to the *Tcs* by read alignment tool to obtain alignment results
i←1，*seq*←Ø，*Tset*←Ø
Obtain the alignment information *seq* of *i*-th structural variant in *S*
j←1
Obtain the sequence seq_j of *j*-th variant sequence from *seq*
align seq_j to *Tcs*?
yes
no
Calculate the identity between seq_j with *Tcs* and record the ME class *pte* corresponding the maximum
Save *pte* and *identity* to the set *Tset*
Save the ME class *te* mapped with seq_j to *Tset*
j←*j*+1
yes
j≤|*seq*|?
no
yes
Tset=Ø?
no
Apply the multi-level judgment rule to determine the ME class me_i of *i*-th variant based on the distribution of *te*, *pte* and identity in *Tset*
Save the information of the *i*-th ME variant to VCF file
i←*i*+1，*seq*←Ø，*Tset*←Ø
yes
i≤|*Cset*|?
no
Output VCF file
End

Fig. 1. Procedure of proposed mobile element variant detection algorithm ricME


CIGAR strings
INS variant sequence
seq
intra-alignment signatures
(CIGAR strings)
reference
genome
split read
seq
inter-alignment signatures
(Split read)
(a)Extracting INS variant sequenc
pos
svl
reference genome
chr
seq
(b)Extracting DEL variant sequence
(c) Saving variant sequence

```
>chr1_1337322_352_INS_1
TTAGTCCAGGGATGCAGATGGCTCACGCCTGAA…TCCCCAGCACTTTGGGAGCGGCCGAGGGC
>chr1_1337322_352_INS_2
GACAATTGCCAGGTGCAGTGGCTCTCACGCCTG…ATGCCCAGCCTTGTTTGGAGCTCGAGGCAT
```

Fig. 2. Variant sequence extraction and storage in FA format

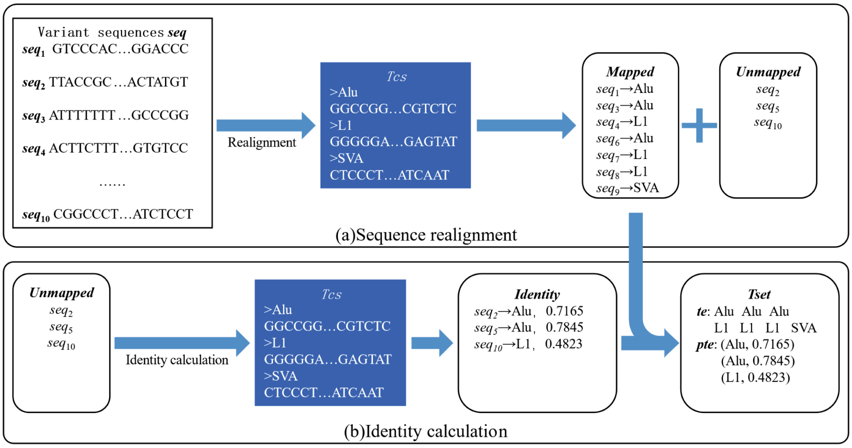


Fig. 3. Example of sequence realignment and identity calculation

the longer variant sequence will always get a higher identity with the longer transposon sequence. To avoid this bias caused by sequence length, the algorithm ricME uses the local-based alignment algorithm based on affine gap penalty to calculate the identity of two sequences. The ricME calculates the identity between unmapped variant sequences and the TCS to obtain the potential ME class *pte* and the corresponding identity *identity*, and stores the *te*, *pte*, and *identity* in the set *Tset*.

2.4 Mobile Element Variant Determination

In order to avoid the random selection error caused by the same frequency of transposon class *te*, the algorithm ricME introduces the potential variant class *pte* and identity *identity* as judgment factors, and proposes a multi-level judgment rule based on three factors of *te*, *pte* and *identity* to accurately detect the ME variant.

The process of the ME class judgment rule is as follows.

(1) To improve the accuracy of detection, the ricME defines the identity threshold $identity_0$ and eliminates the *pte* with $identity < identity_0$ within *Tset*.
(2) The ricME constructs 3 triples T_{Alu}, T_{L1} and T_{SVA} according to the distribution of *te*, *pte* and *identity* within *Tset*, where the triple $T_{me} = (n_{te}, n_{pte}, score)$, $me \in \{\text{Alu, L1, SVA}\}$, n_{te} denotes the number of *te* in *Tset* with class *me*, n_{pte} denotes the number of *pte*, and *score* denotes the *sum* of *identity* corresponding to the *pte* of class *me*.
(3) A multilevel judgment rule is used to determine the ME variant class. The ricME ranks the three tuples T_{Alu}, T_{L1} and T_{SVA} in the order of priority of n_{te}, $n_{pte,}$ and *score*.
(4) The ricME stores the ME variant information and transposon class to the VCF file as the final detection output.

3 Experiment

3.1 Experimental Environment and Data

Experimental Environment. The experiment was carried out on the computing node X580-G30 with CPU 2 × Intel Xeon Gold 6230, GPU 2 × Tesla T4, and main memory 192GB DDR4 of Sugon 7000A parallel computer cluster system at Guangxi University. The running operating system is CentOS 7.4. The proposed method was implemented by Python3.8 programming. The PyTorch was used to train and test the constructed network model.

Datasets

Simulated Datasets. Referring to the work of rMETL [9], the simulated datasets were generated as follows. Firstly, the 20,000 real ME variants of classes Alu, L1 and SVA, respectively, were collected from the database RepeatMasker [14]. And the sequences corresponding to the positions of the selected ME variants were deleted in chromosome 1 of the human reference genome. We extracted the sequence of chromosome 1 of the normal reference genome as seq_1, the sequence of chromosome 1 containing the MED variants as seq_2, and recorded the chromosomes, positions, lengths and classes of the real ME variants as the ground truth set. Secondly, we executed the tool PBSIM [15]to simulate sequencing reads for seq_1 to generate 4 PacBio CLR datasets with coverage of 50×, 30×, 20×, and 10×, respectively, and executed read alignment tool NGMLR to map the 4 datasets to the seq_2 to generate 4 long read alignment files for detecting MEI variants. Thirdly, we also used the PBSIM to simulate sequencing reads for seq_2 to generate 4 datasets with coverage of 50×, 30×, 20×, and 10×, respectively, and executed NGMLR to align the 4 datasets to the seq_1 to generate 4 read alignment files for detection MED variants. Finally, we used SAMtools [16] to sort and generate indexes for the 8 simulated datasets.

Real Datasets. The real dataset used is the HG002 CCS [17] dataset generated by the PacBio platform, which relates to the ground truth set that is a portion of the ME variants of HG002 dataset validated in [18]. This ground truth set contains 1353 Alu, 197 L1, and 90 SVA ME variants.

3.2 Detecting Performance Evaluation Metrics

The experiments use the precision *Pre*, recall *Rec* and F1-score $F1$ as the detection performance evaluation metric.

In the determination of true positive ME variant, when the detected ME variant *call* and the ME variant *base* of the ground truth set satisfy Eq. 1, then *call* is considered a true positive variant, otherwise it is a false positive variation:

$$\begin{cases} call_t = base_t \\ call_m = base_m \\ call_c = base_c \\ \min(call_e + 1000, base_e) - max(call_s - 1000, base_s) \geq 0 \\ \frac{\min(call_l, base_l)}{\max(call_l, base_l)} \geq 0.7 \end{cases} \tag{1}$$

where $call_t$, $call_m$, $call_c$, $call_s$, $call_e$ and $call_l$ denote variant type, ME class, chromosome, start position, end position and the length of variant *call*, respectively, and $base_t$, $base_m$, $base_c$, $base_s$, $base_e$ and $base_l$ represent variant type, ME class, chromosome, start position, end position and the length of variant *base*, respectively, $call_m \in$ {Alu, L1, SVA} and $base_m \in$ {Alu, L1, SVA}.

3.3 Experimental Results

Experiments on Simulated Datasets. The experiments were conducted on eight simulated datasets, including four datasets containing MEI variants with coverages of 50× , 30×, 20×, and 10×, respectively, and four datasets containing MED variants with coverages of 50×, 30×, 20×, and 10×, respectively.

Firstly, to verify the effectiveness of the algorithm ricME in detecting the initial variant position, we looked at the detection results of simple INSs and DELs. The results are shown in Table 1, where *TP-call* indicates the number of correctly detected variants in the detection results, and *TP-base* represents the number of correctly detected variants in the ground truth set.

From Table 1, it can be concluded that for the cases of INSs and DELs, the F1-score of the algorithm ricME is about 12–17% higher than that of the algorithm rMETL. For the INSs, the detection performance of algorithm rMETL, especially the recall, decreases significantly as the coverage of the dataset decreases. In contrast, algorithm ricME performed significantly better than algorithm rMETL in terms of recall *Rec*, especially at low coverage, and could detect more than 4000 INSs and had about 23%

higher recall than rMETL. Even though the *Pre* was slightly lower than that of rMETL under partial coverages, the larger recall by the ricME results in a significant higher $F1$ values. For the DELs, both algorithms rMETL and ricME achieved very high detection accuracy on the datasets with different coverages, especially rMETL could reach 99%. In terms of recall *Rec*, the algorithm ricME had a significant advantage over the algorithm rMETL, especially in the low coverage dataset, with a lead of about 13%. The $F1$ values of ricME were also higher than that of algorithm rMETL by about 6–8%. Compared with the similar algorithm rMETL, ricME had higher $F1$ values in the detection of INSs and DELs.

Table 1. Results of algorithms in detecting INSs and DELs on simulated datasets

Type	Coverage	Algorithm	*TP-Call*	*TP-Base*	*FP*	*FN*	*Pre* (%)	*Rec* (%)	$F1$ (%)
INS	50 ×	rMETL	13942	15013	1128	4987	92.515	75.065	82.881
		ricME	**18493**	**18692**	**694**	**1308**	**96.383**	**93.46**	**94.899**
	30 ×	rMETL	13100	14168	1046	5832	92.606	70.840	80.274
		ricME	**18218**	**18482**	**624**	**1518**	**96.688**	**92.410**	**94.501**
	20 ×	rMETL	12182	13235	916	6765	93.007	66.175	77.329
		ricME	**18038**	**18433**	**573**	**1567**	**96.921**	**92.165**	**94.483**
	10 ×	rMETL	9949	10944	636	9056	93.991	54.720	69.170
		ricME	**14980**	**15408**	**399**	**4592**	**97.406**	**77.040**	**86.034**
DEL	50 ×	rMETL	16472	16804	**129**	3196	**99.223**	84.02	90.991
		ricME	**18999**	**19017**	574	**983**	97.067	**95.085**	**96.066**
	30 ×	rMETL	16630	16919	**171**	3081	**98.982**	84.595	91.225
		ricME	**19026**	**19072**	619	**928**	96.849	**95.360**	**96.099**
	20 ×	rMETL	16318	16658	**122**	3342	**99.258**	83.290	90.576
		ricME	**18808**	**18913**	441	**1087**	97.709	**94.565**	**96.111**
	10 ×	rMETL	13500	13918	**60**	6082	**99.558**	69.590	81.919
		ricME	**16388**	**16583**	259	**3417**	98.444	**82.915**	**90.015**

Note that the values in bold represent the best results.

Next, the performance of two algorithms rMETL and ricME is compared for detecting ME variants. The experiment results are shown in Table 2.

From Table 2, we can see that the performance of the algorithm ricME was significantly better than that of the algorithm rMETL in detecting MEI variants and MED variants on datasets with different coverages, with 8–11% higher $F1$ values. For MEI variants, in terms of the detection precision *Pre*, the ricME slightly outperformed rMETL on high coverage datasets, while both performed comparably on low coverage datasets. In terms of recall *Rec*, the algorithm ricME was significantly higher than rMETL, especially about 15% higher on the low-coverage datasets. In addition, the value of metric

TP-base also shows that the ricME detected about 3000 more true positives than rMETL. For the overall performance metric F1-score, the ricME also obtained higher $F1$ values due to the high precision and recall. For the MED variants, in terms of the detection precision *Pre*, the rMETL achieved a high detection precision on all datasets, which was about 2% higher than that of ricME. However, in terms of recall *Rec*, algorithm ricME achieved significantly higher than rMETL on the datasets with each coverage, namely about 7–8% higher. In terms of the $F1$ values, the algorithm ricME obtained higher $F1$ values due to the overall high precision and recall, namely about 2–5% higher than the algorithm rMETL. The combined experiment results in Table 2 show that the algorithm ricME had a better performance in terms of recall *Rec* and was comparable to the algorithm rMETL in terms of detection precision *Pre*. This indicates that the proposed algorithm ricME is able to detect more ME variants than the algorithm rMETL, and has higher accuracy in the variant class judgment stage. The above results show that the proposed algorithm ricME use the sequence realignment and identity calculation to enhance the basis for variant class judgment, which improves the detection precision, recall and F1-score to achieve higher detection performance.

Table 2. Results of algorithms in detecting mobile element variants on simulated datasets

Type	Coverage	Algorithm	*TP-call*	*TP-base*	*FP*	*FN*	*Pre* (%)	*Rec* (%)	$F1$ (%)
MEI	50 ×	rMETL	13785	14837	1285	5163	91.473	74.185	81.927
		ricME	**16894**	**17432**	**1235**	**2568**	**93.188**	**87.16**	**90.073**
	30 ×	rMETL	12964	14017	**1182**	5983	91.644	70.085	79.428
		ricME	**16643**	**17150**	1294	**2850**	**92.786**	**85.750**	**89.129**
	20 ×	rMETL	12047	13085	**1051**	6915	**91.976**	65.425	76.461
		ricME	**16314**	**16861**	1428	**3139**	91.951	**84.305**	**87.962**
	10 ×	rMETL	9831	10807	**754**	9193	**92.877**	54.035	68.321
		ricME	**13397**	**13911**	1148	**6089**	92.107	**69.555**	**79.258**
MED	50 ×	rMETL	16388	16717	**213**	3283	**98.717**	83.585	90.523
		ricME	**17696**	**17875**	677	**2125**	96.315	**89.375**	**92.715**
	30 ×	rMETL	16542	16830	**259**	3170	**98.458**	84.150	90.744
		ricME	**17715**	**17914**	724	**2086**	96.074	**89.570**	**92.708**
	20 ×	rMETL	16238	16577	**202**	3423	**98.771**	82.885	90.133
		ricME	**17507**	**17742**	586	**2258**	96.761	**88.710**	**92.561**
	10 ×	rMETL	13437	13858	**123**	6142	**99.093**	69.290	81.554
		ricME	**15273**	**15562**	400	**4438**	97.448	**77.810**	**86.529**

Note that the values in bold represent the best results.

Experiments on Real Datasets. We used the tool SAMtools to downsample the long-read alignment file HG002 CCS 28 × to generate a new dataset with 10 × coverage, then executed the algorithms ricME and rMETL to detect ME variants on the two datasets, and compared the detection results with the ground truth set to evaluate the detection performance. The detection results of the two algorithms rMETL and ricME are shown in Table 3, where “\” indicates that the calculation of $F1$ is meaningless in the case that both metrics *Pre* and *Rec* are zero.

Table 3. Detection results of algorithms rMETL and ricME on dataset HG002

Type	Coverage	Algorithm	*TP-call*	*TP-base*	*FP*	*FN*	*Pre* (%)	*Rec* (%)	$F1$ (%)
28 ×	Alu	rMETL	4	4	**364**	1349	1.087	0.296	0.465
		ricME	**589**	**589**	1898	**764**	**23.683**	**43.533**	**30.677**
	L1	rMETL	14	14	**292**	183	4.575	7.107	5.567
		ricME	**59**	**59**	946	**138**	**5.871**	**29.949**	**9.817**
	SVA	rMETL	1	1	**101**	89	0.98	1.111	1.042
		ricME	**32**	**32**	1011	**58**	**3.068**	**35.556**	**5.649**
	All	rMETL	19	19	**757**	1621	2.448	1.159	1.573
		ricME	**680**	**680**	3855	**960**	**14.994**	**41.463**	**22.024**
10 ×	Alu	rMETL	2	2	**55**	1351	3.509	0.148	0.284
		ricME	**657**	**657**	2198	**696**	**23.012**	**48.559**	**31.226**
	L1	rMETL	1	1	**65**	196	1.515	0.508	0.76
		ricME	**62**	**62**	1053	**135**	**5.561**	**31.472**	**9.451**
	SVA	rMETL	0	0	**14**	90	0	0	\
		ricME	**26**	**26**	1133	**64**	**2.243**	**28.889**	**4.163**
	All	rMETL	3	3	**134**	1637	2.19	0.183	0.338
		ricME	**745**	**745**	4384	**895**	**14.525**	**45.427**	**22.012**

Note that the values in bold represent the best results.

As can be seen from Table 3, the overall detection performance of the algorithm ricME was significantly higher than that of the algorithm rMETL for the detection of the all classes of ME variants on the real datasets with coverages 28 × and 10 ×. The F1-scores of ricME was about 20% higher than that of rMETL. In terms of recall *Rec*, the ricME was much higher than rMETL, especially for the detection of Alu transposon class, which are 40% higher. In terms of precision *Pre*, although the *Pre* of the ricME was higher than that of rMETL in all classes, both performed less well. For the rMETL, the main reason is that the rMETL detected fewer true positive variants, i.e., insufficient detection of ME variants. For the algorithm ricME, the main reason for the low accuracy is the high number of false positive variants detected. However, it is worth noting that despite the high frequency and importance of ME variants, there are still few studies and annotations on ME variants [19, 20]. The ground truth set corresponding to the real dataset HG002 used in the experiments is the validated information of ME variants given in the work [18], and this ground truth set only contains some of the ME variants with a high confidence. This means that the false positives detected by the ricME may not actually mean that no variants have occurred.

4 Conclusion

Mobile element variant is a very import structural variant that is closely associated with a variety of genetic diseases. We propose the ricME, an algorithm for detecting the variation of movable components that integrates re-matching and sequence consistency calculation, to improve the existing representative ME variation detection algorithm rMETL. The ricME has the following features and innovations. First, the ricME use the detection results of algorithm cnnLSV to obtain the initial results with high recall. Secondly, the ricME extracts the variant sequences of all INSs and DELs in initial results. Thirdly, the ricME realigns and calculates identity between variant sequences with transposon consistency sequences to obtain the corresponding transposon classes and the identities. Finally, the ricME applies a multi-level judgment rule to determine the final ME class based on transposon classes, potential transposon classes and identities. The experiment results show that the proposed algorithm ricME outperforms the existing representative algorithm for ME variant detection in general. In the future, we will investigate algorithms for detecting more types of structural variants on more types of datasets.

Acknowledgement. This work is partly supported by the National Natural Science Foundation of China under Grant No. 61962004 and Guangxi Postgraduate Innovation Plan under Grant No. A30700211008.

References

1. Niu, Y., Teng, X., Zhou, H., et al.: Characterizing mobile element insertions in 5675 genomes. Nucleic Acids Res. **50**(5), 2493–2508 (2022)
2. Hancks, D.C., Kazazian, H.H.: Roles for retrotransposon insertions in human disease. Mob. DNA **7**(1), 1–28 (2016)
3. Lee, E., Iskow, R., Yang, L., et al.: Landscape of somatic retrotransposition in human cancers. Science **337**(6097), 967–971 (2012)
4. Gardner, E.J., Lam, V.K., Harris, D.N., et al.: The Mobile Element Locator Tool (MELT): population-scale mobile element discovery and biology. Genome Res. **27**(11), 1916–1929 (2017)
5. Thung, D.T., de Ligt, J., Vissers, L.E.M., et al.: Mobster: accurate detection of mobile element insertions in next generation sequencing data. Genome Biol. **15**(10), 1–11 (2014)
6. Wu, J., Lee, W.P., Ward, A., et al.: Tangram: a comprehensive toolbox for mobile element insertion detection. BMC Genom. **15**, 1–15 (2014)
7. Mahmoud, M., Gobet, N., Cruz-Dávalos, D.I., et al.: Structural variant calling: the long and the short of it. Genome Biol. **20**(1), 1–14 (2019)
8. Merker, J.D., Wenger, A.M., Sneddon, T., et al.: Long-read genome sequencing identifies causal structural variation in a Mendelian disease. Genet. Med. **20**(1), 159–163 (2018)

9. Jiang, T., Liu, B., Li, J., et al.: RMETL: sensitive mobile element insertion detection with long read realignment. Bioinformatics **35**(18), 3484–3486 (2019)
10. Sedlazeck, F.J., Rescheneder, P., Smolka, M., et al.: Accurate detection of complex structural variations using single-molecule sequencing. Nat. Methods **15**(6), 461–468 (2018)
11. Ma, H., Zhong, C., Chen, D., et al.: CnnLSV: detecting structural variants by encoding long-read alignment information and convolutional neural network. BMC Bioinform. **24**(1), 1–19 (2023)
12. Li, H.: Minimap2: pairwise alignment for nucleotide sequences. Bioinformatics **34**(18), 3094–3100 (2018)
13. Altschul, S.F., Erickson, B.W.: Optimal sequence alignment using affine gap costs. Bull. Math. Biol. **48**, 603–616 (1986)
14. Smit, A.F.A., Hubley, R., Green, P.: RepeatMasker Open-4.0. 2013–2015. http://www.repeatmasker.org
15. Ono, Y., Asai, K., Hamada, M.: PBSIM: PacBio reads simulator—toward accurate genome assembly. Bioinformatics **29**(1), 119–121 (2013)
16. Danecek, P., Bonfield, J.K., Liddle, J., et al.: Twelve years of SAMtools and BCFtools. Gigascience **10**(2), giab008 (2021)
17. Zook, J.M., Catoe, D., McDaniel, J., et al.: Extensive sequencing of seven human genomes to characterize benchmark reference materials. Scientific Data **3**(1), 1–26 (2016)
18. Chu, C., Borges-Monroy, R., Viswanadham, V.V., et al.: Comprehensive identification of transposable element insertions using multiple sequencing technologies. Nat. Commun. **12**(1), 3836 (2021)
19. Hoen, D.R., Hickey, G., Bourque, G., et al.: A call for benchmarking transposable element annotation methods. Mob. DNA **6**, 1–9 (2015)
20. Ou, S., Su, W., Liao, Y., et al.: Benchmarking transposable element annotation methods for creation of a streamlined, comprehensive pipeline. Genome Biol. **20**(1), 1–18 (2019)