JIS Journal of Interdisciplinary Sciences
Volume 1, Issue 1; 32-48
November, 2017. © The Author(s)

A New AB Initio Repeats Finding Algorithm for Reference Genome

Shuaibin Lian*, Ke Gong, Xiangli Zhang and Xinwu Chen

College of Physics and Electronic Engineering, Xinyang Normal University, Xinyang, China
shuai_lian@qq.com*

Abstract: *It has become clear that repetitive sequences have played multiple roles in eukaryotic genome evolution. However, identification of repetitive elements can be difficult in the ab initio manner from reference sequence. Currently, some classical ab initio tools of finding repeats have already presented. The completeness and accuracy of detecting repeats of them are very low and need to be improved. To this end, we proposed a complete and accurate ab initio repeat finding tool, named UnSaReper, which is based on unbiased sampling and dynamic overlapping extension strategy. The performances of UnSaReper are compared in human genome data Hg19 with RepeatScout and RepeatFinder. The results indicate the following conclusions: 1) The completeness of UnSaReper is the best one in almost all chromosomes; 2) In terms of total size, UnSaReper is also more powerful than others. Consequently, UnSaReper is a complete and accurate ab initio repeat finding tool.*

Keywords*:* Repeat Finder, Unbiased Sampling, Dynamic overlap, greedy extension graph.

Introduction

The genomes of all eukaryotes contain repetitive elements of varying lengths, and which can occupy a significant fraction of the total DNA content (Sharma et al. 2004). More than 50% of the human genome is thought to consist of repeats of various types (Lander et al. 2001). Repetitive elements have played, and are continuing to play, critical roles in genome evolution (Kazazian et al. 2004). Mobile repetitive elements (i.e. transposons), in particular, appear to be an agent of evolutionary change with some extreme stress (i.e. when desperate measures such as creating new mutations may prove advantageous) (Bennetzen et al. 2000).

The molecular evidence also suggests that some repeat elements may be instrumental in generation of new genes (Morgante et al. 2005). Moreover, repeats can have profound influences on gene expression (Assaad et al. 1993; Zuckerkand et al. 1995). Likewise, mobile element insertions can cause epigenetic changes in regulation of nearby genes (Lippman et al. 2004). There are two major groups of repeats in eukaryotic genomes: tandem repeats and dispersed repeats (Jason et al. 2010). Tandem repeats are grouped into three major subclasses: satellites, mini-satellites and microsatellites; Likewise, dispersed repeats can also be sub-grouped into five types (Smit 1996): Short Interspersed Nuclear Elements (SINEs), Long Interspersed Nuclear Elements (LINEs), Long Terminal Repeats (LTRs), DNA transposons and others.

Consequently, repeat identification is a critical part of the analysis of a newly sequenced genome and is of considerable importance. A number of computing algorithms have been developed to handle this problem. These algorithms can be classified to two categories: unassembled reads based methods and assembled sequences based methods. For the assembled sequences based, two strategies are employed (The library based strategies and ab initio strategies). The library based methods mainly include RepeatMasker (Smit and Green 2013), Censor (Jurka et al. 1996) and MaskerAid (Bedel et al. 2000).

Library-based systems identify repetitive sequences by comparing input datasets against the repeat database. Thus, their utility in novel repeat discovery is limited, whereas the ab initio algorithms identify repetitive elements in a manner that does not employ known repeat sequences or repeat motifs in the discovery process.

Relatively, the ab initio algorithms are superior to the library based methods for the novel repeats identification, which mainly include Recon (Bao and Eddy 2002), PILER (Edgar and Myers 2005), RepeatScout (Price et al. 2005) and RepeatFinder (Shah et al. 2008a).

For the ab initio methods, two main strategies have been used to address the problem of ab initio repeat identification: similarity search and word counting. However, similarity search programs are not practical for all against all comparison of large genomes, such as Recon and PILER. Consequently, word counting is an alternative way to find repeats.

The rationale is that a genomic region containing a high number of frequent words is most likely a repeat. Currently, these ab initio repeat finders with same properties were assessed comprehensively, and the comparisons (Surya et al. 2008b) indicated that RepeatScout and RepeatFinder were top two algorithms for detecting repeats without any prior repeats database in ab initio manner overall. However, the completeness and accuracy of detected repeats by them are still relatively low and need to be improved.

To this end, we proposed a complete and accurate ab initio repeat finding tool for reference genome, named UnSaReper (Unbiased Sampling Repeat Finder), which has the following properties:

1) detecting repeats in ab initio manner without using any known pattern or repetitive sequence database in the whole process; 2) unlike RepeatScout and RepeatFinder, UnSaReper also can estimate the copies of detected repeats; 3) similar to RepeatScout and

RepeatFinder, the inputs are all assembled sequences; 4) more complete and accurate of identifying repeats.

The performances of UnSaReper were extensively evaluated and compared with other top two methods, RepeatScout and RepeatFinder, in human chromosomes Hg19 datasets. Results indicated that the performances of UnSaReper are superior to others in terms of the completeness and accuracy.

Specifically, 1) for the completeness of Family, UnSaReper is much more powerful than others in almost all chromosomes. For example in chr3, there are 22375, 675 and 1918 families of repeats detected by UnSaReper, RepeatScout and RepeatFinder respectively, the performance of UnSaReper is almost 33 times and 12 times of corresponding other tools.

2) For the size of detected repeats, UnSaReper also outperformed others in almost all chromosomes. For example in chr2, the total size of detected repeats by corresponding three tools are 2655kb, 467kb and 779kb respectively, UnSaReper is almost 5.7 times and 3.4 times of RepeatScout and RepeatFinder. What's more, UnSaReper can also estimate the copy number of detected repeats accurately as an auxiliary function.

Consequently, UnSaReper is a complete and accurate ab initio repeat finder tool. The executable program is freely available for non-commercial users by request from the authors.

Results

The principle of UnSaReper is based on unbiased sampling and repeats assembly strategies. Similar to RepeatScout and RepeatFinder, UnSaReper takes the whole genome or assembled sequence as the input. Then, subsequences are sampled uniformly from the input sequence to construct the reads library. Based on the library, unique process is performed to obtain the frequency of each unique one. The seed of repetitive element is selected according to the highest frequency.

Finally, seed extension strategies are used to capture repeats. The concrete steps and process are detailed as follows. UnSaReper runs in six key steps (Figure 1): unbiased sampling, library sorting, unique processing, hash index, seed selection, repeat extension.

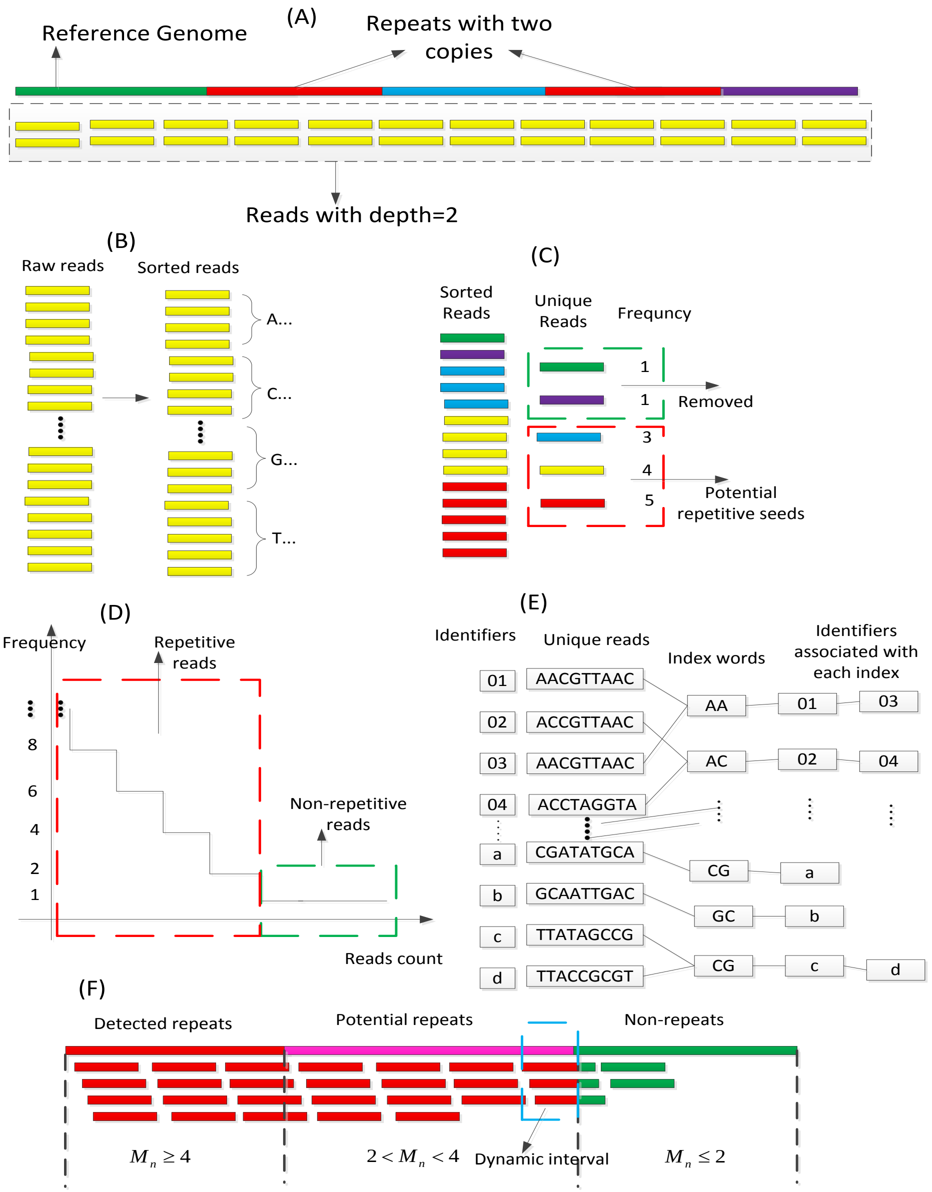


Figure 1. The graphic illustration of key steps of UnSaReper.

1) Unbiased sampling (Figure 1(A)). The subsequences with fixed length are sampled uniformly from the input sequence to construct reads library. By unbiased sampling, the reads library can uniformly cover any given region of input sequence. Therefore, a genomic region containing a higher number of frequent subsequences can be considered as a repeat. For example, if sampling depth $S_d = 2$, which indicates that any region of

the input sequence can be covered by library twice uniformly.

2) Library sorting. (Figure 1(B)). In this step, the reads in the library will be sorted by dictionary order. This sort strategy is to perform unique process and compute the frequency of the unique one.
3) Unique Processing (Figure 1(C, D)). The identical reads are collapsed into one unique read and its corresponding frequency will be recorded. Due to the unbiased sampling, the read with frequency higher than the sampling depth is tend to come from repetitive region, which will be selected as the seed for repeats extension. While, the one with frequency equals to or smaller than sampling depth is tend to come from non-repetitive region, which will be removed in the following steps. By removing the potential non-repetitive elements, the amount of data will be decreased sharply, especially for the densely sampling data.
4) Constructing hash index (Figure 1(E)). In order to improve computing speed, an indirect hash structure was designed and adopted in this part. Firstly, the index key words are transformed into quaternary integers instead of the string itself. 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 greatly reduced. Consequently, this structure is appropriate for the DNA reads, especially for the large datasets.
5) Seed selection (Figure 1(C, D)). An initial read sequence in the library which is so called a seed is necessary to initiate the repeat finding process. Therefore, the seed should be selected with frequency higher than the sampling depth.
6) Repeats finding. After selection of seed, the strategy of greedy graph extension (Dohm et al. 2007) will be performed in the dynamic overlapping interval (Lian et al. 2014), which is an appropriate range of $[min, max]$, where min and max are minimum and maximum overlap. Given a seed, UnSaReper firstly extends at the 3' end and then at the 5' end. In each extension, the mean value $M_n$of sampling depth in the interval will be computed, which is used to control the repeat finding process. If $M_n < S_d + \sigma$, the repeat finding process will be stopped, where 0<$\sigma < S_d/2$ is a tuning parameter.

Figure 1. The graphic illustration of key steps of UnSaReper. (A) Unbiased sampling. Input sequence contains one repeats with two copies and three non-repeats (red represents repeats with two copies, blue, green and violet represents three non-repeats), reads are sampled from the reference genome uniformly, the default sampling depth is set to $\mathrm{S_d} = 2$ that is each locus was sampled twice. (B) Library sorting. The reads in the library will be sorted in dictionary order. (C) Unique Processing. The five different color lines represent the five unique reads in sorted library. Each of them appears with different frequency. By unique processing, the identical reads are collapsed into unique one and its corresponding frequency.

The reads with frequency smaller than S_d will be removed. (D) Seed selection. The unique reads are ranked by frequency (from high to low). The reads with frequency higher than sampling depth S_d will be selected as the seeds for repeats (the red dotted frame), while the others will be removed. (E) Hash index construction. (F) Repeat finding. UnSaReper applied greedy extension strategy to dynamic overlapping interval to capture repeats. In the extension process, UnSaReper firstly extend at 3' end and then 5' end. Meanwhile, the mean value of sampling depth M_n in the interval is used to detect the boundary of repeats and control whether the extension process can be continued. For example, if $S_d = 2$, the mean sampling depth $M_n$in region of repeats will be satisfied $M_n \geq 4$. Therefore, if$M_n < 4 - \sigma,\ \ 0 < \sigma < 1$, which indicates the region is probably the boundary of repeats or non-repeats, and the extension process will be stopped.

Assessments

In this part, we evaluated the performances of UnSaReper and compared with other two ab initio methods, RepeatScout and RepeatFinder, in human genome datasets Hg19. These two methods performed best among the same kind of tools in reference (Surya et al. 2008b), what's more, the inputs of these three tools are all reference genomes or sequences. The detailed results are presented as follows.

A. *Metrics*:

Due to the same characteristics of three tools, it is only fair to use the same inputs and recognized metrics to compare the performances of them. Consequently, the metrics, such as Family, N50, Max, Total size, Repeat Accuracy and Copy Accuracy are employed. Some of them are widely used in reference (Surya et al. 2008b) , such as, Family, N50, Total size. Repeat Accuracy and Copy Accuracy are specially designed for evaluating the correctness and accuracy of detected repeats. Their definitions and effectiveness are as follows:

Family: a group of repetitive sequences inferred to have a common ancestor based upon sequence similarity. Thus, in this paper, the similarity is set to 90%. This metric is used to evaluate the completeness of types of detected repeats. A good repeat finder tool should detect more completeness of families of repeats. The larger family indicates more types of repeats are detected, which means the method has more completeness of detecting repeats. Therefore, larger family is preferred.

N50: The N50 value is the size of the smallest repeat such that 50% of the repeats is contained in repeats of size N50 or larger, and which is used to evaluate the continuity of detected repeats. The larger N50 indicates the better continuity of detected repeats.

Max: the maximum detected repeat, which is used to evaluate the performances of detecting large repeats. Therefore, the larger Max indicates the larger repeat can be detected.

Total size (T-size): the total size of detected repeats, which is used to evaluate the completeness of length of detected repeats, and which is defined as follows. $L_T = \sum_{i=1}^{N} l_i$, where L_T is the total length of all detected repeats, l_i is the ith family of repeat, N is the number of family. For example, reference genome M contains three Families of repeats: repeat A with 100 copies and length 500bp, repeats B with 50 copies and length 1000bp, and repeats C with 20 copies and length 1500bp. Therefore, Family=3, N50=500bp, Max=1500bp, Total size $\mathrm{L_T = 500 + 1000 + 1500 = 3000bp}$.

Repeat Accuracy (R-Acc): the accuracy of detected repeats, which is used to evaluate the accuracy of detected repeats and defined as follows:

$$\mathrm{R - Acc = 1 - \frac{N_e}{N_f}}$$

Where $\mathrm{N_f}$ is the number of real family, $\mathrm{N_e}$ is the number of wrong families. The error detected repeat was defined as the one whose similarity with reference genome is lower than 95%.

Copy Accuracy (C-Acc): the accuracy of the copy numbers of detected repeats, which is defined as follows.

$$\mathrm{C - Acc = 1 - \frac{\sum_{i=1}^{Tc} \frac{|N_{ri} - N_{ei}|}{N_{ri}}}{T_c}}$$

This metric is a relative accuracy of copy numbers of detected repeats, and which can assess the accuracy without the influences of dramatic changes of one repeat. Where $\mathrm{T_c}$ is the total copies of repeats. $\mathrm{N_{ri}}$ is the real copy numbers of $\mathrm{i_{th}}$ family of repeat, $\mathrm{N_{ei}}$ is the estimated copy number of corresponding repeat.

For evaluating the accuracy, the metrics, such as Repeat accuracy and Copy accuracy are computed by aligning the corresponding items back to the reference genome using program swalign in MATLAB platform. The default similarity is set to 95%. Among these metrics, Family, N50, Max and Total size are specially designed for judging the completeness of detecting repeats, while the Repeat accuracy and Copy accuracy are specially designed for judging the accuracy of detected repeats and copy numbers.

B. Performances assessment

Hg19 dataset contains all 24 human chromosomes, which represents a wide range of genome size and different repeats structures. Therefore, the extensive evaluations and comparisons of these three tools can make lots of sense in Hg19. Furthermore, these three tools are all ab initio repeat finder, each of them can detect repeats without any repeat database or pattern. Thus, for comparison, we run three of them independently on every chromosome, and then the corresponding metrics are computed respectively. The detailed results are presented in Table 1 and the following sections.

Table 1: The results of three ab initio repeat finder tools in human 24 chromosomes

Human Chrs	Methods	Family	N50 (.bp)	Max (.kb)	T-size (.kb)	R-Acc	C-Acc
Chr1	UnSaReper	35133	105	9.13	3732.8	98.2	100
	RepeatScout	2051	1500	11	1435	68	71
	RepeatFinder	3270	270	9.5	777.9	100	100
Chr2	UnSaReper	30237	83	9.02	2655.3	99.3	100
	RepeatScout	868	900	10	467	65	63
	RepeatFinder	4570	160	18.5	779	100	100
Chr3	UnSaReper	22375	72	1.64	1681.7	99.1	100
	RepeatScout	675	800	8.1	317	57	64
	RepeatFinder	1918	150	0.8	294	100	100
Chr4	UnSaReper	22456	75	6.5	1781.1	98.4	100
	RepeatScout	774	1100	20	440	57.2	73.4
	RepeatFinder	3061	180	4.7	584.5	100	100
Chr5	UnSaReper	21459	86	9.3	2005.3	99	100
	RepeatScout	1029	2400	20	853	70	81.2
	RepeatFinder	2902	200	1.2	563.6	100	100
Chr6	UnSaReper	21223	76	7.8	1679	98.6	100
	RepeatScout	803	680	8.3	359	53.2	67.1
	RepeatFinder	2012	150	2.7	320.8	100	100
Chr7	UnSaReper	25896	88	4.2	2371	99.2	100
	RepeatScout	1224	900	17.5	655	57.2	72
	RepeatFinder	2537	170	4.2	433	100	100
Chr8	UnSaReper	16808	83	10.5	1527.4	100	100
	RepeatScout	740	2400	20	573	52.2	75
	RepeatFinder	1986	150	6.35	332	100	100
Chr9	UnSaReper	22516	280	9.2	3531	97.6	100
	RepeatScout	1890	440	20	3303	54.3	65.6

	RepeatFinder	1453	200	18	312	100	100
Chr10	UnSaReper	18804	132	7.7	2297	99.4	100
	RepeatScout	1454	1900	20	1088	65	70.4
	RepeatFinder	1430	160	6.2	256.8	100	100
Chr11	UnSaReper	17207	76	3.0	1369.7	99.3	100
	RepeatScout	837	850	10.7	398.7	25	56
	RepeatFinder	1295	160	2.0	205.8	100	100
Chr12	UnSaReper	17092	73	2.8	1304	97.6	100
	RepeatScout	692	600	11.7	292	30	61
	RepeatFinder	1965	160	3.56	315	100	100
Chr13	UnSaReper	9311	73	5.0	714.8	96.8	100
	RepeatScout	576	600	10	251	22	35
	RepeatFinder	1486	150	6.5	244.7	100	100
Chr14	UnSaReper	10698	75	1.0	846.9	97.2	100
	RepeatScout	653	700	15.3	308	21	51.6
	RepeatFinder	1950	190	1.0	326.6	100	100
Chr15	UnSaReper	13937	186	8.6	1916	98.1	100
	RepeatScout	1192	1800	12.8	853	55	81.2
	RepeatFinder	883	160	32.4	146.6	100	100
Chr16	UnSaReper	16551	123	14.7	1887	98.3	100
	RepeatScout	1390	1800	20	811	67	74
	RepeatFinder	1447	150	1.7	234.9	100	100
Chr17	UnSaReper	15832	101	5.4	1629	98.1	100
	RepeatScout	1082	900	13	590	39	58.6
	RepeatFinder	1979	160	1.36	325	100	100
Chr18	UnSaReper	6816	73	4.0	521.5	96.5	100
	RepeatScout	410	600	13.7	163.8	19	46.3
	RepeatFinder	1250	140	2.1	185	100	100
Chr19	UnSaReper	14194	78	3.0	1167	96.7	100
	RepeatScout	965	1000	17.7	567	40	71.6
	RepeatFinder	1877	230	3.4	405	100	100
Chr20	UnSaReper	6974	72	0.75	521.7	96.4	100
	RepeatScout	364	700	20	175	14	26.3
	RepeatFinder	780	140	0.87	116.3	100	100
Chr21	UnSaReper	4136	74	0.7	315.7	95.9	100
	RepeatScout	309	800	20	151	16.5	44.6
	RepeatFinder	474	130	0.41	65.7	100	100
Chr22	UnSaReper	6825	107	4.4	725.7	96.3	100
	RepeatScout	571	1000	15.7	305	58	81
	RepeatFinder	531	140	0.58	77	100	100

ChrX	UnSaReper	22776	81	5.2	2017.7	97.6	100
	RepeatScout	1224	800	20	615	35	84.5
	RepeatFinder	3288	190	12.2	667	100	100
ChrY	UnSaReper	5918	518	8.8	1167.2	96.8	100
	RepeatScout	955	4400	20	1326	30	64.3
	RepeatFinder	2820	190	5.5	564.7	100	100

Table 1 presented the detailed results of compared tools in all human genomes. Each metric has placed in a separate column in Table 1. It is indicated from the metrics that UnSaReper performed best among three tools in detecting repeat. What's more, in order to display the comparisons more clearly, metric Family, Max and T-size were plotted respectively, which make more sense about their performances.

Column 3 displays the number of families of detected repeats by three tools respectively. The metric Family is aimed to judge the completeness of finding repeats, the larger family indicates the better completeness of detected repeats. In order to clearly compare the results, the graphic illustration of column 3 were presented in Figure 2, from which one can clearly see that the metric of family by UnSaReper is much higher than two others in almost all chromosomes. For example, (1) in chr1, there are 35133, 2051 and 3270 families of repeats were detected by UnSaReper, RepeatScout and RepeatFinder respectively, which means that the performances of UnSaReper is almost 17 times and 10 times of RepeatScout and RepeatFinder respectively; (2) in chr3, there are 22375, 675 and 1918 families of repeats were detected by three of them, and corresponding times of UnSaReper for RepeatScout and RepeatFinder are 33 and 12. Furthermore, in order to show the overall performances of family by them, the percentage pie chart was presented in Fig. 3, which shows the percentage of them are 85%, 5% and 10%, the corresponding times of UnSaReper for RepeatScout and RepeatFinder are 17 and 8 respectively. Consequently, in terms of the completeness of finding repeats, UnSaReper performed best among three tools.

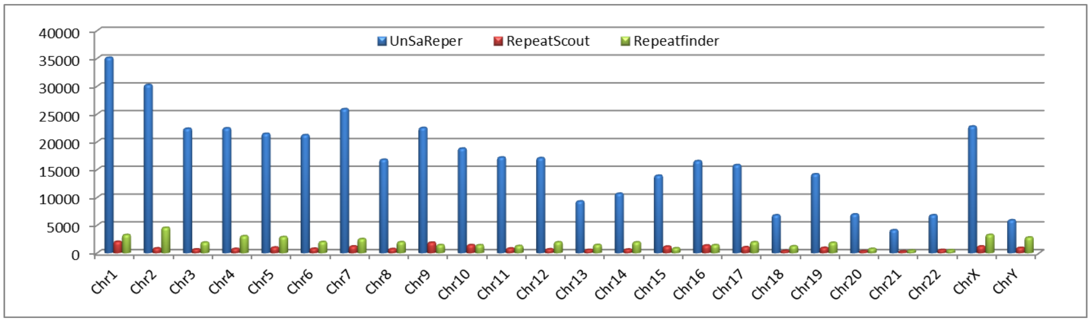


Figure 2. The bar graph of detected repeat families in human 24 chromosomes by three methods.

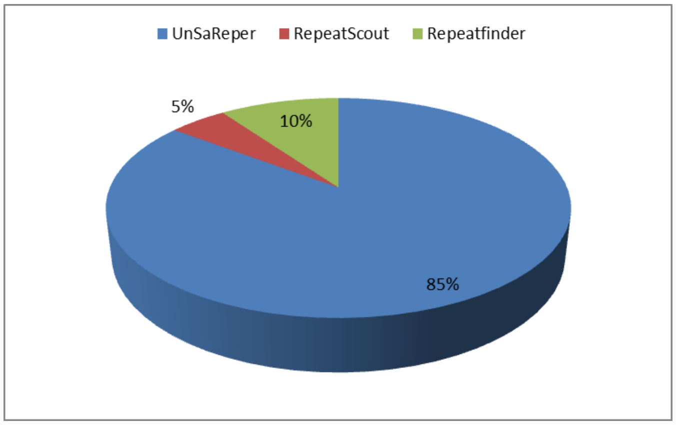
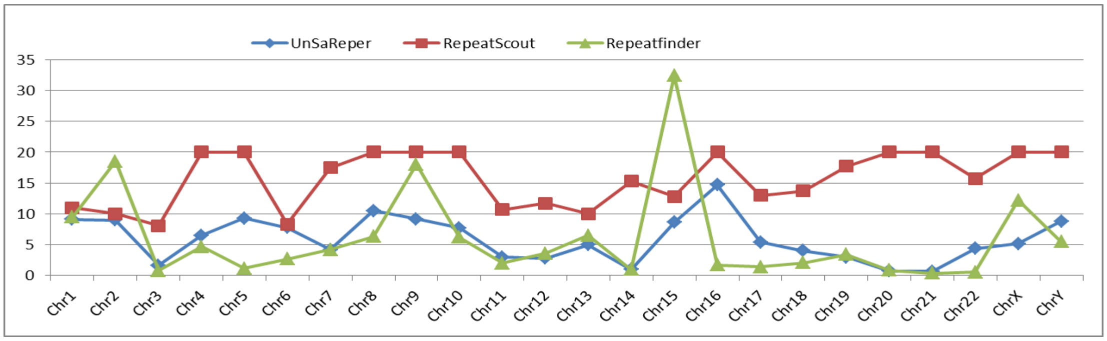


Figure 3. The percentage pie chart of family by three tools

Column 4 displays the mean size of detected repeats. This metric is for the continuity of results, the larger N50 indicates the better continuity of detected repeats. From this metric, RepeatScout performed better than UnSaReper and RepeatFinder, which indicated that the continuity of UnSaReper is less than RepeatScout and RepeatFinder. Column 5 shows the maximum size of detected repeats, which is used to assess the performances of finding large repeats. Likely, the graphic illustration of this metric has presented in Figure 4, which indicated that the performances of three tools are not consistent, and are possibly related to the size of chromosomes and intrinsic structures. Overall, in terms of detecting large repeats, RepeatScout performed best, UnSaReper and RepeatFinder performed little down.



The blue, red and green represent UnSaReper, RepeatScout and RepeatFinder respectively, the unit is kb.

Figure 4. The point plot of maximum repeats detected by three tools in human 24 chromosomes.

Column 6 presented the total size of detected repeats by three tools, this metric is aimed to evaluate the completeness of the size of repeats. For repeat finding tools, the larger T-size is preferred, which showed the better performances of finding repeats. Meanwhile, Figure 5 graphic presented the area chart of T-size by three tools in all 24 chromosomes. Clearly, UnSaReper performed best in all chromosomes in terms of the size of detected repeats.

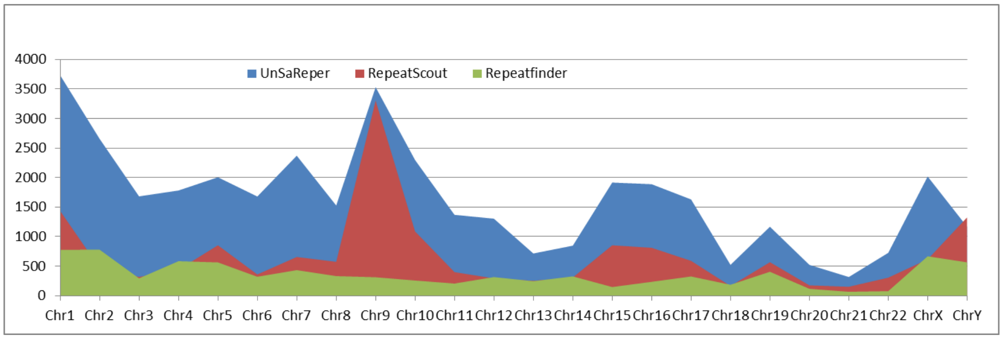


Figure 5. The area figures of total size of detected repeats by three tools in human 24 chromosomes.

Notably, the last two columns of Table 1 presented the accuracy of detected repeats, which told another story beside the quantitative relations. (1) The column 7 displayed the accuracy of detected repeats. This metric distinctly demonstrated that UnSaReper and RepeatFinder have a superior accuracy compared to RepeatScout. Although RepeatScout has the best performances of detecting large repeats, but the accuracy is too much low, which means RepeatScout detect large repeats with the expense of sacrificing the accuracy. (2) The column 8 showed the accuracy of copy numbers, which is used to evaluate the performances of estimating copy numbers. From this metric, one can clearly see that RepeatScout has no capability of estimating copies. Although both of UnSaReper and RepeatFinder can estimate copy numbers with 100% accuracy, but RepeatFinder only can detect repeats with two copies. For the repeats with three or more copies, RepeatFinder detect them separately. This strategy can lead to the redundant repeats and false families, but UnSaReper can resolve this problem well by the selecting different seed for extending. Therefore, UnSaReper has no redundant repeats and false families.

C. Sequence consensus

Comparison with known repeats can make more sense to evaluate the consensus of results. As a consequence, we compare the similarity of detected repeats with famous repeats database, Repbase (Jurka et al. 2005). In Repbase, there are 583 loci of known repeats for human species including classification, sequences, genome ID and source. The Venn diagram of compared results presented in Figure 6. From this diagram, the following conclusions can be safely come. (1) Among these 583 known repeats in Repbase, there are 258, 264 and 54 repeats detected by UnSaReper, RepeatScout and RepeatFinder respectively, and the corresponding consensus rate with known repeats is 44.3%, 45.3% and 9.3% respectively.

The generally low consensus rate indicated that the compatibility of different tools is a little poor. Nevertheless, RepeatScout and UnSaReper outperformed RepeatFinder in terms of sequence consensus rate. (2) In terms of the cross consistency, UnSaReper has 18,652 and 26,543 repeats consistent with RepeatScout and RepeatFinder respectively, the corresponding consensus rate is 82.3% and 57.6%, which means the consistency with RepeatScout is better than RepeatFinder. Consequently, in terms of the sequence consensus, UnSaReper and RepeatScout performed better than RepeatFinder.

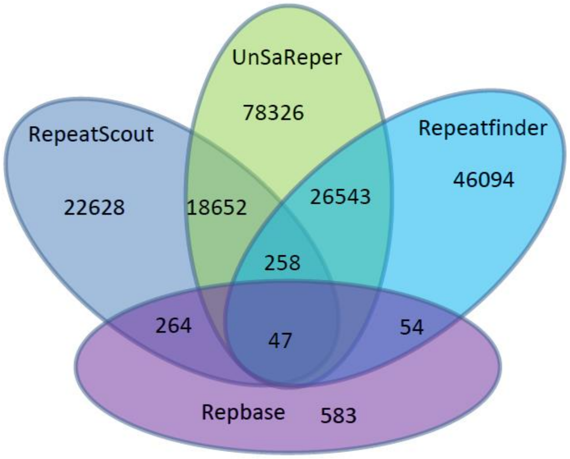


Figure 6. The Venn diagram of compared results with known repeats in database Repbase.

D. Implementations

UnSaReper was implemented in MATLAB platform and the computing requirement is: 3.5GHz eight Intel Celeron CPU with 32GB RAM and 64bit operational windows system. In order to assess the requirements of hardware of three tools, we choose two chromosomes of Hg19 randomly, chr3 and chr14. The lengths of them are 19,479,7136bp and 8,828,9540bp, and which represent a wide range of genome size and repetitive structures. The CPU times and RAM requirements are presented in Table 2.

TABLE 2: The hardware requirements of three tools in chr3 and chr14.

Chrs	Methods	CPU times	RAM
Chr3	UnSaReper	61minutes	29Gb
	RepeatScout	38 minutes	26Gb
	RepeatFinder	5 minutes	13Gb
	UnSaReper	36 minutes	22Gb

Chr14	RepeatScout	18 minutes	17Gb
	RepeatFinder	2 minutes	10Gb

From TABLE 2, we can clearly see that the CPU times of UnSaReper in both chromosomes (61minutes and 36 minutes) is longer than others and corresponding RAM is 29 GB and 22 GB, which is also larger than others. Obviously, UnSaReper has a higher hardware requirement.

Discussions

The identification of repeats in ab initio manner from whole genome or assembled sequences is a difficult task for genome analysis and is still challenging the many repeats finders, due to the complex repetitive structures and big datasets. A large number algorithms including UnSaReper have been proposed to facilitate this problem, but this work is still not finished and challenged by the following factors.

Similarity: repeats can be classified as identical repeats and high similar repeats. For identical repeats, it is a little bit easy to detect as long as the length of repeat is determined. But for the similar repeats, it is difficult to unify the consensus sequences and detect them due to the uncertainty of similarity. Different researchers define different repeats similarity according to the different research task. In general, the range of similarity is about 80%-98%. Non-uniformed similarity lead to the difficulty of detecting high similarly repeats. In this paper, we define the similarity as 90%.

Families: athough repeats are very common in eukaryotes genomes, but the determination of families is lack of uniformed standard and is closely related to similarity, length, copies and biology significance. For example only considering length in Figure 7, it is hard to tell whether there are two families of repeats (A and B) or only one family of repeats C due to abandoning the last sequence A. Therefore the larger family may be not beneficial to the practical biology research.

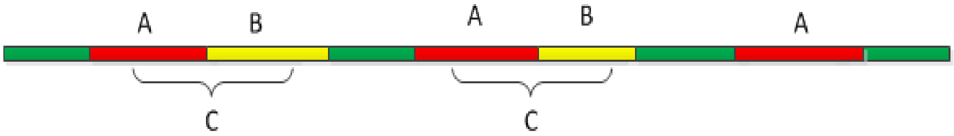


Figure 7: the graphic illustration of lengths and copies of repeats.

Length: the minimum length of repeat is another factor of challenging repeats finder. Different minimum length of repeats usually leads to different detected results. For example in Figure 7, if set the minimum length is 100bp, the detected repeat is only A with three copies, while if we set the minimum length is 90bp, the detected results is two types of repeats: repeats A with three copies and repeats B with two copies. In this paper, the minimum repeats length is set to 100bp.

Types: interspersed repeats, tandem repeats and the compound repeats. The complexity of types of repeats is also the challenge of finding repeats. Eukaryotes genomes always contain different types of repeats. Notably, the compound repeats are almost everywhere. For example in Figure 7, for the detected results, two repeats (A with three copy and B with two copy) and one repeats C with two copy(containing A and B), it is difficult to tell which one is correct. If researchers focus on the length of detected repeats, they may prefer to one repeats C, while others may argue two repeats A and B if they focus on the copies of detected results.

Copies: repeats are referred to the sequences with two or more copies. For repeat finder, it is far from enough only to detect the one with two copies. But different algorithm has different emphasis. For example, the repeats detected by RepeatFinder are all with two copies. For the one with three or more copies, RepeatFinder separated them by adding families to keep two copies, which easily lead to the redundant repeats.

Classification of repeats: Different types of repeats may have different biological significance. Consequently, it is necessary to distinguish classes of repeating elements that are well studied and characterized, such as tandem repeats or large segmental duplication. Currently, these three repeat finding tools are mainly concerned on the identification of repeats rather than the classification (Surya et al. 2008b). Whereas the classification is performed by the specialized methods in post process (Wicker et al. 2007). Therefore, the tools with the ability of identification and classification will be more attractive in the future.

Different repeat finder has different advantages and applications, such as belong to the same assembled sequence based repeat finding tool, RepeatMasker facilitate identification of repeats by comparing with repeat database. Whereas UnSaReper, RepeatScout and other ab initio repeat finding tools identify repetitive elements in a manner that does not employ known repeat database or repeat motifs in the discovery process. Likely, even though UnSaReper performed best in almost metrics in this paper, but it is not indicated that UnSaReper can replace others in repeat finding process. In contrast, UnSaReper simply provides another option for users to identify repeats from assembled genomes. Consequently, users should be aware of the advantages and disadvantages of each tool. In one word, if user takes length as the priority without considering accuracy, RepeatScout should be preferred, if user only want to detect repeats with two copies, RepeatFinder is the first choice, whereas if user takes family or total size as the priority rather than length, the UnSaReper should be preferred.

Conclusions

Genome repeats of eukaryotes occupy a significant fraction of the eukaryotes genomes. Most of them have played and are continuing to play critical roles in genome evolution. In order to detect these repeats more completely and accurately, we proposed a repeat finding algorithms, named UnSaReper, which is an ab initio repeat finding tool similar to RepeatScout and RepeatFinder, the input of them are all whole genome or assembled genome. In order to evaluate their performances, the human genome datasets Hg19 and commonly recognized metrics were employed to evaluate their performance of detecting repeats from different aspects. By the extensive comparisons in Hg19, we can safely come to the following conclusions. Firstly, the completeness of families of detected repeats by UnSaReper is much better than RepeatScout and RepeatFinder. Secondly, UnSaReper is more powerful than others in detecting the total size of repeats. Thirdly, UnSaReper also can estimate the copy numbers of each corresponding items. Lastly, UnSaReper also can resolve the problem of redundancy repeats confusing RepeatScout and RepeatFinder. In one word, UnSaReper is a complete and accurate ab initio repeat finding tool and is very suitable for the large datasets and complex repetitive structures.

Acknowledgments

This work has been supported by the National Natural Science Foundation of China (Grant. 61501392), Nanhu Scholars Program for Young Scholars of XYNU.

References

Assaad, F. F., Tucker, K. L and Signer, E. R. 1993. Epigenetic repeat-induced gene silencing (RIGS) in Arabidopsis. *Plant Mol. Biol*, vol. 22, no. 6, 1067–1085.

Bao, Z. and Eddy, S. R. 2002. Automatedde novoidentification of repeat sequence families in sequenced genomes. *Genome Res*, vol. 12, pp. 1269–1276.

Bedel, J. A., Korf, I. and Gish,W. 2000. MaskerAid: a performance enhancement to RepeatMasker. *Bioinformatics*, vol. 16, pp. 1040–1041.

Bennetzen,J.L. 2000. Transposable element contributions to plant gene and genome evolution. *Plant Mol. Biol*., vol. 42, Issue. 1, pp. 251–269.

Dohm J, Lottaz C, Borodina T, Himmelbauer H. 2007. SHARCGS, a fast and highly accurate short-read assembly algorithm for de novo genomic sequencing. *Genome Res*, vol. 17, pp. 1697–1706.

Edgar,R.C. and Myers,E.W.2005. PILER: identification and classification of genomic repeats. *Bioinformatics*, vol. 21,suppl 1(11), pp. i152–i158.

Jurka, J., Kapitonov, V.V., Pavlicek, A., Klonowski, P., Kohany, O., Walichiewicz, J. 2005. Repbase Update, a database of eukaryotic repetitive elements. *Cytogentic and Genome Research*, vol. 110, pp. 462-467.

Jurka,J., Klonowski,P., Dagman,V. and Pelton,P. 1996. CENSOR-a program for identification and elimination of repetitive elements from DNA sequences, Comput. *Chem*, vol. 20,pp. 119–122.

Kazazian,H.H.,Jr. 2004. Mobile elements: drivers of genome evolution. *Science*, Vol. 303, no. 5664, pp. 1626–1632.

Lander,E.S., Linton,L.M., Birren,B., Nusbaum,C., Zody,M.C., Baldwin,J., et al. 2001. Initial sequencing and analysis of the human genom. *Nature*, Vol. 412, pp. 860–921.

Lippman,Z., Gendrel,A.V., Black,M., Vaughn,M.W., Dedhia,N., McCombie,W.R., Lavine,K., Mittal,V., May,B., Kasschau,K.D. et al. 2004. Role of transposable elements in heterochromatin and epigenetic control. *Nature*, vol. 430, no. 6998, pp. 471–476.

Morgante M, Brunner S, Pea G et al. 2005. Gene duplication and exon shuffling by helitron-like transposons generate intraspecies diversity in maize. *Nat Genet*, vol. 37, no. 9, pp. 997-1002.

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

Saha, S, Susan Bridges, Zenaida V. Magbanua, Daniel G. Peterson. 2008a. Computational Approaches and Tools Used in Identification of Dispersed Repetitive DNA Sequences. *Tropical Plant Biology*, vol. 1, no. 1, pp. 85-96.

Saha, S, Susan Bridges, Zenaida V. Magbanua, and Daniel G. Peterson. 2008b. Empirical comparison of ab initio repeat finding programs. *Nucleic Acids Res*, Vol. 36, no. 7, pp. 2284–2294.

Sharma, D, Biju Issac, G. P. S. Raghava and R. Ramaswamy. 2004. Spectral Repeat Finder (SRF): identification of repetitive sequences using Fourier transformation. *Bioinformatics*, Vol. 20, Issue. 9, pp. 1405-1412.

Shuaibin Lian, Qinyan Li, Zhiming Dai, Qian Xiang, Xianhua Dai.2014. A De Novo Genome Assembly Algorithm for Repeats and Non-Repeats. *BioMed Research International*, Vol. 2014, Article ID 736473, 16 pages.

Smit,A.F. 1996. The origin of interspersed repeats in the human genome, Curr. Opin. *Genet*, vol. 6, no. 6, pp. 743–748.

Smit,A.F.A. and Green,P. 2013. RepeatMasker [Online]. Available: http://repeatmasker.org.

Wicker,T., Sabot,F., Hua-Van,A., Bennetzen,J.L., Capy,P.,Chalhoub,B., Flavell,A. 2007. A unified classification system for eukaryotic transposable elements. *Nature Reviews Genetics*, Vol. 8, no. 12, PP. 973–982.

Zuckerkandl,E. and Hennig,W. 1995. Tracking heterochromatin. *Chromosoma*, vol. 104, no. 2, pp. 75–83.

Paper Received September 10th, Accepted September 29th, Published 2nd November 2017