BIOINFORMATICS ORIGINAL PAPER

Vol. 29 no. 21 2013, pages 2683–2689
doi:10.1093/bioinformatics/btt482

Genome analysis

Advance Access publication August 19, 2013

PRAP: an *ab initio* software package for automated genome-wide analysis of DNA repeats for prokaryotes

Gwo-Liang Chen[1], Yun-Juan Chang[2] and Chun-Hway Hsueh[1,*]

[1]Department of Materials Science and Engineering, National Taiwan University, Taipei 10617, Taiwan and [2]High-Performance Biological Computing, Roy J. Carver Biotechnology Center, The University of Illinois, Urbana, IL 61801, USA

Associate Editor: Martin Bishop

ABSTRACT

Motivation: Prokaryotic genome annotation has been focused mainly on identifying all genes and their protein functions. However, <30% of the prokaryotic genomes submitted to GenBank contain partial repeat features of specific types and none of the genomes contain complete repeat annotations. Deciphering all repeats in DNA sequences is an important and open task in genome annotation and bioinformatics. Hence, there is an immediate need of a tool capable of identifying full spectrum repeats in the whole genome.

Results: We report the PRAP (Prokaryotic Repeats Annotation Program software package to automate the analysis of repeats in both finished and draft genomes. It is aimed at identifying full spectrum repeats at the scale of the prokaryotic genome. Compared with the major existing repeat finding tools, PRAP exhibits competitive or better results. The results are consistent with manually curated and experimental data. Repeats can be identified and grouped into families to define their relevant types. The final output is parsed into the European Molecular Biology Laboratory (EMBL)/GenBank feature table format for reading and displaying in Artemis, where it can be combined or compared with other genome data. It is currently the most complete repeat finder for prokaryotes and is a valuable tool for genome annotation.

Availability: https://sites.google.com/site/prapsoftware/

Contact: hsuehc@ntu.edu.tw

Supplementary information: Supplementary data are available at *Bioinformatics* online.

Received on February 5, 2013; revised on August 13, 2013; accepted on August 14, 2013

1 INTRODUCTION

Advancements in sequencing technology have led to a rapid increase in genomes sequenced and a raised demand for accurate and complete genome annotation. It is highly desirable to have an adequate genome annotation software capable of performing essential tasks such as predicting correct genes and repetitive sequences. All organisms have, in the total DNA of a cell, multicopies of similar sequences; i.e. the repeats. Hence, repeat identification and masking have been considered as an essential phase of the sequence assembly and annotation for the eukaryotic genomes, but not in prokaryotes (Andrey *et al.*, 2010; Smith *et al.*, 2007). For many years, repetitive sequences in prokaryotic genomes have been studied in individual cases. Research now indicates that prokaryotic repeats have an important role as is currently understood for eukaryotes (Barrangou *et al.*, 2007; Brouns *et al.*, 2008; Koressaar and Remm, 2012; Song *et al.*, 2012; Trivedi, 2006; van Belkum *et al.*, 1998).

Repeats in prokaryotes include (i) local repeats, such as low-complexity repeats, simple repeats and tandem repeats, (ii) repeats in genes that are duplicated or functionally related, such as duplicated genes, protein domains and rRNA operons, (iii) mobile repeats that play an important role in genome plasticity and evolution, such as insertion sequences (IS) and transposons and (iv) interspersed or intergenic repeats that may provide the regulatory or structural variations of chromosomes to adapt to environmental challenges, such as clustered regularly interspaced short palindromic repeats (CRISPRs).

However, in many completed prokaryotic genome annotations reported to GenBank, repeat features have been completely absent. Merely ~30% [search NCBI nucleotide database with command: bacteria/archaea (organism) and 'complete genome'(Title) and not plasmid(Title) not 'vector' (title) w/o and 'repeat_region'] of the organisms contain partial repeat features and none contains complete repeat annotations. This is likely due to the following reasons: (i) the repeat functions have been underestimated; (ii) there are no repeat libraries available for the prokaryotic genomes in the RepeatMasker database, which is currently the most widely used repeat finder software; (iii) based on the assumption that all open reading frames are candidates of prokaryotic genes, gene prediction and analyses have been performed on prokaryotic genomes without obtaining repetitive DNA information, which is indispensable to eukaryotic genome annotation; and (iv) the last and most important reason is that currently there is no suitable tool for the automated annotation of full spectrum repeats in the whole prokaryote genome.

A number of computing algorithms have been developed either for finding all repeat classes, such as RECON (Bao and Eddy, 2002), Parsimonious Inference of a Library of Elementary Repeats (PILER) (Edgar and Myers, 2005), WindowMasker (Morgulis *et al.*, 2006), RepeatScout (Price *et al.*, 2005) and RepeatFinder with REPuter (Kurtz *et al.*, 2001; Volfovsky *et al.*, 2001); or only for a specific pattern, such as CRISPR Recognition Tool (CRT) (Bland *et al.*, 2007), CRISPRFinder (Grissa *et al.*, 2007) and Tandem Repeats Analysis Program (TRAP) (Sobreira *et al.*, 2006) in the genome. These tools were designed to find either repeats directly or repeat families that led to the building of the repeat library used by RepeatMasker, whose purpose is to manage masking and identifying repeats.

[*]To whom correspondence should be addressed.

Each algorithm has its specific advantages and disadvantages in finding repeats. Among them, RECON used a self-aligned sequence read as a starting seed to find the groups of elements, which were then clustered into families. It has worked well for shotgun sequences or a relatively short assembled sequence in the order of Mb. RECON is capable of identifying more potentially novel repeats than the peer tools (Saha *et al.*, 2008). Thus, it was adopted in our study as the repeat library builder to develop the PRAP. MegaBlast uses the greedy algorithm to search similar sequences, and was thus chosen to perform the self-search (all-against-all) in the first step of repeat identification. In this article, we report this fast and accurate automated method that is aimed to identify all families of repeats at a genomic scale. We believe that this is currently the most complete repeat identification tool designed for prokaryotic genomes.

2 METHODS

The code was written in Perl and run in UNIX. Perl Interpreter 5.6.1 or later (http://www.perl.org/), MegaBlast/blastn (http://www.ncbi.nlm.nih.gov/BLAST/), RECON (http://selab.janelia.org/recon.html), VisCoSe (http://bio.math-inf.uni-greifswald.de/viscose/html/download.html), RepeatMasker (http://www.repeatmasker.org/) and Artemis (http://www.sanger.ac.uk/Software/Artemis/) are required to run the code. All codes must be installed in the user's execution path.

Figure 1illustrates how PRAP assisted in automatically running the MegaBlast, RECON, VisCoSe and RepeatMasker. The integrative pipeline is outlined below to explain the complete genome repeat analysis for prokaryotes.

2.1 Genome-wide repeat elements identification

The pipeline begins with identifying potential repeat elements via a similarity self-search (Altschul *et al.*, 1990; Claverie and States, 1993) using MegaBlast. Two sequence regions are considered as potential repeats if they share more than the users' desired (default 80%) sequence identity. The default minimum length of the element for search was set as 20 bp.

2.2 DUST filtering

Low complexity or highly repetitive sequences generate artificially high-scoring alignments that could perturb the sequence relationships in similarity search. Such sequences should be managed carefully when performing sequence similarity search. There are a few sequence filters, such as DUST for nucleic acids (Tatusov and Lipman, unpublished manuscript) and XNU for amino acids (Claverie and States, 1993), available for BLAST programs. To capture full spectrum repeats, two MegaBlast runs are performed: one with and one without DUST filtering. The raw hits from the two runs are merged and further screened with desired similarity score. Those duplicated hits are removed to maintain a unique record. The run with DUST filtering (–F T) will find short repetitive segments embedded in the long repetitive sequences, whereas the run without DUST filtering (–F F) will ensure that long repetitive segments are not disrupted by filtering. This enables the identification of low complexity repeats within long repeats, e.g. gene motif inside a gene. This approach is an important step to capture all possible repetitive segments. The necessity of combining two MegaBlast runs in the pipeline is further illustrated in Section 3.5.

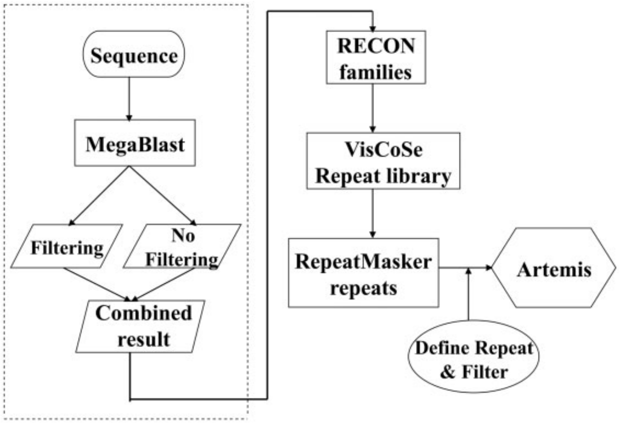

Fig. 1. The flowchart of the PRAP software package

2.3 Repeat families by RECON

The repeat elements from the above step are parsed and converted into the RECON input format. RECON is then used to identify primary repeat families. Overlapping elements are considered as the synoptic by clustering alignment to form families (Bao and Eddy, 2002). RECON is one of the major components in this package for family identification. Owing to the pairwise similarity algorithms adopted, RECON has been reported to be slow in computing because it requires large amounts of main memory and may have difficulty in defining repeat boundaries (Price *et al.*, 2005). However, for prokaryotic genomes, this potential issue is minimal owing to the relatively small genome size.

2.4 Constructing repeat library and masking repeats

The consensuses of RECON families are generated using VisCoSe with default options. When outliers exist in repeat element length distribution of a repeat family, a parameter will be invoked to specify the longest element allowed to reduce the noise. VisCoSe uses motif conservation rate and frequency to construct consensus from aligned sequences. This is relatively simpler than using the weight matrix methods (Spitzer *et al.*, 2004). The repeat type is determined by using blastn of the repeat family consensus against known sequence databases, such as NCBI nt, ISFinder and GenBank, and by detecting the repeat pattern using the script included. The resulting consensus is then used as the repeat library for RepeatMasker running with the '–lib' option, which indicates a custom library. A tab-delimited text file is generated listing the repeat family, repeat type, frequency, location within the source sequence and the longest and the shortest sequence lengths in the family.

2.5 Outputs and optional filters

A few optional filter programs can be applied to the RepeatMasker output to sort out repeats based on the number of repeat occurrences, repeat length or repeat type, and to remove unwanted features according to the user's desire. Finally, the filtered output is parsed to the EMBL/GenBank feature table format for displaying and viewing in Artemis, where it can be combined or compared with other genome analyses data.

2.6 Pipeline optimization and efficiency

The software default parameters have been optimized for novice users and can be easily applied to most prokaryotic genomes. For experienced users, customized parameters can be adjusted to improve accuracy by removing ambiguous repeats, which might be accumulated and carried from the limitations of the individual software component in the package. This software has been demonstrated to be efficient for microbial genome sequences sized (0.5 ~ 10 Mb) in an optimum condition. MegaBlast and RepeatMasker take most of the running time. For an average genome size (~5 Mb) in a UNIX system, Intel Core i7-920 Processor @ 2.67 GHz, it takes ~30, 12 and 0.5 min to run MegaBlast, RepeatMasker and RECON, respectively. The time consumption of running MegaBlast varies widely based on chosen parameters. RepeatMasker runs in this package using the default options with ‘–lib’ to specify the library obtained from the previous step, which contains only hundreds of data elements for prokaryotic genomes. Hence, the actual running time of RepeatMasker is minimized, and this renders the tool with high efficiency.

3 PIPELINE PERFORMACE

To evaluate the PRAP performance, we compared its ability in identifying repeats with other well-known tools and verified its accuracy with manually curated and experimental data.

3.1 Comparison using general purpose tools for genome-wide repeats structure

Chloroflexus sp. Y-400-fl, whose complete GenBank annotation includes significant repeat features, is a repeat-rich prokaryote. Hence, we selected this organism as an example for a genome-wide comparison of our software performance with other existing tools, RepeatFinder, RepeatScout and RepeatModeler (Smit and Hubley, 2012). RepeatFinder finds repeats directly without applying RepeatMasker. The other tools create a custom repeat library to be analyzed as sequences in RepeatMasker. To ensure a fair comparison, the default parameters were used to run the tools.

Table 1 summarizes the repeats found by different tools. The three listed values indicate repeat elements found, not found and falsely reported, respectively. The element is reported as false when it does not follow the repeat type definition. Detailed results are listed in Supplementary Table SI. No repeat is generated by running RepeatModeler, which, in turn, indicates that its algorithm may not be applicable to the prokaryotic genome and is therefore excluded from the comparison shown in Table 1. For simplicity, we limited our comparison to those repeat elements listed in GenBank file. The GenBank results did not specify the method used and it reported 46 repeat regions: 15 transposons in 3 families, 4 CRISPRs and 27 segmental repeats. Compared with other tools’ results, GenBank data missed 7 transposon elements (including 3 in ISDge7 type, 3 in ISMhu9 type and 1 in ISEhe2 type) and 36 segmental repeats. These missing mobile elements were mistakenly annotated as pseudogenes. Similarly, most missing segmental repeats were annotated as either hypothetical genes or unknown CDSs (see details in Supplementary Table SI). GenBank data were found to falsely report short-period tandem repeats as CRISPRs.

Among these tools, RepeatFinder exhibits low quality, whereas RepeatScout and PRAP show comparable results. All reported repeats in GenBank can be identified by RepeatScout and PRAP with reasonable boundaries, as evidenced by comparison with GenBank’s transposon genes boundaries. The major difference between the two tools occurs within the sufficiently diverged regions, where RepeatScout tends to only identify partially conserved domains to form disconnected subrepeats in the same family, whereas PRAP identifies the entire region as a repeat belonging to other family instead (see repeats in Supplementary Table SI marked in green). It should be noted that both RepeatScout and RepeatFinder failed to correctly determine CRISPRs elements, which are one of the most needed repeat features in the prokaryotic genome. This indicates that their capability of finding short repeats is inferior to PRAP. Hence, our method achieves better prediction in this comparison.

Table 1. Genome-wide repeats comparison among different tools

Repeat family	GenBank	Repeat finder	Repeat scout	PRAP
CRISPRs	**4**/0/**1**	0/**3**/0	3/0/**3**	3/0/0
Repeat segments	27/**36**/0	20/**43**/0	63/0/0	63/0/0
Transposons				
ISDge7	8/**3**/0	7/**4**/0	11/0/0	11/0/0
ISMhu9	2/**3**/0	4/**1**/0	5/0/0	5/0/0
ISEhe2	5/**1**/0	3/**3**/0	6/0/0	6/0/0

Note: The first, second and third numbers of each entry indicate the number of elements found, not found and falsely reported by the method. The boldface numbers highlight the incorrect results. ‘0’ indicates good result in the second and the third entries.

3.2 Comparison using specialized tools on interspersed repeats structure

CRISPRs are one of the most important repeat families in prokaryotic genomes. The locus contains a succession of highly conserved regions (direct repeats) varying in size from 23 to 47 bp, separated by similarly sized unique non-repetitive sequences (spacer) usually of viral origin (Jansen *et al.*, 2002a, b; Mojica *et al.*, 2005). Organisms with multiloci CRISPR have conserved the leader sequence at either the 5’ or 3’ flanking region.

Our method was verified with tools specifically designed for the purpose of finding CRISPRS, such as PILER-CR (Edgar and Myers, 2005), CRISPRFinder (Grissa *et al.*, 2007) and CRT (Bland *et al.*, 2007). We chose three genomes as examples that contain different numbers of CRISPRs cluster and possess structures that are difficult to identify: *Anaeromyxobacter dehalogenans 2cp-1 genome* (NC_011891) with two different CRISPRs clusters; *Chloroflexus sp. Y-400-f1* genome (NC_012032) with three different CRISPRs clusters; and *Zymomonas mobilis ZM4* genome (NC_006526), whose third CRISPR is split by a longer spacer (164 bp) into two parts and contains three similar short CRISPR (6–8 U) clusters. Table 2 summarizes the CRISPR analysis of three genomes adopting all above tools in comparison with our PRAP analysis.

PRAP has proved its unique merits that are superior to its peer: (i) It is the only tool that identifies all CRISPRs in three tested genomes, whereas PILER-CR misses one with both *A.dehalogenans and Zymomonas* genomes, and CRISPRFinder and CRT each misses a truncated one with *Zymomonas* genome; (ii) It is also the only tool that possesses the ability to find the leader sequence; however, this is true only when multiple CRISPRs belong to the same repeat family. CRISPR loci and leader sequences are identified as different repeat families in this method; (iii) Most importantly, it has the unique capability to minimize the background noises from simple repeats or tandem repeats. The CRT tool produces a considerable tandem repeating background as previously reported (Grissa *et al.*, 2007). Both PILER-CR and CRISPRFinder also misidentify one tandem

Table 2. Comparison of CRISPRs analysis of microbial genomes using different tools

Genomes	CRISPR	PRAP	PILER-CR	CRISPR finder	CRT
A.dehalogenans 2cp-1			549216..549454		561814..561985
	1	**883707**..886604	**883770**..886604	883709..886604	883709..886604
	2	**888743**..**892373**	Not found	888746..892491	888746..892491
Chloroflexus sp. Y400-f1	1	**709766**..731306	**709768**..**717284** **717403**..**731307**	**709669**..731306	**709670**..731306
				1637506..1637742	
	2	3958399..3986006	3958399..**3985935**	3958399..3986006	3958399..**3986066**
	3	**4543859**..4550058	4543861..4549766	4543861..4550058	4543861..4550058
					5036221..5038313 5195607..5195974
Z.mobilis ZM4	Leader	*114171*..*114318*			
	1	113843..114170	113843..**114172**	Not found	**113783**..114170
	Leader	*1242807*..1242954			
	2	1242955..1243408	1242955..1243408	1242955..**1243466**	1242955..**1243466**
	Leader	*1590207*..*1590355*			
	3I	1590356..1590747	1590356..**1590687**	1590356..1590747	1590356..**1590746**
	3II	**1590914**..1590946	Not found	1590917..1590946	Not found

Note: CRISPR regions are identified and illustrated with boundary locations.
(1) Repeat boundaries deviating from consensus sequences are highlighted in boldface; (2) CRISPRs can be interrupted by transposon or non-coding region and split into two parts as shown in the last two rows (3I and 3II) of ZM4 section; (3) Leader sequences are underlined and italicized and only found by PRAP in ZM4 section; (4) Cells that contain mis-identification of tandem repeats as CRISPRs are marked in gray; and (5) PILER-CR mistakenly splits the first locus into two loci as shown in *Chloroflexus sp* section.

repeat as CRISPRs. These tandem repeats are marked in gray in Table 2. The use of DUST filter in our PRAP algorithm masks those low-complexity repeats in the initial step and thus avoids subsequent mis-annotation of tandem repeats as CRISPR.

Moreover, both PRAP and CRISPRFinder are successful in identifying split CRISPR, as shown with the third CRISPR in *Zymomonas* genome, whereas PILER-CR and CRT fail. PILER-CR also shows the drawback of splitting the first locus into two loci as shown in *Chloroflexus sp.* section. However, it should be noted that no tool can predict boundaries perfectly. Those boundaries deviating from the consensus are marked in bold in Table 2. Nevertheless, PRAP exhibits an insignificant deviation of 1–3 bp at the 5' site and occasionally a few dozen bases off at the 3' location.

3.3 Evaluation using curated mobile repeats (insertion sequences and transposons)

In this section, we verified our algorithm with manually curated mobile repeats. The ability to identify and annotate mobile repeats is an important indication of the tool performance. *Clostridium thermocellum*, as a transposon-rich genome, was chosen for the comparison using the most frequently applied mobile repeat tool IS_Finder (Siguier *et al.*, 2006), plus RepeatFinder, RepeatScout and PRAP. The results are summarized in Table 3 (details see Supplementary Table SII). The IS_Finder data, which have been manually annotated, were used as a standard reference. In the IS_Finder column, the numbers of full-length and partial elements are given for each family. The numbers in the other columns list the deviation of the repeat number from IS_Finder finding with '+' for additional repeats found and '–' for less repeats found. The absolute deviation from each family is summed up as the total absolute deviation, which is an indication of the performance of the repeat finding ability of each tool compared with IS_Finder. A lower number indicates better performance. The bottom row in the table gives the ratio of total repeats with boundary discrepancy <6 bp from the reference repeat to the entire reference set of full or partial length repeats. The total full-length and partial elements of reference set are 95 and 23, respectively (for details see Supplementary Table SII). A higher ratio suggests more accurate boundary results. We again demonstrated that PRAP outperformed its peer in identifying IS by both number and boundary accuracy.

3.4 Evaluation using experimental data—MIRUs

Mycobacterium tuberculosis H37Rv genome experimental data were parsed from the GenBank file (NC_000962) to extract the mycobacterial interspersed repetitive units (MIRUs). A total of 65 MIRUs have been reported based on hybridization evidence and were grouped into three classes, I, II and III (Supply *et al.*, 1997). PRAP predicted all 65 of MIRUs with an additional 5 predicted; however, they are grouped into 6 families (4, 82, 56, 309, 333 and 848). Family numbers have no biological meaning. Table 4 summarizes the comparison results between experimental data and PRAP analysis of MIRUs in *M.tuberculosis*, whereas detailed repeat boundaries are listed in Supplementary Table SIII. More than half of the predicted boundaries are within a 5-bp difference from the experimental data. The boundary discrepancy could result from the experimental data that were visually inspected to trim the flank sections. Class II diverges into families 4 848 309 and 56, and class III diverges into families

Table 3. Summary for comparison of PRAP with other tools in identifying mobile repeats of *Clostridium thermocellum*

IS name/family	IS_Finder		PRAP		RepeatScout		RepeatFinder	
	Full	Partial	Full	Partial	Full	Partial	Full	Partial
IS120/IS3	16	6	0	−3	0	−3	−1	−5
ISCth1/IS982	7	0	0	+3	0	+3	−1	0
ISCth2/IS30	10	1	0	0	0	0	−1	−1
ISCth3/IS30	16	4	0	−1; +3	0	−2	0	+3
ISCth4/IS256	14	4	0	+3	0	+3	−3	−4
ISCth5/IS256	9	2	0	0	0	0	0	−1
ISCth6/IS110	3	0	0	0	+1	+1	−1	0
ISCth7IS110	2	0	0	0	−2	0	0	0
ISCth8/IS110	4	0	0	0	0	0	0	0
ISCth9/IS21	4	0	0	+2	0	+2	−1	0
ISCth10/IS200	2	0	0	+1	0	0	0	0
ISCth11/IS66	1	0	−1	0	−1	0	−1	0
ISCth12/IS21	2	2	0	0	−2	+2	0	−1
ISCth13/ISL3	2	2	0	0	0	−1	0	−3
ISCth14/IS110	1	0	0	0	0	0	-1	0
ISCth15/IS21	1	0	−1	+2	−1	+1	−1	0
ISChy16/IS701	1	1	−1	+1	−1	−1	−1	−1
Unknown/IS256	0	1	0	0	0	0	0	−1
Total absolute deviation			3	19	8	19	12	20
Repeat boundary discrepancy <6 bp			84/95	8/23	61/95	8/23	60/95	0/23

Note: The entire *Clostridium thermocellum* IS families found by four algorithms. IS_Finder results were taken as a standard reference. The entry number in IS_Finder columns is the number of IS elements in the family. The entry number of other algorithms indicates the number of the element deviating from IS_Finder result. '+' for additionally repeats found and '−' for less repeats found. The total absolute deviation is the sum of the absolute values of the entire column. The bottom row gives the ratio of total repeats with boundary discrepancy <6 bp from the reference repeat to the entire reference set of full-length or partial elements. The total full-length and partial elements of reference set are 95 and 23, respectively.

Table 4. Summary of comparison results between experimental data and PRAP analysis of MIRUs in *Mycobacterium tuberculosis H37Rv*

Experimental data		PRAP	
Class	Repeats	Family	Repeats
I	10	82	10
II	37	4	34
		848	1
		309	1
		56	1
III	18	848	12
		333	4
		82	2
Not found		4	4
		333	1

Note: PRAP fine-tuned experimental MIRUs families into more families and found more elements. Family numbers here have no biological meaning.

848 333 and 82. The families 82, 4 and 848 are dominant in classes I, II and III, respectively.

The alignment (Supplementary Fig. SI) of those 65 MIRU sequences and additional MIRUs predicted by PRAP show that classes I and II are highly conserved and consistent with the PRAP identification as families 82 and 4, respectively. However, class III is not sufficiently conserved to be identified as a single family by PRAP. The following two observations are worth mentioning: (i) Two class II MIRUs are mosaicked in longer repeats and are grouped into additional families 56 and 309. (ii) By examining the homologous regions of family 333 (highlighted in Supplementary Fig. SI) in class III, separated subclasses should be considered, which merits further study. Overall, this demonstrates that the capabilities of PRAP in identifying repeats are as reliable as the experimental work.

3.5 Effects of DUST filtering on the PRAP performance

To illustrate the necessity of combining two MegaBlast runs, PRAP was used to analyze genomes ZM4 (*Zymomonas mobilis* ZM4), Saga (*Simiduia agarivorans* SA1T) and Mutb (*Mycobacterium tuberculosis* H37Rv) in three scenarios: (i) one MegaBlast run with DUST, (ii) one MegaBlast run without DUST, and (iii) merge of (i) and (ii). The number of MegaBlast hits and the number of RECON families generated in this analysis are presented in Table 5. To study the actual difference in blast hits and RECON families between scenarios, the content of each hit and the consensus sequence of each family should be examined and compared, as counted number alone can be misleading. The results are shown in the right column of Table 5.

Table 5. Illustration of DUST effects on the PRAP performance

Genomes tested	PRAP with one MegaBlast run				PRAP with two MegaBlast runs (merged results)			
	With DUST(A)		Without DUST(B)					
	MegaBlast hits	RECON families	Genomes tested	RECON families	Merged MegaBlast hits	RECON families	DifferenceHits/families	
							From (A)	From (B)
ZM4	584	186	813	218	813	218	229 (28)/32 (15)	0/0
Saga	4635	534	5736	544	5736	544	1105 (19)/37 (7)	0/0
Mutb	10 252	1203	24 815	1288	24 833	1320	14 781 (59)/478 (36)	481(2)/176(13)

Note: All hits <80% similarity were removed. The value in 'Difference' columns indicates the number of difference in blast hits and RECON repeat families between the single and the merged runs. The number in parentheses is the percentage.

For all tested genomes, the numbers of blast hits and repeat families with dust filtering (one MegaBlast run) showed significant difference from those of merged run, and the differences ranged from 19 to 50% and 7 to 36%, respectively. For single MegaBlast run without dust filtering, the blast hit and repeat family numbers showed no difference from the merged run for genomes ZM4 and Saga. However, the effect of the merged MegaBlast run is well demonstrated for genome Mutb. By merging the two MegaBlast runs, PRAP successfully captured 481 or 2% different blast hits and 176 or 13% different repeat families compared with one MegaBlast run without dust. This could be explained by the low complexity repeat content in genome, as both ZM4 and Saga genomes are poor, but Mutb is highly rich, in low complexity structures (Nandi *et al*., 2003).

When two runs were merged, the blast hits were summed up from two single runs followed by removing the duplicated hits and subjected to other cutoff constrains. Repeat families were then created from these merged hits. In the scenarios of ZM4 and Saga genomes, the limited low complexity regions (if existing) do not influence repeats in other regions. Therefore, the without dust hits fully include the dust hits and are equivalent to the merged one. In the case of Mutb, whose low complexity region constitutes a higher fraction of the genome, the short repeats embedded in the vicinity of low-complexity regions will likely skew the alignment during similarity search, thus influencing the repeat detection in other regions. This would lead to different repeats identified between two single runs. After merging, repeat family for the merged run is determined again, which may be different from each single run. Nevertheless, we do not expect a huge difference between without dust and merged runs, as both runs process similarity search against the same (unmasked) sequence. The degree of difference is dependent on the genome's intrinsic low-complexity repeat structure and similarity identity chosen; other possible factors are not completely excluded.

This exercise demonstrates the necessity of combining the two MegaBlast runs. Using MegaBlast with dust or without dust alone would possibly lead to incomplete discovery of repeat elements or repeat families. The effects are not known a priori. To avoid the unpredictable loss, even small, users are advised to adopt the 'merged' run as default.

4 DISCUSSION

We described an *ab initio* system for rapid identification of all spectrum repeats in prokaryotic genome sequences and the assignment of these repeats to families. Prokaryotic genome, at the scale of Mb, allows BLAST to use a smaller size of search word to identify both short and long repeat segments accurately. The nature of the repeats within the prokaryotic genomes facilitates the algorithm to minimize errors generated from grouping the repeat family, thus achieving better performance.

To run PRAP, it is required to download and install BLAST, RECON, VisCoSe, RepeatMasker and Artemis from different external sources. Downloading, installing and granting the licensee permission are users' responsibility. Although these prerequisites do not update frequently, adjustment of the executing commands of the software might be required when using a different version for initiating installation or updating.

False-positive rate is a common index of specificity to assess the performance of discovery algorithms. However, by definition, a DNA repeat is two similar sequences, and the precise definition of a false positive is debatable. Moreover, unlike eukaryotes, which have extensive curated known RepeatMasker libraries (RepBase) to aid in identifying true positive repeats (Saha *et al*., 2008), prokaryotes have difficulty in evaluating the tools in terms of sensitivity and specificity. Therefore, we used the direct comparison of repeat numbers/boundary approach instead of the sensitivity and specificity of repeat as benchmarks for comparing the code performance.

PRAP currently categorizes the repeats as transposon, protein motif repeat, RNA, CRISPRs and unknown. However, it is challenging to clearly distinguish repeat families, especially when overlap and mosaic subrepeat structure occur. As stated in Section 3.4, repeat elements may be categorized into different families owing to embedding in a longer repeat belonging to the different family. Furthermore, PRAP tends to distinguish the less conserved repeats from a conserved group into different families, and it may require some adjustment using the similarity cutoff setting.

Mobile repeats are often nested in one another in the form of fragments, which become a major source of fragmented transposons. Hence, fragmented transposons, as opposed to intact transposons, are more difficult to be identified and assigned to a family. The open reading frame within these fragments can be mis-annotated as a pseudogene if the repeat is not accurately identified. On the other hand, identifying fragmented repeats can help to better annotate the pseudogenes impacted by frame shift, split or premature stop codon effect. The PRAP can identify both intact and fragmented repeats as described in Section 3.3, although the short fragmented transposons are difficult to be distinguished from interspersed repeats that are also transposable. This ability enables the tool to avoid miscalling genes in the repeat regions, therefore enhancing the gene prediction as discussed in Section 3.1. It is believed that a significant number of false hypothetical/pseudogenes can be removed from annotation when this tool is applied.

In addition, the PRAP is capable of detecting novel repeats. By facilitating its '*ab initio*' and 'identifying full spectrum repeats in genome-wide' characteristics, a new repeat library containing all repeat families can be generated for each new organism. This ensures the discovery of novel repeat. However, to achieve this, a prokaryotic repeat database has to be established such that the novel repeats can be easily identified. Our tool is currently the most complete repeat-finding tool designed for prokaryote genome. It can improve genome annotation and offer valuable potentials in repeat-related biological applications such as the characterization of metagenomes by signature repeats.

ACKNOWLEDGEMENTS

This work was initiated when Drs Chen and Chang were working at the Bioscience Division, Oak Ridge National Laboratory, USA. The authors thank National Science Council (NSC), Taiwan, for the supporting grant, and the facilities in National Taiwan University and HPCbio, the University of Illinois to complete this work. They are also grateful to Mr Marshall Lin for helping with the UNIX software installation.

Funding: National Science Council, Taiwan under Contract no. NSC 100-2221-E-002-128 and NSC 101-2811-E-002-045.

Conflict of Interest: none declared.

REFERENCES

Altschul,S.F. *et al.* (1990) Basic local alignment search tool. *J. Mol. Biol.*, **215**, 403–410.

Andrey,S. *et al.* (2010) Identification of repetitive elements in the genome of oreochromis niloticus: tilapia repeat masker. *Mar. Biotecnol.*, **12**, 121–125.

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

Barrangou,R. *et al.* (2007) CRISPR provides acquired resistance against viruses in prokaryotes. *Science*, **315**, 1709–1712.

Bland,C. *et al.* (2007) CRISPR Recognition Tool (CRT): a tool for automatic detection of clustered regularly interspaced palindromic repeats. *BMC Bioinformatics*, **8**, 209.

Brouns,S.J. *et al.* (2008) Small CRISPR RNAs guide antiviral defense in prokaryotes. *Science*, **321**, 960–964.

Claverie,J.M. and States,D.J. (1993) Information enhancement methods for large scale sequence analysis. *Comput. Chem.*, **17**, 191–201.

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

Grissa,I. *et al.* (2007) CRISPRFinder: a web tool to identify clustered regularly interspaced short palindromic repeats. *Nucleic Acids Res.*, **35**, W52–W57.

Jansen,R. *et al.* (2002a) Identification of a novel family of sequence repeats among prokaryotes. *OMICS*, **6**, 23–33.

Jansen,R. *et al.* (2002b) Identification of genes that are associated with DNA repeats in prokaryotes. *Mol. Microbiol.*, **43**, 1565–1575.

Koressaar,T. and Remm,M. (2012) Characterization of species-specific repeats in 613 prokaryotic species. *DNA Res.*, **19**, 219–230.

Kurtz,J.V. *et al.* (2001) REPuter: the manifold applications of repeat analysis on a genomic scale. *Nucleic Acids Res.*, **29**, 4633–4642.

Mojica,F.J. *et al.* (2005) Intervening sequences of regularly spaced prokaryotic repeats derive from foreign genetic elements. *J. Mol. Evol.*, **60**, 174–182.

Morgulis,A. *et al.* (2006) WindowMasker: window-based masker for sequenced genomes. *Bioinformatics*, **22**, 134–141.

Nandi,T. *et al.* (2003) A novel complexity measure for comparative analysis of protein sequences from complete genomes. *J. Biomol. Struct. Dyn.*, **20**, 657–668.

Price,A. *et al.* (2005) *De novo* identification of repeat families in large genomes (RepaetScout). *Bioinformatics*, **21** (**Suppl. 1**), i351–i358.

Saha,S. *et al.* (2008) Empirical comparison of ab initio repeat finding programs. *Nucleic Acids Res.*, **36**, 2284–2294.

Siguier,P. *et al.* (2006) ISfinder: the reference centre for bacterial insertion sequences. *Nucleic Acids Res.*, **34**, D32–D36.

Smit,A. and Hubley,R. (2012) RepeatModeler - 1.0.5. Institute for Systems Biology. http://www.repeatmasker.org/RepeatModeler.html (6 June 2011, date last accessed).

Smith,C.D. *et al.* (2007) Improved repeat identification and masking in dipterans. *Gene*, **389**, 1–9.

Sobreira,T.J. *et al.* (2006) TRAP: automated classification, quantification and annotation of tandemly repeated sequences. *Bioinformatics*, **22**, 361–362.

Song,S. *et al.* (2012) Genome-wide detection of spontaneous chromosomal rearrangements in bacteria. *PLoS One*, **7**, e42639.

Spitzer,M. *et al.* (2004) VisCoSe: visualization and comparison of consensus sequences. *Bioinformatics*, **20**, 433–435.

Supply,P. *et al.* (1997) Identification of novel intergenic repetitive units in a mycobacterial two-component system operon. *Mol. Microbiol.*, **26**, 991–1003.

Trivedi,S. (2006) Comparison of simple sequence repeats in 19 archaea. *Genet. Mol. Res.*, **5**, 741–772.

van Belkum,A. *et al.* (1998) Short-sequence DNA repeats in prokaryotic genomes. *Microbiol. Mol. Biol. Rev.*, **62**, 275–293.

Volfovsky,N. *et al.* (2001) A clustering method for repeat analysis in DNA sequences (RepeatFinder). *Genome Biol.*, **2**. research0027-research0027.11.