RepLong - de novo repeat identification using long read sequencing data

Guo, Rui; Li, Yan-Ran; He, Shan; Ou-Yang, Le; Sun, Yiwen; Zhu, Zexuan

DOI:
10.1093/bioinformatics/btx717

RepLong - *de novo* repeat identification using long read sequencing data

Rui Guo [1,†], Yan-Ran Li [1,†], Shan He [2,3], Le Ou-Yang [4], Yiwen Sun [5,*], and Zexuan Zhu [1,*]

[1] College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China
[2] School of Computer Science, University of Birmingham, Birmingham B17 2TT, U.K.
[3] Centre for Computational Biology, University of Birmingham, Birmingham B15 2TT, U.K.
[4] College of Information Science, Shenzhen University, Shenzhen 518060, China
[5] School of Medicine, Shenzhen University, Shenzhen 518060, China

[*] To whom correspondence should be addressed.
[†] The authors wish it to be known that, in their opinion, the first two authors should be regarded as Joint First Authors

Abstract

Motivation: The identification of repetitive elements is important in genome assembly and phylogenetic analyses. The existing *de novo* repeat identification methods exploiting the use of short reads are impotent in identifying long repeats. Since long reads are more likely to cover repeat regions completely, using long reads is more favorable for recognizing long repeats.
Summary: In this study, we propose a novel *de novo* repeat elements identification method namely RepLong based on PacBio long reads. Given that the reads mapped to the repeat regions are highly overlapped with each other, the identification of repeat elements is equivalent to the discovery of consensus overlaps between reads, which can be further cast into a community detection problem in the network of read overlaps. In RepLong, we first construct a network of read overlaps based on pair-wise alignment of the reads, where each vertex indicates a read and an edge indicates a substantial overlap between the corresponding two reads. Secondly, the communities whose intra connectivity is greater than the inter connectivity are extracted based on network modularity optimization. Finally, representative reads in each community are extracted to form the repeat library. Comparison studies on Drosophila melanogaster and human long read sequencing data with genome-based and short-read-based methods demonstrate the efficiency of RepLong in identifying long repeats. RepLong can handle lower coverage data and serve as a complementary solution to the existing methods to promote the repeat identification performance on long-read sequencing data.
Availability: The software of RepLong is freely available at https://github.com/ruiguo-bio/replong .
Contact: zhuzx@szu.edu.cn
Supplementary information: Supplementary data are available online.

1 Introduction

Repetitive DNA sequences are segment sequences occuring more than once in a genome. Based on their organization, repetitive DNA sequences can be divided into interspersed repeats and tandem repeats. Interspersed repeats are highly identical and separately distributed in the genome. They are mainly derived from transposable elements (TEs), i.e., mobile genetic elements in eukaryotic genomes. Tandem repeats are adjacent to each other and composed of satellites and simple sequence repeats (Schlötterer, 2000). Repeats account for a large proportion of many eukaryotic genomes, e.g., about 50% of the human genome (Lander *et al.*, 2001), more than 80% of the maize genome (Schnable *et al.*, 2009), and about 20% of the fruit fly genome consist of TEs (Bergman *et al.*, 2006).

Repeat identification is crucial to phylogenetic analyses and can help to infer the underlying relationships of some closely related species (Feschotte and Pritham, 2007). Repeats are key players in generating genomic novelty (Bennetzen and Wang, 2014) and modulating human RNA abundance and splicing (Kelley *et al.*, 2014). The prior knowledge of repeats in a genome allows a rough estimate of the complexity of the genome (Eddy, 2012) and alleviates the misassembled rearrangements (Treangen and Salzberg, 2011).

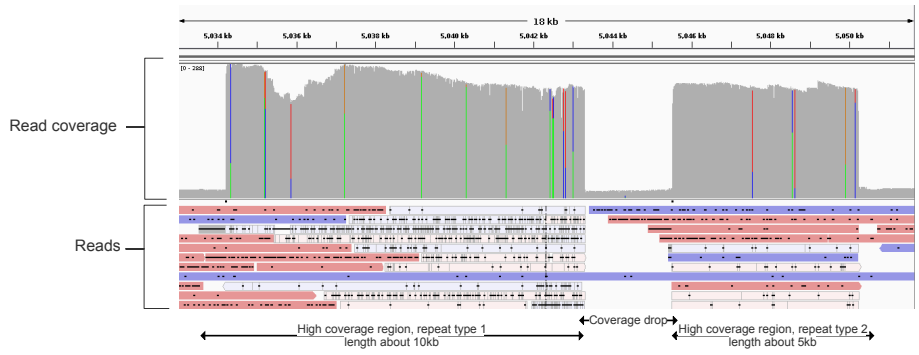


Fig. 1. Reads mapped to two repeat regions of a reference genome in Integrative Genomics Viewer. The upper part shows the coverage of reads, and the lower part depicts the reads. The two bulges of coverage indicate two repeat regions. There are significant coverage drops at the boundaries of the repeat regions.

Many computational methods have been proposed to identify repetitive DNA sequences and they are widely divided into three categories including *de novo*, homology-based, and structure-based methods (Bergman and Quesneville, 2007). Since *de novo* methods need no prior information of the repeat structure or similarity to known the repeat sequences, they tend to be more flexible than the other two methods. The majority of *de novo* methods are based on query vs. query similarity searches or word counting/seed extension strategy (Feschotte and Pritham, 2007).

In *de novo* methods, two different types of input are considered, i.e. the entire genome and read sequences. Before the invent of the high-throughput sequencing technology, most *de novo* methods are targeted at the entire genome. For example, RECON (Bao and Eddy, 2002) uses the genome multiple alignments and single linkage clustering to extract the repeats. RepeatModeler (http://www.repeatmasker.org/RepeatModeler/) combines two *de novo* repeat finding programs, i.e., RECON and RepeatScout (Price *et al.*, 2005) that employs complementary computational methods for identifying repeat element boundaries and family relationships from sequence data. PILER-DF (Edgar and Myers, 2005) identifies lists of matches covering a maximal contiguous region in the input genome and then looks for intact transposable elements and aligns at least three similar regions to create a repeat. The methods using the whole genome as input rely on the availability and accuracy of the genome. If no accurate genome is available, it is hard for such methods to work precisely.

In recent years, with the advance of high-throughput sequencing technology, a number of new methods have been proposed to use short read sequences directly to extract repeat library. For example, ReAS (Li *et al.*, 2005) uses shotgun sequencing reads and the seed extension method to identify repeats. ReAS was shown to cover only a small proportion of the RepBase library (Jurka *et al.*, 2005), which is manually curated and often used as a benchmark for the evaluation of the identified repeats. P-clouds (de Koning *et al.*, 2011) adopts the oligo frequencies and creates clusters of similar repeated oligos. RepARK (Koch *et al.*, 2014) identifies high frequency k-mers in the short reads and then assembles them to form the repeat library. It covers a large proportion of RepBase library, consisting majorly fragmented sequences with low N50, which is a metric defined as the shortest sequence length at 50% of the read contigs. Tedna (Zytnicki *et al.*, 2014) works as a *de novo* TE assembler of short reads based on de Bruijn graphs. MixTaR (Fertin *et al.*, 2015) detects tandem repeats using short reads by de Bruijn graphs. REPdenovo (Chu *et al.*, 2016), following the k-mer counting and assembly paradigm, assemblies the frequent k-mers into raw configs and builds a directed contig graph to identify repeats.

Repeat identification based on short reads tend to identify fragment repeats, and downstream assembly is necessary to recover long repeats. However, the intrinsic ambiguity of short reads assembly deteriorates the precision of long repeats identification. Compared to short reads, long reads are deemed to have a few advantages (Eid *et al.*, 2009). In particular, the substantial increase of read length enables long read sequencing to solve the ambiguity of assembly and ease the identify long repeats. Unfortunately, the existing k-mer counting and assembly methods that work well in short-read-based repeat identification are not effectively applicable to long reads, due to the lower coverage and higher sequencing error rate of long read data (English *et al.*, 2012). Hence, specific and efficient repeat identification methods based on long reads are highly demanded.

In this paper, a *de novo* repeat identification method using long reads namely RepLong is proposed. The motivation is explained as follows. It is observed that the reads mapped to a reference genome usually form a pile in a repeat region, i.e., with a relatively high coverage and sharp coverage dropping at the boundaries of that region. As depicted in Figure 1, the reads mapped to the same repeat regions are highly overlapped with each other. Inspired by this observation, we can identify the repeat regions by finding the nontrivial consensus overlaps of the read piles even if the reference genome is not provided. Especially, with long reads, the overlap length tend to be longer and the consensus overlaps are more easily recognized. If a network of the reads is constructed with edges indicating the overlaps between the reads, the identification of the consensus overlaps between reads can be cast into a community structure detection problem (Fortunato, 2010; Schaeffer, 2007; Harenberg *et al.*, 2014; Girvan and Newman, 2002) in such network.

Particularly, in RepLong, the overlaps between the long reads are firstly identified based on pair-wise alignment, and secondly a network of reads is constructed to represent the overlap relationships between reads. The repeat identification is then cast into a community detection problem and solved by network modularity optimization. Finally, representative reads in the detected communities are extracted to identify the repeat library. Both Drosophila melanogaster and human repeat library are built using RepLong and PacBio reads (Eid *et al.*, 2009). The results are compared to RepBase repeat library (version 21.04) and the repeat library built by the short-read-based method RepARK. The comparison studies demonstrate the efficiency of RepLong in identifying long repeats. RepLong can work together with short-read-based methods to complement with each other.

The remaining of this paper is organized as follows. Section 2 describes the details of the proposed RepLong, Section 3 presents the experimental results, and finally Section 4 concludes this work.

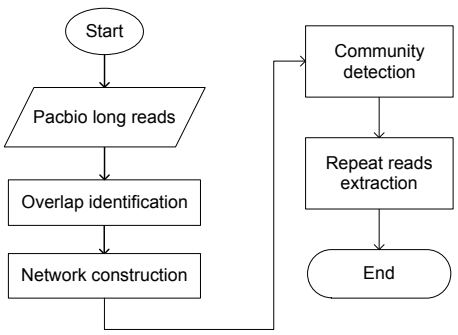


Fig. 2. Schematic diagram of RepLong.

Table 1. Recorded overlap information

Id1	Strand	Starting position	Id2	Strand	Starting position	Overlap length
1	0	5250	2	1	4250	228
1	0	5250	3	1	10250	6436
1	0	5250	4	0	8000	325
1	0	5250	5	1	2250	246
4	0	8000	3	1	10250	300
2	1	4250	3	1	10250	642

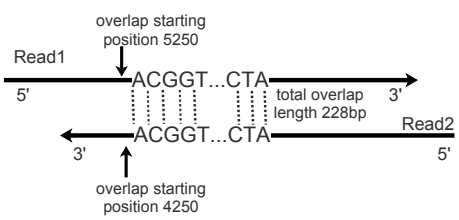


Fig. 3. An overlap between two reads, where Read1 and Read2 overlap for 228 bases.

2 Method

In this section, we describe the *de novo* repeat identification method RepLong based on longs reads and community detection. As shown in Figure 2, RepLong consists of four phases, i.e., reads overlap identification, network construction, community detection, and repeat reads extraction.

2.1 Read overlap identification

Identifying pair-wise overlap of the reads is the prerequisite of finding the consensus overlaps. In RepLong, the MHAP algorithm (Berlin *et al.*, 2015) is used to identify the overlaps of every read pair. MHAP is a computational efficient technique for estimating similarity between long reads. It uses MinHash sketches for efficient alignment filtering and the Jaccard similarity to estimate the number of shared k-mers between two reads. MHAP is implemen ted in Canu genome assember (Koren *et al.*, 2016) and equipped with FALCON(Chin *et al.*, 2016) for PacBio reads error correction. For each overlap, the read ids, strands, starting overlap positions and overlap length are recorded as the examples shown in Table 1. The first example is also depicted in Figure 3, where the positive strand (strand 0) Read1, starting at position 5250, overlaps the negative strand (strand 1) Read2 at position 4250 for 228 bases.

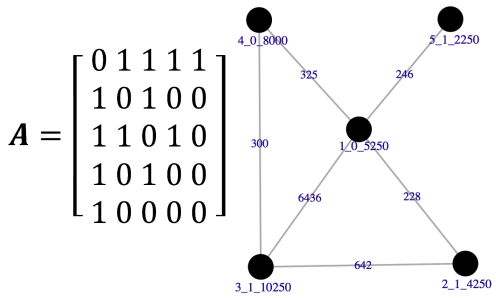


Fig. 4. The network and the corresponding adjacent matrix constructed based on the overlap information provided in Table 1. The node label includes the read id, strand, and starting overlap position. The edge is labelled with the overlap length between the two nodes.

2.2 Network construction

After obtaining the pair-wise overlaps of the reads, a network is constructed to reflect the similarity relationship of the reads. Because MHAP works more precisely on long reads, only overlaps longer than 100bp are retained to build the network. A network is an ordered pair $G = (V, E)$ comprising a set V of vertices or nodes, together with a set E of edges. In G, the nodes represent the reads and the edges indicate the overlaps between the reads. To analyse the connections of network nodes, we define an adjacency matrix A of size $|V| \times |V|$, where $|V|$ is the cardinarity of V. Each element of A is set to one or zero to represent a corresponding connection or disconnection, i.e., A_{ij} is set to one if the overlap length between reads i and j is greater than 100bp, and zero otherwise. The degree of a node i denoted by k_i indicates the number of edges connected to node i, which is equal to the sum of the i-th row of A:

$$k_i = \sum_{j=1}^{|V|} A_{ij} \tag{1}$$

The total number of edges in the network can be computed by

$$m = \frac{1}{2} \sum_{i=1}^{|V|} k_i \tag{2}$$

Figure 4 shows the constructed network and the corresponding adjacent matrix according to the overlap information provided in Table 1. In this network, each node represents a read and an edge between two nodes indicates an overlap between these two reads. The overlap length is recorded as the edge attribute. Each node is labeled by the overlap information including read id, strand, and starting overlap position. For example, the positive strand Read1 with overlap position on 5250 is denoted as '1_0_5250'. It is worth noting that since a read with different strands and/or overlap positions is represented as different nodes in the network, the number of nodes in the network is actually larger than the number of reads. Nevertheless, the nodes with the same read id and strand but slightly different overlap positions, i.e., starting overlap positions that are apart less than 100bp, are considered as a single node in the network. For example, the two nodes '1_1_5250' and '1_1_5284' are actually merged into one node '1_1_5250' in the network. The overlap directions also have been encoded in the strand information of each network node, therefor it is not necessary to construct a directed network. As such, the complexity of network can be reduced and the community structures in network become clearer.

2.3 Community detection

The overlaps between the repeat reads are more intensive than that of the other reads, which in the constructed network is characterized with some topology structures of denser intra-connectivity and meanwhile sparse inter-connectivity. Such topology structures are also known as community in graph theory. Thus repeat identification can be transformed into a community identification problem when the network is constructed. For example, in Figure 5, two piles of reads after sequence alignment correspond to two repeats. In each pile, the coverage inside is much higher than outside, so there are corresponding community structures in the network of read overlaps thanks to the dense intra-connections in the read piles.

We define a vector $C = [c_1, \ldots, c_i, \ldots, c_{|V|}]$ to indicate the community labels of all nodes. The i-th component c_i of C means that the i-th node belongs to c_i-th community. Communities can be detected based on modularity optimization (Newman, 2006) for community labels in C. Modularity reflects the concentration of edges within communities compared with random distribution of links between all nodes regardless of communities. Particularly, the modularity $Q(C)$ with respect to a node label vector C is defined as follows (Blondel *et al.*, 2008):

$$Q(C) = \frac{1}{2m} \sum_{i,j=1}^{|V|} (A_{ij} - \frac{k_i k_j}{2m}) \delta(c_i, c_j), \tag{3}$$

where k_i and k_j are the degrees of the i-th and j-th nodes with their corresponding community labels c_i and c_j in the network, respectively, and A_{ij} is the element in the adjacency matrix of the network. The delta function is defined as $\delta(c_i, c_j) = 1$ if $c_i = c_j$, otherwise $\delta(c_i, c_j) = 0$. $Q(C)$, in the range of $[-1/2, 1)$, actually measures the strength of division of a network into communities. A larger $Q(C)$ indicates a better grouping of communities. To find the best community structures, one can maximize $Q(C)$ by the following constrained optimization problem:

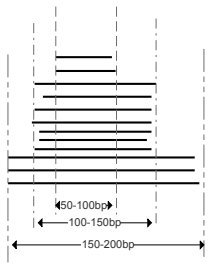
$$\begin{cases} \max_{C=[c_1,\ldots,c_{|V|}]} Q(C) \\ \text{subject to} : 1 \leq c_i \leq |V|, i = 1, \ldots, |V| \end{cases} \tag{4}$$

The Louvain method (Blondel *et al.*, 2008), which has shown promising performance in terms of accuracy and computing time (Yang *et al.*, 2016), is utilized to solve the modularity optimization model in (4) for community identification. The Louvain method initially assigns each network node as a single community. Afterward, the following two steps are iteratively proceeded until a maximum modularity is reached: 1) each node is removed from its own community and placed in the community of its neighbor node such that the modularity gain is maximized, and 2) nodes in the same community are aggregated to form a super-node and a new smaller scale network is generated.

2.4 Repeat reads extraction

Once the communities in the network are detected, the representative reads in each community are extracted to identify the corresponding repeat sequence. In particular, the overlap lengths associated the reads in one community are firstly divided into several continuous intervals. Afterward, the fold change (explained below) of each interval is calculated and the boundaries of the repeat sequence are detected in the interval with the smallest fold change. Finally, the consensus overlap within target intervals is extracted as the repeat sequence contained in the community.

The details of repeat reads extraction are illustrated using an example shown in Figure 6. Given the reads contained in a community, the numbers of reads with overlap lengths in different intervals, e.g., 50-100bp, 100-150bp, and 150-200bp, are calculated and recorded. Two, seven, and three reads fall within the three overlap length intervals, respectively. The fold change of the i-th interval is calculated as $\ln(p_i) - \ln(p_{i-1})$, where p_i is the number of reads of which the overlap lengths fall within the i-th interval. For example, in Figure 6, there are seven and two reads bearing 50-100bp and 100-150bp overlaps, respectively. Therefore, the fold change of interval 100-150bp is $\ln(7) - \ln(2) = 1.25$.



Overlap length interval	Number of reads	Fold change(log value)
50-100bp	2	
100-150bp	7	1.25
150-200bp	3	-0.84

Fig. 6. Reads in different overlap length intervals and the corresponding fold changes. The reads in a community are dispatched to different intervals according to their associated overlap lengths. The fold change of the i-th interval is calculated as $\ln(p_i) - \ln(p_{i-1})$, where p_i is the number of reads of which the overlap lengths fall within the i-th interval.

A positive fold change indicates an increase of read number from one overlap interval to the next interval. Conversely, a negative fold change suggests a decrease of read number. As observed in Figure 1, there are sharp coverage drops in the boundary of a repeat region, which can be detected by negative fold changes. As such, to determine the boundary of the repeat sequence contained in one community is equivalent to identify the interval with the smallest negative fold change. In this way, no predefined threshold, which could be bias and problem dependent, is required. In Figure 6, the smallest negative fold change -0.84 is with interval 150-200bp, so the boundary of the repeat sequence is most likely in the previous interval, i.e., 150-200bp. For safety, the repeat sequence is extracted in a shorter interval say 100-150bp. The reads with overlap length in 100-150bp interval are then traced back to extract the consensus overlap region as the target repeat sequence using faidx (Shirley *et al.*, 2015). Note that redundant repeat sequences could be extracted from different communities and they are merged to form a unique one in the final repeat library (the set of all identified repeat sequences).

3 Results

The proposed method is tested on Drosophila melanogaster and human PacBio long read sequencing data (Berlin *et al.*, 2015). The repetitiveness of the identified repeat sequences is verified based on the alignment results on reference genomes. The precision and coverage of the known repeat library are also evaluated. The PacBio raw sequences are sub-sampled to 800,000 (7.1GB) and 900,000 (4.4GB) sequences for Drosophila melanogaster and human data, respectively, to reduce the computational complexity. To the best of our knowledge, RepLong is the first *de novo* repeat identification method using long reads. Hence, for comparison study, RepLong is pitted against two short-read-based methods, i.e., RepARK and REPdenovo, and a genome-based method, i.e., RepeatModeler. RepARK and REPdenovo are tested using the same long read data as RepLong. The results of RepARK on short read data available in Koch *et al.* (2014) (distinguished as RepARK*) are also included in the comparison. For RepeatModeler, the Canu pipeline is first used to assemble the long reads, and then the assembled genome is input to the method. The details of all data used in this study can be found in the Table S1 of the Supplementary Materials. All experiments are carried out on a server with two Intel Xeon E5-2670 v3 CPUs of 12 cores and 256GB memory.

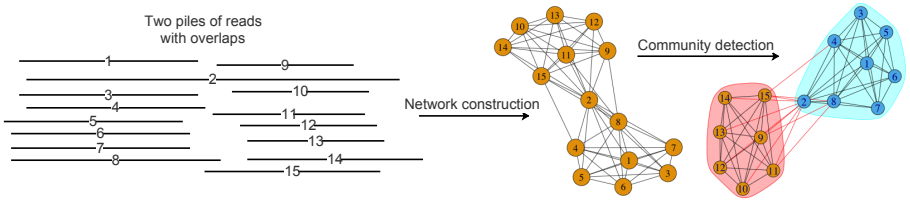


Fig. 5. The community detection process. The network is built by read overlaps and two communities are found in the network, which correspond to the two piles of reads.

Table 2. Alignment results on reference genome. The symbol * indicates the corresponding results are obtained by the short read input.

Method	RepLong	RepARK*	RepARK	REPdenovo
Num of identified sequences	1218	19677	1947	1029
Mapped	1211	19541	1943	1023
Unmapped	7	136	4	6
N50 (bp)	8527	87	640	4294
Library size	8.8MB	1.9MB	639KB	1.6MB
Repetitive	1123	18104	1932	1012

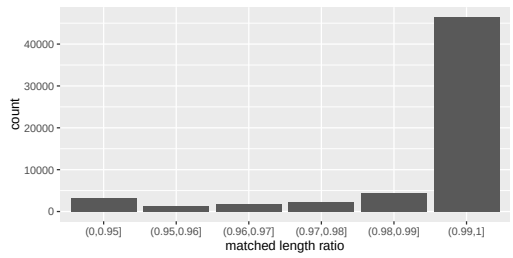


Fig. 8. Match length ratio distribution of the repeats identified by RepLong.

Fig. 7. Repeat frequency distribution of the repeats identified by RepLong.

Fig. 9. Indel length ratio distribution of the repeats identified by RepLong.

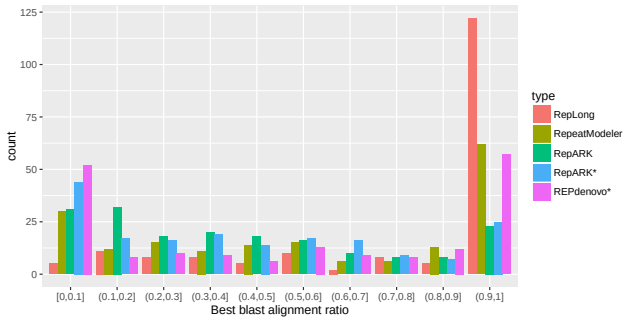


Fig. 10. BAR results of the compared methods.

3.1 Results on Drosophila melanogaster data

Drosophila melanogaster is one of the most extensively studied species in repeat identification and the repeats contained are relatively well known. In this regard, the proposed method is firstly evaluated on Drosophila melanogaster data. The identified repeat sequences are aligned to the *D. melanogaster* reference genome [1] using BWA (Li and Durbin, 2009) with an option "mem -x pacbio -a" that is specially designed to handle PacBio data with a sequencing error rate up to 20% and the results are shown in Table 2.

RepLong is shown to obtain much larger N50 value than RepARK and REPdenovo. In the 1218 read sequences identified by RepLong, 1211 of them can be mapped to the reference genome, and 1123 sequences are mapped multiple times. In these 1123 sequences, the repeat frequency is calculated and reported in Figure 7, where most sequences are shown to repeat 2-40 times. To verify if the remaining 95 sequences are repetitive, another reference genome assembled with PacBio long reads is used, and the 95 sequences are successfully aligned with BWA. Thirty-two sequences out of the 95 sequences can be identified as repeats, and the remaining 63 sequences are found to contain some repeat region using BLAST (Altschul *et al.*, 1990). In a word, the majority of the identified sequences by RepLong are truly repetitive.

For each repeat sequence identified by RepLong, both matched length ratio and indel length ratio are also calculated and the ratio distributions are depicted in Figures 8 and 9, respectively. It is shown that most sequences have a high matched ratio and low indel ratio. Most of the repeat sequences have a matched ratio over 99%. The sequences identified by RepLong contain only 2% indels, which is much lower than the reportedly average ∼15% indel ratio (English *et al.*, 2012) of PacBio data, i.e., RepLong is less affected by the high error rate of PacBio data.

The identified repeat sequences of RepLong and the compared methods are aligned to the RepBase library using BLAST to see the completeness of these results. Each sequence in RepBase can have multiple matches by the repeat sequences identified by the repeat identification methods. We define the "best alignment ratio (BAR)" of a RepBase sequence as

[1] http://humanparalogy.gs.washington.edu/dm3/dm3wgac.html

Table 3. BLAST results on RepBase library. The symbol * indicates the corresponding results are obtained with short read input.

Method	RepLong	RepARK*	RepARK	REPdenovo
Matched number	184	**199**	165	148
BAR≥10%	**179**	149	143	135
BAR≥50%	**147**	45	74	99
NAR≥10%	**136**	133	131	105
NAR≥50%	**83**	35	58	56

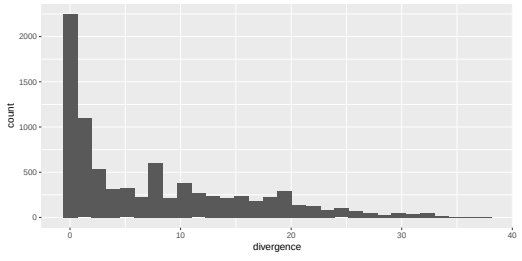


Fig. 11. Divergence of the repeats identified by RepLong.

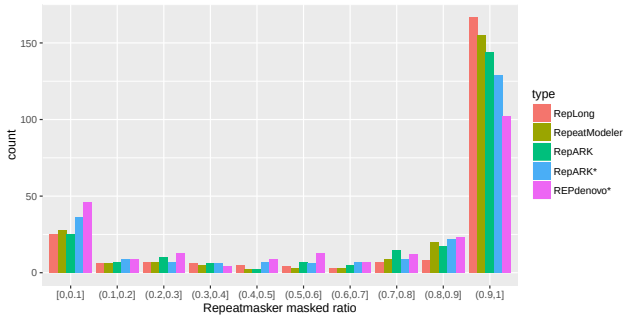


Fig. 12. Repbase sequence masked ratio comparison by RepLong and the compared methods.

Table 4. Repeat library masking result. The symbol * indicates the corresponding results are obtained with short read input.

Method	RepLong	RepARK*	RepARK	REPdenovo
Masked RepBase ratio	**86.94%**	85.61%	79.11%	72%
Masked genome ratio	**28.03%**	26.10%	27.71%	26.79%
2nd run masked genome ratio	**3.20%**	5.33%	3.1%	4.07%

the largest percentage of this RepBase sequence matched by a single identified sequence, and the "nearest alignment ratio (NAR)" of a RepBase sequence as the percentage of this RepBase sequence matched by a single identified sequence that has the closest length to this RepBase sequence. BAR is similar to the statistics C_m used in Chu *et al.* (2016). The distributions of the BAR values obtained by RepLong and the compared methods are shown in Figure 10, and the NAR distributions are shown in Figure S1 of the Supplementary Materials. It is observed that most identified repeat sequences of RepLong cover over 90% of the corresponding sequences in RepBase library and the identified repeated sequences of RepLong individually aligns the RepBase sequences much longer than that of the compared methods. The numbers of matched sequences, i.e. BAR≥10%, BAR≥50%, NAR≥10%, and NAR≥50% of RepLong, RepARK and REPdenovo are listed in Table 3. RepARK has more matched sequences than the other methods thanks to the considerably larger identified repeat library, i.e., 19677 vs. 1218 as shown in Table 2. Whereas the other four metrics indicate the identified repeat sequences of RepLong individually aligned the RepBase sequences longer than that of RepARK and REPdenovo. The detailed alignment results of RepLong are provided in Tables S2-S3 of the Supplementary Materials.

The coverages of the identified repeat libraries of RepLong, RepARK, and REPdenovo against the RepBase library and *D. melanogaster* reference genome are also evaluated using RepeatMasker (http://www.repeatmasker.org/). The results are presented in Table 4 and Figure 12. RepLong is observed to mask more percentages of the RepBase library and the reference genome. Particularly, the RepLong library masks 86.94% of the RepBase library and 28.03% of the *D. melanogaster* reference genome, i.e., both are better than that of RepARK and REPdenovo. The unmasked parts of the reference genome are subsequently masked by the RepBase library in the second run to see how much repeat segments are left after the first masking. In the second run masking, the lower masked ratio of a method indicates the fewer repeats are missing in the first masking, i.e., the better coverage is obtained by the method in the first masking. RepLong also shows superiority to RepARK and REPdenovo in the second run masking. To estimate the repeat divergence of RepLong, we use RepBase library to mask the repeat library identified by RepLong using RepeatMasker. As the results shown in Figure 11, most identified repeats have low divergence compared to RepBase. Detailed RepeatMasker results are also provided in Table S4 of the Supplementary Materials.

Table 6. Time and memory usage of the methods on the Drosophila melanogaster data.

Method	RepLong	RepeatModeler	RepARK	REPdenovo
Running time (m)	1030	2538	23	58
Memory (GB)	29	42	12	32

With long reads input, the common way to find repeat sequences is to perform whole genome *de novo* assembly and then detect repeats from the assembled genome, which is exactly the method done in RepeatModeler. The main difference between RepLong and RepeatModeler is RepLong does not involve the reads assembly but only the pair-wise read overlap identification. One major impact factor to assembly performance and overlap identification could be the input data coverage. To investigate the effects of the data coverage to the performance of both RepLong and RepeatModeler, we sampled a number of test data sets from the corrected Pacbio data with coverage ranging from 4.7X (781MB) to 44X (7.1GB). The results of RepLong and RepeatModeler on these test data sets are reported in Table 5. Generally, the performance of RepLong is comparable to that of RepeatModeler, yet RepLong is much more computationally efficient as shown in Table 6, thanks to the exemption of more time-consuming reads assembly.

The time and memory usages of all methods on the Drosophila melanogaster data are reported in the Table 6. It is shown that RepLong and RepeatModeler consume more running time than the short-read-based methods, i.e., RepARK and REPdenovo due to the involvement of the Canu pipeline either for pair-wise overlap identification or genome assembly. In terms of memory consumption, RepeatModeler and RepARK use the largest and smallest spaces, respectively.

3.2 Results on human data

To test the performance of RepLong on more complicate genome, the PacBio human long read data is used. The identified repeat sequences are

Table 5. Comparison of RepLong and RepeatModeler on a number of input with different coverages.

Coverage	4.73X (781MB)		9.92X (1.6GB)		14.2X (3.2GB)		44X (7.1GB)	
Method	Replong	RepeatModeler	Replong	RepeatModeler	Replong	RepeatModeler	Replong	RepeatModeler
Num of identified sequences	521	754	688	785	1139	793	1218	758
Mapped	521	715	688	785	1139	760	1211	724
N50 (bp)	6777	2370	7800	2564	8286	3010	8527	3024
Library size	2.6MB	958KB	3.9MB	1.1MB	6.5MB	1.2MB	8.8MB	1.1MB
Repetitive	418	693	523	710	804	733	1123	689
RepBase BLAST matched number	159	158	178	171	189	171	184	171
BAR$\geq$10%	157	151	173	163	186	166	179	162
BAR$\geq$50%	123	88	138	100	152	104	147	105
NAR$\geq$10%	123	125	131	125	144	137	136	132
NAR$\geq$50%	70	61	80	69	79	79	83	67
Masked RepBase ratio	82.97%	82.47%	86.36%	86.17%	90.18%	86.4%	86.94%	87.43%
Masked genome ratio	27.99%	28.18%	25.81%	28.92%	27.05%	28.93%	28.03%	30%
2nd run masked genome ratio	2.71%	2.77%	5.67%	2.25	4.21%	2.21%	3.20%	1.18%

Table 7. Results on human data. The symbol * indicates the corresponding results are obtained with short read input.

Method	RepLong	RepARK*	REPdenovo
Num of identified sequences	5799	62425	10864
Mapped	5799	57613	11017
Unmapped	0	4812	258
Repetitive	1377	53727	10864
Indel Bases Ratio	2.97%	0.3%	2.82%
N50 (bp)	13200	143	2366
Library size	57MB	9.8MB	15MB
RepBase BLAST matched number	536	422	233
BAR$>$10%	507	311	91
BAR$\geq$50%	355	39	186
NAR$\geq$10%	465	260	73
NAR$\geq$50%	213	30	172
Masked RepBase sequences	81.05%	62.70%	26.46%
Masked genome ratio	34.98%	28.77%	18.88%
2nd run masked genome ratio	7.44%	11.9%	20.28%
Running time (m)	7765	N.A.	30
Memory (GB)	57	N.A.	64

mapped to human hg19 reference genome[2] using BWA with arguments "mem -x pacbio -a". The human repeat library is much larger than that of Drosophila melanogaster and thus calls for much longer computational time. To fast the repeat identification of human data, we consider a much lower coverage say 1.5X for human data, which can also serve as a more difficult problem to test the robustness of the compared methods. Canu fails to assemble the genome for RepeatModeler with such low coverage input and RepARK cannot work on this data either as it requires at least 10X coverage (Koch *et al.*, 2014). The remaining methods, namely, RepLong, RepARK* (with short reads input) and REPdenovo, are tested on the data and the results are shown in Table 7. Consistent to the observation on Drosophila melanogaster data, RepLong is shown to successfully cover more RepBase library and mask more bases on the reference genome than the other two methods. The superiority of RepLong on low coverage data is more obvious.

[2]http://hgdownload.soe.ucsc.edu/goldenPath/hg19/bigZips/

Table 8. RepLong result comparison using different input and process. RepLong† means the RepLong pipeline without the correction step.

Input	Raw	Raw	Polished	Polished
Method	RepLong	RepLong†	RepLong	RepLong†
Num of identified sequences	1218	596	884	966
Mapped	1211	592	883	965
Unmapped	7	4	1	1
Repetitive	1123	568	626	688
Matched Bases Ratio	96.73%	91.44%	97.52%	97.83%
Indel Bases Ratio	2.05%	10.14%	1.21%	1.61%

3.3 Effects of error correction

In the current stage, long read data tends to have higher error rate than short read data. To investigate the effects of error correction to the performance of RepLong, we use both raw and MHAP polished PacBio Drosophila reads with or without the post processing in FALCON correction. Table 8 shows the comparison results, where RepLong†is used to indicate the results without using the FALCON correction. Since both RepLong with raw input and RepLong†with polished input are better than RepLong†with raw input, error correction in preprocessing or postprocessing does help to improve the repeat identification accuracy. The fact that RepLong and RepLong†with polished input are comparable suggests double error corrections does not necessarily improve the identification efficiency. To guarantee the maximal compatibility, RepLong is configured to take raw PacBio reads as input together with the FALCON correction.

4 Discussion and Conclusion

In this paper, we propose a *de novo* repeat identification method namely RepLong based on PacBio long reads. RepLong firstly applies the MinHash algorithm to find the overlap between each pair of the reads and then a network of read overlaps is built to find the community structures. The representative reads in each community are extracted to identify repeat sequences and form the repeat library. The Drosophila melanogaster and human data are used to test the proposed method. RepLong is shown to identify longer repeats than the short-read-based methods RepARK and REPdenovo, and obtain comparable performance to genome-based method RepeatModeler while with much less running time. Moreover, RepLong can handle lower coverage data than the compared methods. The

superiorities of RepLong to the other representative methods are analyzed and summarized as follows:

Replong vs. short-read-based methods: The majority of short-read-based methods follow the k-mer counting and assembly framework, which is very computationally efficient but requires a high-coverage and low-error-rate input data. When applied to long read data, the short-read-based methods cannot take full advantage of the data and tend to break the long repeats apart and assemble the pieces back, which usually deteriorate the identification of long repeats. RepLong is designed based on overlap identification that intrinsically suits more for long repeats identification and less affected by the data coverage.

Replong vs. genome-based methods: The genome-based methods work relying on a high quality assembled genome that is also subject to the coverage of the input data. Moreover, the assembling of genome usually consumes huge time and space resources. RepLong using the overlap identification rather than the assembly process is more computationally efficient.

Replong vs. assembling, mapping, and coverage analysis methods: Following the observations in Figure 1, another way of long read based repeat identification could be implemented by a) assembling the error-corrected reads, b) mapping the reads onto the assembly, and c) identifying highly covered regions and their boundaries to get the repeats. As shown in Figures S2-S3 of the Supplementary Materials, the alternative method theoretically can obtain similar results to RepLong, however it faces the same problems as genome-based methods for involving the assembly process.

RepLong is expected to serve as a complementary solution to the existing methods to promote the repeat identification performance on long read sequencing data. RepLong might also be extended to detect repeat copy number gain with more specific design as investigated in Figure S4 of the Supplementary Materials. In the future work, inferring method will be introduced to overlap identification to speed up RepLong by avoiding the exhaustive read alignment, and the input data could be further sub-sampled to quest after a trade-off of accuracy and speed. A concise library with little loss in completeness will also be in pursuit.

Funding

This work was supported by National Natural Science Foundation of China [61471246, 61602309, 61575125], Guangdong Special Support Program of Top-notch Young Professionals [2014TQ01X273, 2015TQ01R453], Guangdong Foundation of Outstanding Young Teachers in Higher Education Institutions [Yq2015141], Shenzhen Fundamental Research Program [JCYJ20160307113632699, JCYJ20150324141711587, and JCYJ20170302154328155], and Natural Science Foundation of SZU [2017077].

References

Altschul, S. F., Gish, W., Miller, W., Myers, E. W., and Lipman, D. J. (1990). Basic local alignment search tool. *Journal of molecular biology*, **215**(3), 403–410.

Bao, Z. and Eddy, S. R. (2002). Automated de novo identification of repeat sequence families in sequenced genomes. *Genome research*, **12**(8), 1269–1276.

Bennetzen, J. L. and Wang, H. (2014). The contributions of transposable elements to the structure, function, and evolution of plant genomes. *Annual review of plant biology*, **65**, 505–530.

Bergman, C. M. and Quesneville, H. (2007). Discovering and detecting transposable elements in genome sequences. *Briefings in Bioinformatics*, **8**(6), 382–392.

Bergman, C. M., Quesneville, H., Anxolabéhère, D., and Ashburner, M. (2006). Recurrent insertion and duplication generate networks of transposable element sequences in the Drosophila melanogaster genome. *Genome biology*, **7**(11), 1.

Berlin, K., Koren, S., Chin, C.-S., Drake, J. P., Landolin, J. M., and Phillippy, A. M. (2015). Assembling large genomes with single-molecule sequencing and locality-sensitive hashing. *Nature Biotechnology*, **33**(6), 623–630.

Blondel, V. D., Guillaume, J.-L., Lambiotte, R., and Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, **2008**(10), P10008.

Chin, C.-S., Peluso, P., Sedlazeck, F. J., Nattestad, M., Concepcion, G. T., Clum, A., Dunn, C., OâŁ™Malley, R., Figueroa-Balderas, R., Morales-Cruz, A., *et al.* (2016). Phased diploid genome assembly with single molecule real-time sequencing. *Nature methods*, **13**(12), 1050.

Chu, C., Nielsen, R., and Wu, Y. (2016). REPdenovo: Inferring De Novo Repeat Motifs from Short Sequence Reads. *PLOS ONE*, **11**(3), e0150719.

de Koning, A. J., Gu, W., Castoe, T. A., Batzer, M. A., and Pollock, D. D. (2011). Repetitive elements may comprise over two-thirds of the human genome. *PLoS Genet*, **7**(12), e1002384.

Eddy, S. R. (2012). The C-value paradox, junk DNA and ENCODE. *Current biology*, **22**(21), R898–R899.

Edgar, R. C. and Myers, E. W. (2005). PILER: Identification and classification of genomic repeats. *Bioinformatics*, **21**(Suppl 1), i152–i158.

Eid, J., Fehr, A., Gray, J., Luong, K., Lyle, J., Otto, G., Peluso, P., Rank, D., Baybayan, P., Bettman, B., and others (2009). Real-time DNA sequencing from single polymerase molecules. *Science*, **323**(5910), 133–138.

English, A. C., Richards, S., Han, Y., Wang, M., Vee, V., Qu, J., Qin, X., Muzny, D. M., Reid, J. G., Worley, K. C., and Gibbs, R. A. (2012). Mind the Gap: Upgrading Genomes with Pacific Biosciences RS Long-Read Sequencing Technology. *PLoS ONE*, **7**(11), e47768.

Fertin, G., Jean, G., Radulescu, A., and Rusu, I. (2015). Hybrid de novo tandem repeat detection using short and long reads. *BMC medical genomics*, **8**(3), S5.

Feschotte, C. and Pritham, E. J. (2007). Computational analysis and paleogenomics of interspersed repeats in eukaryotes. *Computational genomics: current methods*, pages 31–53.

Fortunato, S. (2010). Community detection in graphs. *Physics Reports*, **486**(3-5), 75–174.

Girvan, M. and Newman, M. E. (2002). Community structure in social and biological networks. *Proceedings of the national academy of sciences*, **99**(12), 7821–7826.

Harenberg, S., Bello, G., Gjeltema, L., Ranshous, S., Harlalka, J., Seay, R., Padmanabhan, K., and Samatova, N. (2014). Community detection in large-scale networks: A survey and empirical evaluation: Community detection in large-scale networks. *Wiley Interdisciplinary Reviews: Computational Statistics*, **6**(6), 426–439.

Jurka, J., Kapitonov, V., Pavlicek, A., Klonowski, P., Kohany, O., and Walichiewicz, J. (2005). Repbase Update, a database of eukaryotic repetitive elements. *Cytogenetic and Genome Research*, **110**(1-4), 462–467.

Kelley, D. R., Hendrickson, D. G., Tenen, D., and Rinn, J. L. (2014). Transposable elements modulate human RNA abundance and splicing via specific RNA-protein interactions. *Genome biology*, **15**(12), 537.

Koch, P., Platzer, M., and Downie, B. R. (2014). RepARK–de novo creation of repeat libraries from whole-genome NGS reads. *Nucleic Acids Research*, **42**(9), e80–e80.

Koren, S., Walenz, B. P., Berlin, K., Miller, J. R., and Phillippy, A. M. (2016). Canu: Scalable and accurate long-read assembly via

adaptive k-mer weighting and repeat separation. Technical Report biorxiv;071282v1.

Lander, E. S., Linton, L. M., Birren, B., Nusbaum, C., Zody, M. C., Baldwin, J., Devon, K., Dewar, K., Doyle, M., FitzHugh, W., and others (2001). Initial sequencing and analysis of the human genome. *Nature*, **409**(6822), 860–921.

Li, H. and Durbin, R. (2009). Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics*, **25**(14), 1754–1760.

Li, R., Ye, J., Li, S., Wang, J., Han, Y., Ye, C., Wang, J., Yang, H., Yu, J., Wong, G. K.-S., and Wang, J. (2005). ReAS: Recovery of Ancestral Sequences for Transposable Elements from the Unassembled Reads of a Whole Genome Shotgun. *PLoS Computational Biology*, **1**(4), e43.

Newman, M. E. (2006). Modularity and community structure in networks. *Proceedings of the national academy of sciences*, **103**(23), 8577–8582.

Price, A. L., Jones, N. C., and Pevzner, P. A. (2005). De novo identification of repeat families in large genomes. *Bioinformatics*, **21**(Suppl 1), i351–i358.

Schaeffer, S. E. (2007). Graph clustering. *Computer Science Review*, **1**(1), 27–64.

Schlötterer, C. (2000). Evolutionary dynamics of microsatellite DNA. *Chromosoma*, **109**(6), 365–371.

Schnable, P. S., Ware, D., Fulton, R. S., Stein, J. C., Wei, F., Pasternak, S., Liang, C., Zhang, J., Fulton, L., Graves, T. A., and others (2009). The B73 maize genome: Complexity, diversity, and dynamics. *science*, **326**(5956), 1112–1115.

Shirley, M. D., Ma, Z., Pedersen, B. S., and Wheelan, S. J. (2015). Efficient" pythonic" access to FASTA files using pyfaidx. Technical report, PeerJ PrePrints.

Treangen, T. J. and Salzberg, S. L. (2011). Repetitive DNA and next-generation sequencing: Computational challenges and solutions. *Nature Reviews Genetics*.

Yang, Z., Algesheimer, R., and Tessone, C. J. (2016). A Comparative Analysis of Community Detection Algorithms on Artificial Networks. *Scientific Reports*, **6**, 30750.

Zytnicki, M., Akhunov, E., and Quesneville, H. (2014). Tedna: A transposable element de novo assembler. *Bioinformatics*, page btu365.