Genetics and Molecular Research

A repetitive sequence assembler based on next-generation sequencing

S. Lian[1], Y. Tu[1], Y. Wang[1], X. Chen[1] and L. Wang[2]

[1]School of Physics and Electronic Engineering, Xinyang Normal University, Xinyang City, China
[2]School of Life Science, Xinyang Normal University, Xinyang City, China

Corresponding author: L. Wang
E-mail: wangleibio@126.com

Genet. Mol. Res. 15 (3): gmr.15038790
Received May 16, 2016
Accepted June 3, 2016
Published July 25, 2016
DOI http://dx.doi.org/10.4238/gmr.15038790

ABSTRACT. Repetitive sequences of variable length are common in almost all eukaryotic genomes, and most of them are presumed to have important biomedical functions and can cause genomic instability. Next-generation sequencing (NGS) technologies provide the possibility of identifying capturing these repetitive sequences directly from the NGS data. In this study, we assessed the performances in identifying capturing repeats of leading assemblers, such as Velvet, SOAPdenovo, SGA, MSR-CA, Bambus2, ALLPATHS-LG, and AByss using three real NGS datasets. Our results indicated that most of them performed poorly in capturing the repeats. Consequently, we proposed a repetitive sequence assembler, named NGSReper, for capturing repeats from NGS data. Simulated datasets were used to validate the feasibility of NGSReper. The results indicate that the completeness of capturing repeat is up to 99%. Cross validation was performed in three real NGS datasets, and extensive comparisons indicate that NGSReper performed best in terms of completeness and accuracy in capturing repeats. In

conclusion, NGSReper is an appropriate and suitable tool for capturing repeats directly from NGS data.

Key words: Next-generation sequencing; Interspersed repeats; Tandem repeats; Repetitive genome assembly

INTRODUCTION

The genomes of all eukaryotes contain repetitive elements of variable length that can occupy a significant fraction of the total DNA content (SanMiguel et al., 1999), e.g. ~20% of *Caenorhabditis elegans* and *C. briggsae* genomes (Stein et al., 2003) and ~50% of the human genome (Lander et al., 2001). Furthermore, repeats play important roles in genome evolution by causing mutations and rearrangements (Bowen and Jordan, 2002; Ma et al., 2004) that might lead to altered gene functions (Buard and Jeffreys, 1997). Moreover, molecular evidence suggests that some repetitive elements may lead to new genes (Morgante et al., 2005). Thus, the study of repetitive sequences provides a comprehensive understanding of both gene and genome functions in eukaryotes. In addition, the repeats challenge the genome algorithms (Treangen and Salzberg, 2012). Therefore, repeat identification is of considerable importance and a critical part of the analysis of newly sequenced species. Fortunately, next generation sequencing (NGS) technologies provide the possibility of capturing repetitive elements from sequence reads, if the corresponding reference genome is not available.

NGS technologies are characterized by shorter reads, higher throughput, parallel operation and lower cost (Pop, 2009). Currently, the commercially available NGS platforms (Metzker, 2010; Loman et al., 2012) include 454 from Roche (400 to 600 bp); GA, MiSeq, and HiSeq from Illumina (100 to 150 bp); SOLiD (typically 75 bp) (Miller et al., 2012) and Ion Torrent from Life Technologies (~200 bp); RS system from Pacific Bioscience; and Heliscope from Helicos Biosciences (Harris et al., 2008). To date there are more than tens of genome assemblers based on NGS data, being Velvet (Zerbino and Birney, 2008), ABySS (Simpson et al., 2009), SOAPdenovo (Li et al., 2010), SGA (Simpson and Durbin, 2012), MSR-CA (http://www.genome.umd.edu/masurca.html), Bambus2 (Koren et al., 2011), ALLPATHS-LG (Gnerre et al., 2011) and SWA (Lian et al., 2014) the leading ones. Ideally, a good genome assembly algorithm not only can achieve long contigs, but also can accurately capture repetitive elements directly from NGS reads. Their performance has been assessed comprehensively including contiguity, consistency, and accuracy of the assembled genomes as well as hardware and software requirements (Salzberg et al., 2012), except for the ability of capturing repeats.

Herein, we first evaluated the performance of NGS technologies in capturing repeats, as a good genome assembler not only should assemble genomes but also capture repeats completely. Unfortunately, our results indicate that the performance of the currently available assemblers in capturing repeats is very poor. Thus, a repetitive sequence assembler form NGS reads, named NGSReper, was proposed. The feasibility of NGSReper was validated by simulated datasets, and the corresponding results indicated that the completeness and accuracy in capturing repeats were nearly 98 and 100% respectively. Finally, real NGS datasets were used for cross validation. Extensive comparisons with other seven assemblers, such as Velvet, SOAPdenovo, SGA, MSR-CA, Bambus2, ALLPATHS-LG and AByss were conducted in three Illumina-generated datasets presented in GAGE (Salzberg et al., 2012). Results indicated that NGSReper performed best in terms of completeness and accuracy in capturing repeats.

The authors provide the free executable software for non-commercial use by request.

MATERIAL AND METHODS

The datasets used to evaluate the performances of NGSReper are described herein. Three kinds of datasets were used: simulated, reference, and real NGS datasets. Simulated and reference datasets are used to validate the feasibility of NGSReper, real NGS datasets are used to cross validation. For simulated datasets, three genomes containing different types of repeats, such as interspersed, tandem, and compound repeats were generated. The detailed steps were as follows. Firstly, three 500 kb sequences, namely sequence A, sequence B, and sequence C, were generated randomly from a finite alphabet {A,T,C,G} respectively. Secondly, different types of repeats, with a wide range of lengths and copies, were inserted into the corresponding sequences. Table 1 shows the detailed information of size and copies with different kinds of repeats. Thirdly, NGS data were randomly sampled from three sequences for different coverage and read length. For reference genome datasets, we downloaded the references genomes of *S. cerevisiae* and *C. elegans* from UCSC (http://hgdownload.soe.ucsc.edu/downloads.html) and *E. coli* k12 from GenBank (U00096.3).

For *S. cerevisiae* and *C. elegans*, the chromosomes were randomly selected, chromosome IV for *S. cerevisiae* (chrIV-S.c) and chromosome III for *C. elegans* (chrIII-C.e), respectively. The chromosome sizes of chrIV-S.c, *E. coli*, and chrIII-C.e were 1,531,933, 5,132,068, and 13,783,700 bp respectively. The NGS reads were randomly sampled from these references. By whole genome scan using HashRepeatFinder (Lian et al., 2016), we detected 41, 56, and 339 repeats longer than 200 bp in chrIV-S.c, *E. coli*, and chrIII-C.e, respectively.

Table 1. The detailed information of repeats in simulated datasets.

Sequence	Repeats	Length (bp)	Copies
Sequence A	Interspersed repeats	500	8
		1000	10
		2000	8
		3000	6
		4000	5
		5000	3
Sequence B	Tandem repeats	1000 and 5000	5
		2000 and 4000	4
		3000 and 3000	6
Sequence C	Compound repeats	1000	5
		2000	4
		3000	3
		2000 and 3000	5
		1500 and 2500	6
		2500 and 2500	3

For real datasets, we use two bacterial genomes (*S. aureus* and *R. sphaeroides*) and human chromosome 14, which were downloaded from http://gage.cbcb.umd.edu/data. In the GAGE study, all reads were error-corrected before their assembly by these assemblers. For an appropriate comparison, we obtained these corrected datasets for GAGE use. As the two bacterial genomes and human chromosome 14 had been captured using Sanger data, each of them could be used as a reference assembly. For these three species, their repetitive structures, including repeats and copies, were detected by whole genome scan. For S. aureus,

R. sphaeroides and human chromosome 14, 54, 21, and 2635 repeats longer than 100 bp were detected respectively. Their total sizes accounted for 15.8, 3.6, and 381.5 kb, respectively.

RESULTS

Overview

The principle of NGSReper is based on the combinational strategy of dynamic overlapping and smoothing filtering. The estimation of read counts for a single point is affected by the sequencing bias, whereas the estimation in a continuous interval will be more scientific than in a single point. Moreover, based on the overlapping interval, the filtering function can be used to remove the sequencing bias and increase the confidence of estimating reads counts. Accordingly, the strategy of dynamic overlapping can be applied to search the best seed for extension. The concrete steps and process are detailed as follows. NGSReper runs in five key steps (Figure 1): pre-processing, unique processing, repetitive seed selection, index construction, and repeat capture.

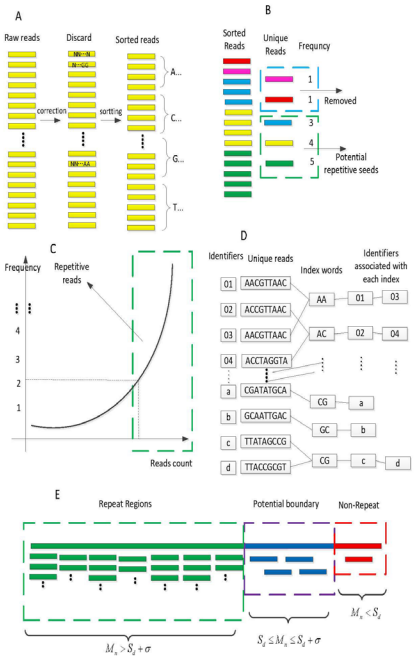

Figure 1. Graphic illustration of key steps of NGSRepeatFinder. **A.** Pre-process. This step contains data cleaning and read sorting. **B.** Unique Processing. The five different color lines represent five unique reads in sorted reads. Each of them appears with different frequencies. By unique processing, the identical reads are collapsed into one single read with its corresponding frequency. **C.** Seed selection. The unique reads are ranked by frequency (from low to high). The reads with frequency larger than sequencing depth are selected as the seeds for repeats (the green dotted frame). **D.** Hash index. Unique reads with associated identifiers are shown on the left, the index words are chosen using two letters from AA to TT, and the hash index is used to transform the comparisons between strings to numerical mapping between identifiers. **E.** Repeat capturing. The repetitive seeds are extended using a greedy graph and dynamic overlapping strategy. M_n is the mean read counts, S_d is the sequencing depth, σ is the tuning parameter, $0 < \sigma < S_d$. For example, if $S_d = 2$, $\sigma = 1$, the region with $M_n > 3$ will be considered as a repetitive sequence.

1) Pre-processing (Figure 1A). The real NGS reads contain large sequencing errors, which affect the assembled results. Consequently, the reads with low quality value or containing any 'N' are removed first, and then the cleaned reads are sorted by dictionary order. This sorting strategy is specially designed for computing frequency and unique processing.

2) Unique processing (Figure 1B). Identical reads are collapsed into one single read and its corresponding frequency is recorded. After unique processing, the reads with frequencies higher than the sampling depth might be potential repeats, which will be considered as the seed of repeats for extension in the following steps. By unique processing, the amount of data decreases significantly, in particular for densely sampling data.

3) Seed selection (Figure 1C). In NGSReper, each sequence requires a unique read, called a seed, to initiate the extension. Ideally, a good seed should be a unique read without sequencing error. Thus, selecting a set of good seeds is essential for capturing repeats by NGSReper. The current version uses read counts and base quality as the criteria to select seeds. After unique processing, those reads with frequencies higher than the threshold are selected as the potential seeds. Among these potential seeds, the one with highest quality value is used to initiate the extension process. This strategy tends to avoid choosing seeds with errors to maximum extent.

4) Constructing hash index (Figure 1D). In order to improve computing speed, an indirect hash structure was designed and adopted in this part. Firstly, the index keywords are directly transformed into quaternary integers instead of the string. Secondly, the identifiers of the unique reads are recorded in decimal list. Thirdly, constructing the mapping relations between unique reads and decimal list. This index structure adopts integer arithmetic instead of string operations, and the computational complexity is significantly reduced. Consequently, this structure is appropriate for DNA sequencing reads, especially for large datasets.

5) Repeat capture (Figure 1E). After seed selection, the repeat extension process is performed by combining the strategy of greedy overlapped graph (Dohm et al., 2007) and dynamic overlapping assembly (Lian et al., 2014). Based on the greedy overlapped graph, dynamic overlapping interval and sliding window function were applied to capture repeats. The mean value of read counts in dynamic interval M_n was used as the threshold to detect the boundary of potential repeats. If S_d is the sequencing depth, σ is the tuning parameter, $0 < \sigma < S_d$, the region with $M_n > S_d + \sigma$ will be considered as a repetitive sequence.

Metrics

To evaluate the performance in capturing repeats, widely recognized metrics were used. These metrics (Lian et al., 2014) include Family, accuracy of family (F-acc), accuracy of captured repeats (R-acc), total size of captured repeats (T-size), N50 accuracy (N50-acc), and Max repeats accuracy (Max-acc).

Family: a group of repetitive sequences inferred as having a common ancestor based on sequence similarity. Note that in the context of this study, the similarity was set to 95%. This metric is used to evaluate the completeness of types of detected repeats. Generally, larger family indicated that more types of repeats were detected. Therefore, the method has more completeness.

Accuracy of family (F-acc) is used to evaluate the completeness of the families of captured repeats and defined as:

$$\text{F-acc} = \frac{F_d}{F_r}$$

where F_d represents the detected families of repeats in NGS data, while the F_r represents the real families in reference genomes. For example, if the reference genome contains four repeats: repeat A, B, C, and D, but the NGSReper capture only repeat A, B, and C from NGS data, thus, $F_d = 3$, $F_r = 4$, the corresponding family accuracy will be F-acc = 75%. Notably, the similarity was set to 95%.

Accuracy of captured repeats (R-acc) is used to evaluate the correctness of all types of captured repeats and is defined as:

$$\text{R-acc} = 1 - \frac{\sum |R_c - R_r|}{\sum R_r}$$

where R_r represents the length of the real repeat and R_c the length of captured repeat. The higher the R-acc indicated the better performance of capturing repeats.

Total size (T-size), or the total size of detected repeats, is used to evaluate the completeness of whole size of the detected repeats, and which is defined as:

$$\text{T-size} = \sum_{i=1}^{N} l_i \times c_i$$

where l_i represents the *ith* family of repeat, C_i the corresponding copies and N the number of family. For example, if a reference genome contains four families of repeats: repeat A with 50 copies and 500 bp, repeats B with 100 copies and 1000 bp, repeats C with 30 copies and 1200 bp, repeats D with 40 copies and 1500 bp, therefore, family = 4, T-size = 50 x 500 + 100 x 1000 + 30 x 1200 + 40 x 1500 = 221 kb.

N50 accuracy (N50-acc) is the accuracy of N50 between captured and real repeats, which is used to evaluate the total similarity between captured and real repeats and is defined as:

$$\text{N50-acc} = 1 - \frac{|R_{n50} - C_{n50}|}{R_{n50}}$$

where R_{n50} represents the N50 of real repeats and C_{n50} the N50 of captured repeats.

Max accuracy (Max-acc) is the accuracy of maximum captured repetitive contig with maximum real repeats and is defined as:

$$\text{Max-acc} = 1 - \frac{|R_{\max} - C_{\max}|}{R_{\max}}$$

where R_{max} represents the maximum size of real repeats and C_{max} the maximum size of captured repeats.

To evaluate the accuracy, the metric F-acc and R-acc were computed by aligning back to the reference genome using MATLAB platform, and the default similarity was set to 90%.

In the process of evaluating the captured repeats, we first analyzed the completeness and then the accuracy of the assembled repeats. Among these metrics, Family, T-size, N50-acc, and Max-acc were specially designed to evaluate the completeness in capturing repeats, while R-acc and F-acc were designed to evaluate the accuracy of the captured repeats.

Simulation studies

In this part, we validated the feasibility of NGSReper with two kinds of datasets, simulated and reference data, respectively. Table 2 presented the detailed results.

Table 2. Statistics of captured repeats in a simulated study.

Data sets		Family	F-acc (%)	R-acc (%)	T-size (.kb)	N50-acc (%)	Max-acc (%)
Simulated data	Sequence A	6	100	96	15,562	100	100
	Sequence B	5	100	98.7	10,641	100	100
	Sequence C	11	100	98.7	26,352	100	100
Reference genome	chrIV-S.c	32	82.9	89.4	15,764	95	100
	E.coli	50	98.2	99	39,263	100	100
	chrIII-C.e	476	82	84.3	150,525	92	100

Parameters: $S_d = 2$, $L_r = 50$, $L_w = 3$, and kmer = 9. Contigs smaller than 200 were removed.

Table 2 shows the detailed feasibility of NGSReper in simulated and reference genome data. For simulated data, three sequences containing tandem, interspersed, and compound repeats with different lengths and copy numbers (see Material and Methods) were used to generate the NGS reads. For reference genome data, the real reference genomes of three species were used to generate the NGS reads, and the corresponding repeats were detected with HashRepeatFinder (Lian et al., 2016). The results presented in Table 2 indicate that NGSReper performed well in capturing all kinds of repeats from NGS reads. Specifically, (1) from simulated data, there were 6, 5, and 11 families of repeats, which were set in sequences A, B, and C respectively, in advance. Therefore, all F-acc were up to 100%, which shows that the completeness of NGSReper correlates perfectly. For the R-acc metric, the corresponding sequences were up to 96, 98.7, and 98.7% respectively, and the maximum error rate of R-acc was less than 3%, which indicates that the accuracy of captured repeats is very high. Furthermore, all N50-acc and Max-acc were up to 100%. (2) From reference genome data, the real families of repeats were 34, 55, and 275, and the detection items of NGSReper were 32, 50, and 476 respectively. Therefore, the corresponding F-acc were 82.9, 98.2, and 82% respectively. By analyzing the similarity, R-acc were 89.4, 99, and 84.3% respectively. Thus, the completeness in type and size of the captured repeats are still good. Meanwhile, the N50-acc was 95, 100, 92% respectively, which indicates that there is a small difference between the mean size of captured repeats and real repeats. However, the Max-acc for families was 100%, which indicates that NGSReper performed well in terms of capturing large repeats. In conclusion, NGSReper is suitable for capturing repeats from NGS data with both completeness and accuracy.

Comparisons in real NGS datasets

In order to cross validate the performances in capturing repeats, an extensive comparison with other leading assemblers using real NGS reads was performed. The real

NGS data and leading assemblers are descripted in GAGE (Salzberg et al., 2012). The detailed results are presented in Table 3. Moreover, in order to display the comparisons more clearly, metrics Family, F-acc, and Max-acc were plotted, which provides additional information about their performances.

Table 3. Assemblies of repeats in three species.

Species	Assemblers	Family	F-acc (%)	R-acc (%)	T-size (.kb)	N50-acc (%)	Max-acc (%)
S. aureus	NGSReper	30	79.6	82	20.7	95.2	100
	ABySS	7	31	30	6.8	27.7	53.3
	Allpaths-LG	3	11	4.3	2.7	10	4.9
	Bambus2	6	27.8	35	7.5	65	53.3
	MSR-CA	7	27.8	15	4.4	10	11.6
	SGA	18	66.7	81.8	20	94.2	100
	SOAPdenovo	9	33.3	38.6	7.5	65	53.3
	Velvet	15	50	39.3	13.8	23.5	53.3
Rhodobacter sphaeroides	NGSReper	13	67	74.5	7.8	97.2	89.3
	ABySS	13	67	44.5	8.3	75.2	19.3
	Allpaths-LG	2	9.5	6.7	1.4	96.9	19.3
	Bambus2	2	14.3	9.2	1.3	74.5	16.6
	MSR-CA	7	38	25.4	4.1	77.2	19.3
	SGA	9	67	64.3	3.9	81.9	80
	SOAPdenovo	9	76	84.5	3.7	30	86
	Velvet	5	38	24.5	2.2	74.5	19.3
Human Chr 14	NGSReper	2477	94	88.3	336.7	96	95.8
	ABySS	1246	47.3	45.4	173.5	95.6	77.1
	Allpaths-LG	2060	78.2	71.4	272.6	93	84
	Bambus2	2170	82.3	74.2	283.3	93	84.4
	MSR-CA	2339	88.8	80	308.8	94	33
	SGA	2271	86.2	77.6	296	93.5	63.8
	SOAPdenovo	2309	87.6	82.4	314.9	95.5	77
	Velvet	2089	79.3	72	275	94.3	77.2

All contigs were corrected and those smaller than 200 were removed.

Column 3 and 4 display the number and accuracy of families of captured repeats by different tools in three species. The metric family is aimed to judge the completeness in capturing repeats; the larger family indicates better completeness. For a better comparison between different tools, the circular area is plotted in Figure 2. From column 3 and Figure 2, we can observe that for any species, NGSReper can capture the most families of repeats. Furthermore, the percentage of families captured by NGSReper in three species are 32, 22, and 15% respectively, and is the highest in all corresponding species. Consequently, in terms of the completeness of captured repeats, NGSReper performed better than other tools.

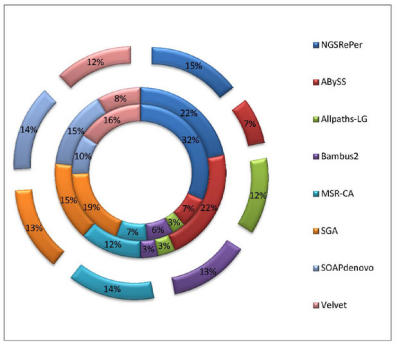

Figure 2. Circular area figure of family by different tools in three species. The percentage of different tools are noted with different colors in concentric circles, and the representation species in three concentric circles arranged from inside to outside are S.a, R.c, and H.14, respectively.

Column 5 displays the accuracy of captured repeats by different tools in three species. This metric was used to compare the accuracy of captured repeats with real repeats. Larger R-acc indicate higher accuracy. In order to graphically display the differences, a point plot of R-acc is presented in Figure 3. As shown, the performances of assemblers differ greatly in different species except for NGSReper and ABySS, whereas the performance of NGSReper is superior to that of ABySS. Additionally, unlike other tools, NGSReper was more robust to different species than other tested tools.

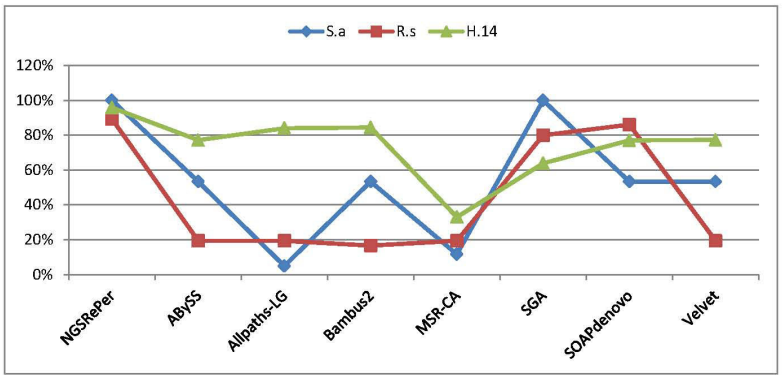

Figure 3. The point plot of R-acc by eight tools in three species. Red represents R.s, blue represents S.a, and green represents H.14. The corresponding tools are represented at the bottom of the figure.

The three metrics represented in columns 6 to 8 were used to evaluate the continuity of captured repeats, especially for N50-acc and Max-acc. For the total size of captured repeats, NGSReper capture 20.7, 7.8, and 336.7 kb in S.a, R.s, and H.14 respectively, and was nearly superior to the other assemblers. To compare the performance of assembly tools in capturing large repeats, a histogram of Max-acc has been plotted in Figure 4, which clearly indicates that NGSReper is more robust to species when compared to the other seven tools. Moreover, compared to other tools, the overall performances of NGSReper in capturing large repeats is also better.

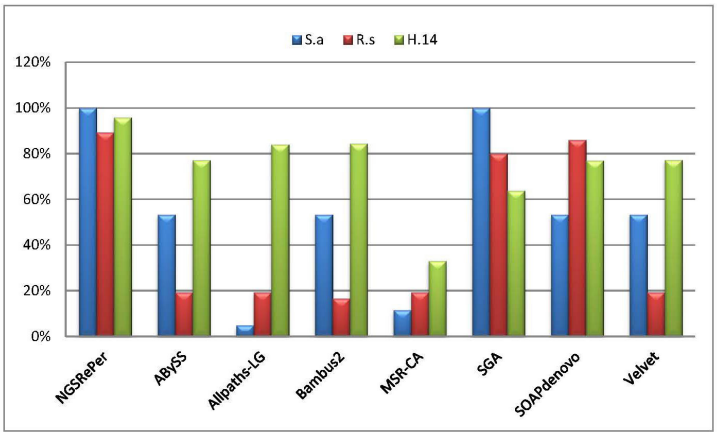

Figure 4. The histogram of Max-acc by eight different tools in three species. Red represents R.s, blue represents S.a, and green represents H.14. Different columns present the results of the different tools in three species.

The three real NGS datasets studied represent a wide range of genome sizes. From the comparison, we can conclude that (1) in terms of the completeness and accuracy in capturing repeats, NGSReper performed best among the eight assemblers, and that (2) NGSReper is more robust to species than others in terms of detecting large repeats. However, there is still some variations of the results between different species using the same method, implying that different assemblers have different emphasizes in the process of genome assembly. Moreover, the differences in the intrinsic repeat structure among different species also influences the performance and, consequently, the results.

DISCUSSION

The identification of *de novo* repeats from NGS data is a difficult task for genome analysis, and is still challenging to many genome assembly algorithms. Theoretically, a good genome assembly algorithm not only can assemble genome accurately, but also can capture a repeat completely. Although, a large number of tools have been proposed to address this problem, they still need to be improved. Various challenges and further improvements are discussed as follows.

1) Short reads: the importance of NGS technologies is their high throughput and short reads. Shorter reads provide less information for repeats capturing in the assembly process. Furthermore, the size of the repeat is always larger than the read, therefore, one read cannot stride across the whole repeat, which leads to a repeat being sub-sampled by tens or hundreds of reads.

2) Similarity: repeats can be classified into identical and similar repeats. Researchers define similarity differently according to their research task. Consequently, the similarity is another challenge for capturing repeats from NGS data. In general, the range of similarity is between 80-98%. Non-uniformed similarity leads to a difficulty in detecting highly similar repeats. Herein, we define similarity as 95%. Moreover, the sequencing errors hamper repeat similarity. As a result of the intrinsic error of the technique, the assemblers cannot distinguish whether the difference is caused by similarity or sequencing error.

3) Families: although repeats are very common in eukaryote genomes, the determination of families lack an uniform standard and is closely related to similarity, length, copy number, and biological significance. For example, in Figure 5, only considering length it is difficult to distinguish whether there are two repeat families (A and B) or only one repeat family C, when the last sequence A is abandoned. Therefore, larger families may be not beneficial for the practical biological research.

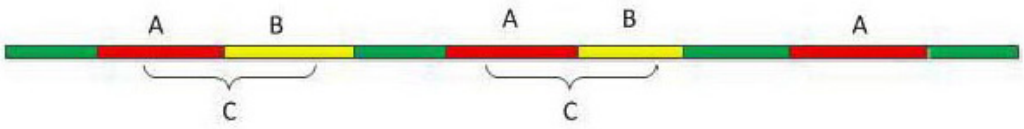

Figure 5. Graphic illustration of repeat length and copies. Green represents the reference genome, red and yellow represent different types of repeats with variable length (repeat A: three copies 150 bp, and repeat B: two copies 100 bp).

4) Types: the repeats can be classified into interspersed, tandem, and compound repeats. Each type also can be classified into many different sub-types. The complexity of repeat types is an obstacle for capturing repeats. Eukaryote genomes contain different types

of repeats. Notably, the compound repeats are found nearly everywhere. For example, Figure 5 shows two repeats (A with three copies and B with two copies) and one repeat (C with two copies, containing A and B), making it difficult to determine which one is correct. If a research focuses on the length of the detected repeat it would use the C repeat, while those focusing on the copy number may prefer the two repeats A and B.

5) Classification and annotation: the abilities of repeat classification and annotation are also important features for repeat capture tools. In addition, the annotation of repeats is helpful for the downstream bioinformatics analysis. Moreover, the inclusion of classification abilities may improve their overall utility and encourage a widespread use. However, the classification and annotation of repeats are continually changing as new element types and relations between elements are being discovered. Consequently, the current NGS-based repeat capture tools lack generalized classification and annotation abilities.

6) Identification of short repeats, such as motifs. De novo repeat capturers assemble sequences or sequence sets searching for nucleotide motifs that occur more commonly than expected if nucleotide distribution were random. For the repeats larger than the read length, the current tools can easily capture them by assembling from NGS reads, whereas for those smaller than the read length, such as motifs, the assembly-based tools are useless. Consequently, for capturing short repeats, a motif identification-based method, such as kmerHMM (Wong et al., 2013), will be more helpful than the assembly-based method.

CONCLUSION

Eukaryotic genomes contain large number of repeats, which can be used in the construction of high-density genetic maps and enable the molecular tagging of genes. The identification of these repeats is gaining importance. Simultaneously, the fast development of NGS technologies has inspired a flood of new projects aiming to sequence a variety of animals and plants. Consequently, capturing repeats directly from NGS data in the genome assembly process is becoming more attractive. However, the current leading assemblers perform poorly in directly capturing repeats from NGS reads. An optimal genome assembler not only should assemble genomes, but also should capture repeats from NGS data.

To this end, a *de novo* genome assembly for capturing repeats based on NGS data, named NGSReper, is proposed herein. The strategy of NGSReper is based on the combination of dynamic overlapping assembly and a greedy extension graph. To evaluate the performances of NGSReper, different types of datasets, including simulated and real NGS data, were used. Simulated data was used to validate the feasibility of NGSReper, whereas real NGS reads were used for cross validation. Consequently, extensive comparisons were conducted with other seven leading assemblers, such as Velvet, SOAPdenovo, SGA, MSR-CA, Bambus2, ALLPATHS-LG and ABySS, in real NGS datasets. Our results indicate that: 1) In terms of the completeness and accuracy of captured repeats, NGSReper performed best among the eight tested assemblers, and 2) in terms of the robustness in capturing repeats from different species, NGSReper also outperformed other assemblers. Consequently, NGSReper is a suitable repeat capture tool for NGS data.

Conflicts of interests

The authors declare no conflicts of interest.

ACKNOWLEDGMENTS

Research supported by the National Natural Science Foundation of China (Grant. #61501392), the Key Scientific Research Project of the Education Department of Henan Province of China (#15A510010), and the Doctoral Scientific Research Start-up Funds of XYNU (#0201447). In addition, this study was financed in part by the Nanhu Scholars Program for Young Scholars of XYNU.

REFERENCES

Bowen NJ and Jordan IK (2002). Transposable elements and the evolution of eukaryotic complexity. *Curr. Issues Mol. Biol.* 4: 65-76.

Buard J and Jeffreys AJ (1997). Big, bad minisatellites. *Nat. Genet.* 15: 327-328. http://dx.doi.org/10.1038/ng0497-327

Dohm JC, Lottaz C, Borodina T and Himmelbauer H (2007). SHARCGS, a fast and highly accurate short-read assembly algorithm for de novo genomic sequencing. *Genome Res.* 17: 1697-1706. http://dx.doi.org/10.1101/gr.6435207

Gnerre S, Maccallum I, Przybylski D, Ribeiro FJ, et al. (2011). High-quality draft assemblies of mammalian genomes from massively parallel sequence data. *Proc. Natl. Acad. Sci. USA* 108: 1513-1518. http://dx.doi.org/10.1073/pnas.1017351108

Harris TD, Buzby PR, Babcock H, Beer E, et al. (2008). Single-molecule DNA sequencing of a viral genome. *Science* 320: 106-109. http://dx.doi.org/10.1126/science.1150427

Koren S, Treangen TJ and Pop M (2011). Bambus 2: scaffolding metagenomes. *Bioinformatics* 27: 2964-2971. http://dx.doi.org/10.1093/bioinformatics/btr520

Lander ES, Linton LM, Birren B, Nusbaum C, et al.; International Human Genome Sequencing Consortium (2001). Initial sequencing and analysis of the human genome. *Nature* 409: 860-921. http://dx.doi.org/10.1038/35057062

Li R, Zhu H, Ruan J, Qian W, et al. (2010). De novo assembly of human genomes with massively parallel short read sequencing. *Genome Res.* 20: 265-272. http://dx.doi.org/10.1101/gr.097261.109

Lian S, Li Q, Dai Z, Xiang Q, et al. (2014). A *De Novo* Genome Assembly Algorithm for Repeats and Non-Repeats. BioMed Research International. Vol. 2014, Article ID 736473, 16 pages.

Lian S, Chen X, Wang P, Zhang X, et al. (2016). A Complete and Accurate Ab Initio Repeat Finding Algorithm. *Interdiscip. Sci.* 8: 75-83. http://dx.doi.org/10.1007/s12539-015-0119-6

Loman NJ, Misra RV, Dallman TJ, Constantinidou C, et al. (2012). Performance comparison of benchtop high-throughput sequencing platforms. *Nat. Biotechnol.* 30: 434-439. http://dx.doi.org/10.1038/nbt.2198

Ma J, Devos KM and Bennetzen JL (2004). Analyses of LTR-retrotransposon structures reveal recent and rapid genomic DNA loss in rice. *Genome Res.* 14: 860-869. http://dx.doi.org/10.1101/gr.1466204

Metzker ML (2010). Sequencing technologies - the next generation. *Nat. Rev. Genet.* 11: 31-46.

Miller JM, Malenfant RM, Moore SS and Coltman DW (2012). Short reads, circular genome: skimming solid sequence to construct the bighorn sheep mitochondrial genome. *J. Hered.* 103: 140-146. http://dx.doi.org/10.1093/jhered/esr104

Morgante M, Brunner S, Pea G, Fengler K, et al. (2005). Gene duplication and exon shuffling by helitron-like transposons generate intraspecies diversity in maize. *Nat. Genet.* 37: 997-1002. http://dx.doi.org/10.1038/ng1615

Pop M (2009). Genome assembly reborn: recent computational challenges. *Brief. Bioinform.* 10: 354-366. http://dx.doi.org/10.1093/bib/bbp026

SanMiguel P, Tikhonov A, Jin YK, Motchoulskaia N, et al. (1996). Nested retrotransposons in the intergenic regions of the maize genome. *Science* 274: 765-768. http://dx.doi.org/10.1126/science.274.5288.765

Simpson JT and Durbin R (2012). Efficient de novo assembly of large genomes using compressed data structures. *Genome Res.* 22: 549-556. http://dx.doi.org/10.1101/gr.126953.111

Simpson JT, Wong K, Jackman SD, Schein JE, et al. (2009). ABySS: a parallel assembler for short read sequence data. *Genome Res.* 19: 1117-1123. http://dx.doi.org/10.1101/gr.089532.108

Stein LD, Bao Z, Blasiar D, Blumenthal T, et al. (2003). The genome sequence of *Caenorhabditis briggsae*: a platform for comparative genomics. *PLoS Biol.* 1: E45. http://dx.doi.org/10.1371/journal.pbio.0000045

Salzberg SL, Phillippy AM, Zimin A, Puiu D, et al. (2012). GAGE: A critical evaluation of genome assemblies and assembly algorithms. *Genome Res.* 22: 557-567. http://dx.doi.org/10.1101/gr.131383.111

Treangen TJ and Salzberg SL (2012). Repetitive DNA and next-generation sequencing: computational challenges and solutions. *Nat. Rev. Genet.* 13: 36-46.

Wong KC, Chan TM, Peng C, Li Y, et al. (2013). DNA motif elucidation using belief propagation. *Nucleic Acids Res*. 41: e153.

Zerbino DR and Birney E (2008). Velvet: algorithms for de novo short read assembly using de Bruijn graphs. *Genome Res.* 18: 821-829. http://dx.doi.org/10.1101/gr.074492.107