BIOINFORMATICS ORIGINAL PAPER Vol. 25 no. 20 2009, pages 2632–2638
doi:10.1093/bioinformatics/btp482

Sequence analysis

T-REKS: identification of Tandem REpeats in sequences with a K-meanS based algorithm

Julien Jorda* and Andrey V. Kajava*

Centre de Recherches de Biochimie Macromoléculaire UMR 5237, CNRS, University of Montpellier 1 and 2, Montpellier, France

Received on March 25, 2009; revised on August 5, 2009; accepted on August 7, 2009

Advance Access publication August 11, 2009

Associate Editor: Ivo Hofacker

ABSTRACT

Motivation: Over the last years a number of evidences have been accumulated about high incidence of tandem repeats in proteins carrying fundamental biological functions and being related to a number of human diseases. At the same time, frequently, protein repeats are strongly degenerated during evolution and, therefore, cannot be easily identified. To solve this problem, several computer programs which were based on different algorithms have been developed. Nevertheless, our tests showed that there is still room for improvement of methods for accurate and rapid detection of tandem repeats in proteins.

Results: We developed a new program called T-REKS for *ab initio* identification of the tandem repeats. It is based on clustering of lengths between identical short strings by using a K-means algorithm. Benchmark of the existing programs and T-REKS on several sequence datasets is presented. Our program being linked to the Protein Repeat DataBase opens the way for large-scale analysis of protein tandem repeats. T-REKS can also be applied to the nucleotide sequences.

Availability: The algorithm has been implemented in JAVA, the program is available upon request at http://bioinfo.montp.cnrs.fr/?r=t-reks. Protein Repeat DataBase generated by using T-REKS is accessible at http://bioinfo.montp.cnrs.fr/?r=repeatDB.

Contact: julien.jorda@crbm.cnrs.fr; andrey.kajava@crbm.cnrs.fr

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

1 INTRODUCTION

An increasing number of complete genome sequences are being generated and deposited into the databases. The next great challenge is to understand the genome data. A significant portion of the proteins carrying fundamental functions contain arrays of tandem repeats (Marcotte *et al*., 1999). Over the last years a number of evidences has been accumulated about the high incidence of tandem repeats in the sequences of virulence factors of pathogenic agents, toxins and allergens (Kajava *et al*., 2006). Furthermore, the tandem repeats frequently occur in amyloidogenic, prion and other disease-related sequences (Baxa *et al*., 2006; Nelson and Eisenberg, 2006). This implies that this class of sequences may have a broader role in human diseases than was previously recognized. Along this line, the discovery of these domains and their structure–function study promise to be a fertile direction for research leading to the identification of targets for new medicaments and vaccines.

A systematic bioinformatics analysis of protein repeats in genomes can provide a global view on these motifs, on their structures, functions and evolution and, in its turn, may result in a significant improvement of our understanding of the biological meanings of the genome sequences. The 'biological' tandem repeats are usually not perfect containing a number of mutations (substitutions, insertions, deletions) accumulated during evolution and some of them cannot be easily identified. Over the last years, several algorithms, software and approaches have been developed (Andrade *et al*., 2000; George and Heringa, 2000; Heger and Holm, 2000; Landau, Schmidt et al., 2001; Kajava and Steven, 2006; Newman and Cooper, 2007) for identification of repeats in biological sequences. Programs such as INTREP (Marcotte *et al*., 1999), RADAR (Heger and Holm, 2000) and TRUST (Szklarczyk and Heringa, 2004) are based on sequence self-alignment (SSA) algorithms. These programs are especially efficient for detection of long repeats (more than ~10 residues long), however, they frequently fail to identify short repeats and do not distinguish between tandem and interspersed repeats. In addition, the SSA algorithms with their time complexity of $O(n^2)$ (where n is the length of sequence), are relatively slow, and, therefore, do not suit well for the large scale analysis. Other type of programs, such as Tandem Repeats Finder (Benson, 1999), XSTREAM (Newman and Cooper, 2007) or MREPS (Kolpakov *et al*., 2003) rely on short string extension algorithm or as STAR (Delgrange and Rivals, 2004) and TRED (Sokol *et al*., 2007) use improved dynamic programming algorithms. They have time complexity lower than SSA algorithms and therefore, are more rapid. Most of these programs are predestined for DNA sequences. XSTREAM is well adapted for a large-scale search of protein repeats, however, it fails to identify some tandem repeats. It is worth also mentioning, approaches that apply sequence profile methods and HMMs (Gribskov *et al*., 1987). These approaches are the best for detection of long imperfect repeats (Kajava *et al*., 1995; Lupas *et al*., 1997). However, they require *a priori* generated alignments of putative repeats and, therefore, are not suitable for automated *ab initio* large scale analysis.

Thus, despite the existence of a number of methods for determination of tandem repeats there is still room for their improvement and for development of an accurate and rapid program

*To whom correspondence should be addressed.

dedicated for the systematic analysis of the repeats in genomes. In this work, we describe a new program T-REKS for protein tandem repeat identification which is based on the analysis of distribution of short strings within the sequence by using *K*-means algorithm. Furthermore, we used output of our program to collect a database of protein tandem repeats. The content of this database is publicly accessible via a web-site. Our objective is to make the results of the systematic bioinformatics analysis available via a regularly updated web-server, which will be a useful tool for scientists interested in structure, function, evolution and application in medicine and technology of proteins with tandem repeats.

2 METHODS

A flowchart of the algorithm is shown on Figure 1.

2.1 Short string probes and *K*-means clustering

To probe an analyzed sequence for the presence of tandem repeats we use short strings (SS) no longer than the repeat length (see example in Fig. 2). For proteins, the size of SS was chosen equal to two because the longer SS were less efficient for detection of two-residue repeats and some degenerous repeats. The search with one-residue probe turned to be less selective. Prior to the application of our algorithm, all homorepeats (tandem repeats of a single residue) were excluded from the analyzed sequences and were registered in our Protein Repeat DataBase for further studies (Fig. 1).

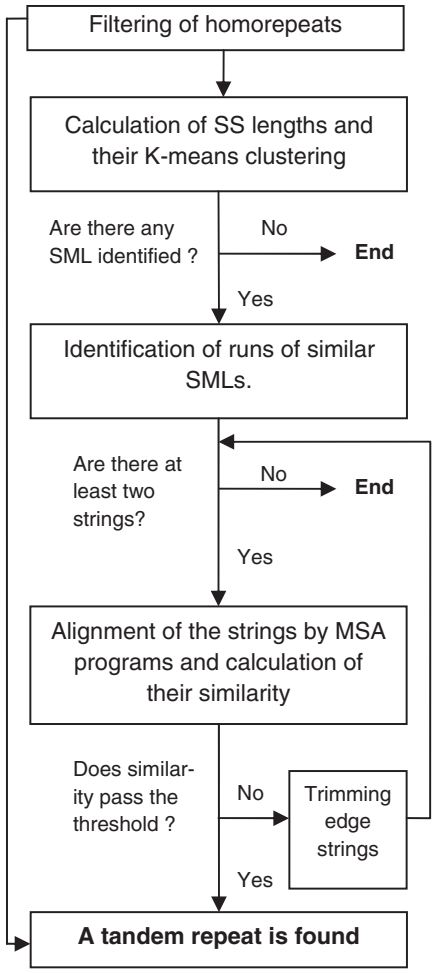

Fig. 1. Flowchart of the T-REKS program applied to a given sequence.

In a tandem repeat region, the most frequently occurred length between identical SSs should be equal to the repeat length. Therefore, detection of regions of an analyzed sequence where certain lengths between identical SSs have anomalously high occurrence may lead to the localization of the tandem repeats. Tandem repeats in the sequences of biological macromolecules (proteins and DNA) have two properties that may hamper the application of this approach. First, the degenerous character of the 'biological' repeats diffuses peaks of the SS length distribution. Second, a given protein or DNA sequence can have several different tandem repeat regions. In this case the same type of SS can be involved in different repeats and calculation of a simple mean of the length occurrences allows to identify only repeats of one length and leaves the remaining repeats unrecognized. To overcome these problems, we use a well-known algorithm for unsupervised classification called *K-means algorithm* (MacQueen, 1967). In our program, the lengths between identical neighbouring SSs were used as datapoints of the K-means algorithm to find potential lengths of the tandem repeats. For example, in Figure 2 the datapoints are lengths 5, 10 and 11 of a short string **EL**. This method partitions all datapoints into *K* clusters for user-defined *K*. For each partition a central datapoint (*centroid)* may be defined. *K* initial centroids are selected from the dataset either randomly, by applying hierarchical clustering or other techniques. Then euclidian distances are calculated between each datapoint and the centroids to assign the datapoint to the cluster which has the nearest centroid.

After this, the positions of the centroids are changed within each of the clusters and this procedure iteratively repeated until the consistency of clusters does not change anymore.

Usually, the *K* starting datapoints are selected randomly. However, it has been demonstrated that different choice of the initial centroids can lead to different results. To solve this problem, we implemented an algorithm which determines the initial centroids based on divisive hierarchical clustering (Johnson, 1967). This algorithm starts from one single cluster which includes all datapoints. This cluster is then iteratively subdivided into smaller clusters based on a rule to maximize the distance between the clusters. This process stops once the number of clusters becomes equal to user-defined *K*. The *K* centroids of these clusters constitute starting points in subsequent clustering by the *K*-means algorithm. Application of this algorithm to protein sequences required adjustment of several parameters. For example, *K* can not be less than the number of different types of tandem repeats in the analyzed sequence.

On the other hand, time complexity of this algorithm is $O(n \cdot K)$ and this favors smaller *K*. Our tests suggested that $K = 10$ gives the most accurate and rapid results for the identification of the protein repeats. The accuracy of this approach is also a function of the length of the analyzed sequence. Statistically, the longer is the sequence the higher is the number of occurences of a given short string. The increase of the occurrences will amplify a background noise and decrease the quality of detection at the clustering steps. Our tests of the algorithm with $K = 10$ shows that this problem appears when protein length is longer than 1500 residues. In this case, our program splits the sequences before and concatenates after the analysis.

2.2 Establishment of tandem repeat lengths

The procedure consists of three steps:

(1) The first step is separately applied to each type of SS found in the analyzed sequence. For example, Figure 2 demonstrates this procedure for a short string **EL**. All lengths between neighbouring **EL**s (5, 10, 11) are considered as datapoints for *K*-means algorithm. Within each cluster generated by this algorithm, we select the most frequent length and call it a Short string Main Length (SML). If a cluster has several most frequent lengths that occur the same number of times, the shortest length is chosen. As a result of this step, *K*

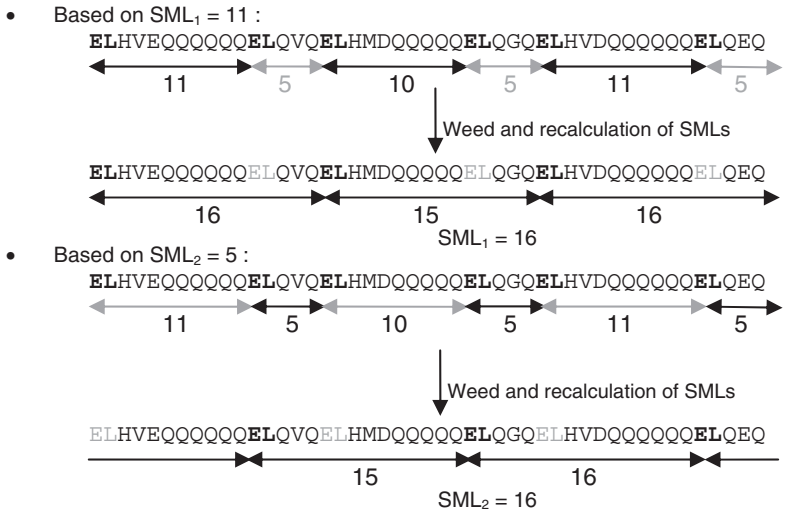

Fig. 2. Example of establishment of SMLs for short string **EL** and $K = 2$.

different SMLs of **EL** are obtained. In Figure 2, $K = 2$ and $SML_1 = 11$; $SML_2 = 5$.

(2) Not all SMLs may correspond to the tandem repeat lengths, because a given short string may occur more than one time within a repeat (Fig. 2). To unmask the real repeat length, we use a procedure of SS 'weed'. First, we consider only **EL**s which are separated by lengths that are equal or close to the SMLs. The threshold of closeness of the length to the SML (Δl) is defined by users as a value which is proportional to the length. This is to take into account the variability of the lengths in biological tandem repeats. Second, we scan the analyzed sequence and do not consider a downstream **EL** of the neighbouring SS except for those which length correspond to one chosen SML (weeded ELs are in grey in Fig. 2). After this, we recalculate lengths and store them. The scanning is repeated for each of the SMLs and leads to K new sets of recalculated lengths. If the most frequent recalculated length differs from the current SML of a given set, it will be considered as a new SML ($SML_1 = SML_2 = 16$ in Fig. 2) and the weed is applied to the next SML. This operation ends when all K SMLs are tested. The original SML remains unchanged when it occurs more frequently than the recalculated lengths or if SMLs and the recalculation lengths are equal to each other. Steps 1 and 2 are performed one by one for all types of SSs of the analyzed sequence.

(3) Finally, K-means algorithm is simultaneous applied to all SMLs of all type of short strings. This operation provides K most frequent SMLs that can be considered as candidates for the real repeat lengths of the analysed sequence.

2.3 Contiguity filtering

An array of tandem repeats is defined as at least two adjacent copies having similar lengths. To take into account the contiguity of the repeats, we select SSs whose *SMLs* are equal or close (within user-defined Δl) to the most frequent SML of a given cluster created after the last step of K-means algorithm. Then, we scan the sequence by considering only these SSs and by looking for sequential repetitions of approximately equal lengths that we define as 'runs' of SMLs. Regions containing these 'runs' may represent tandem repeats. This procedure is redone one by one for all K clusters of SMLs. According to the locations and values of SML found in runs, hypothetical repeats are identified.

2.4 Extension and bridging of runs

Runs with the same SML can be interrupted by a region that is 2 or more times larger than this SML. To clarify the question, whether this region together with two flunking runs belongs to one tandem repeat or not we apply the following procedure. First, we divide the sequence downstream of the run into strings of length equals to the SMLs identified for the current cluster. Then each string joins the preceding run one by one in their order of appearance in the sequence if it contains half or more occurrences of short strings whose SMLs are present in the current cluster. In a special case of two-residue repeats, when the repeat length is equal to the SS length, we consider a string as a part of the run if it has at least one common residue with the SS. We add strings to the preceding runs until we arrive to the next run and by doing so we 'bridge' two runs with the same SMLs. This process starts after each run and stops at any string that has unacceptable SS composition.

2.5 Similarity filtering

The final step of the program is to evaluate the level of sequence similarity between the putative repeats of each run by using Multiple Sequence Alignement (MSA) approaches. For this purpose we used a combination of three MSA programs, a build-in program based on 'center-star algorithm', and two external programs CLUSTALW (Thompson *et al*., 1994) and MUSCLE (Edgar, 2004).

Based on the obtained MSA of the repeats constituting the runs, we deduce a consensus sequence and subsequently use it as a reference for similarity calculation. Let us consider an alignment made by m repeats of length l. In this alignment an indel is considered as an additional 21st type of amino acid residues. We calculate D_i that is a *Hamming distance* (Hamming, 1950) between the consensus sequence and a repeat R_i with $1 \leq i \leq m$. Then, we define a similarity coefficient for the whole alignment as $P_{\text{sim}} = (N - \sum_{i=1}^{m} D_i)/N$ with $N = m \times l$ and $0 \leq P_{sim} \leq 1$. Our program allows to select the runs which have the repeat similarities higher than a user-defined P_{sim} threshold (elsewhere noted as $P*_{\text{sim}}$).

Sometimes, strings located at the extremities of the alignment can diminish the total level of the alignment similarity. To solve this problem, we apply an additional trimming to alignments with similarities not exceeding the $P*_{\text{sim}}$. The operation eliminates the extreme strings one by one until it passes the threshold.

If a run meets all the criteria defined above, it will be considered as a tandem repeat.

2.6 Sequence databases for tests of T-REKS and the other programs

During debugging and tests of T-REKS we used two tandem repeat databases. First, we downloaded from TRIPS website (Katti *et al*., 2000) a database of tandem repeats detected by a sliding window technique and empirical mismatch levels from Swissprot Release of July 1999. The tandem repeats of TRIPS contain only residue substitutions and not indels. We extracted from this database all the data except homorepeats that brought the number of sequences to 890. Second, we generated artificial databanks of 1000 amino acid sequences each of them 1000 residue long. These sequences contained tandem repeats with *a priori* known features. For this, we used a built-in Java linear congruential generator to produce random sequences from an alphabet of 20 amino acids. Then, we inserted one array of perfect tandem repeats in each of these random sequences. The inserted tandem repeats were different between each other having variable repeat lengths (from 2 to 21), number of copies (from 2 to 20). Then we randomly mutated the perfect repeats by substitutions of amino acid residues or by introduction of indels (either insertion or deletion). One original sequence with a perfect tandem repeat yielded a set of sequences with the similarity levels between a user defined P_{sim} value and 1. The generated repeats were then aligned, their similarity level was calculated and if it was over $P*_{\text{sim}}$ the tandem repeat was removed. Following the described procedure, we generated nine databanks with different similarity levels ($P_{\text{sim}} \geq 0.50$; $P_{\text{sim}} \geq 0.55$; $P_{\text{sim}} \geq 0.60$; $P_{\text{sim}} \geq 0.65$; $P_{\text{sim}} \geq 0.70$; $P_{\text{sim}} \geq 0.75$; $P_{\text{sim}} \geq 0.80$; $P_{\text{sim}} \geq 0.85$ and $P_{\text{sim}} \geq 0.90$) and stored them in Generated Repeat Databanks (GRD) from GRD50 to GRD90 correspondingly. Each sequence in the GRDs contains one tandem repeat which is flanked by the randomized sequences. None of the tested programs (see Section 3.2) found tandem repeats within the random sequences of the GRDs. Positive hits were always located within the inserted tandem repeats. Therefore, during the tests, number of the hits was counted as number of sequences with the identified repeats.

To control the level of false positive results, we also generated a databank of 890 random sequences (the same number of sequences and characters as in TRIPS). The first random protein sequence was obtained by using the RandSeq tool from Expasy (Gasteiger *et al*., 2003) with the average amino acid composition of SwissProt. Then, this random sequence has been shuffled by the ShuffleSeq tool from EMBOSS (Rice *et al*., 2000) to obtain the other random sequences. To draw a line between true and false positive results we used the following procedure. Frequently, the random sequences also contain short runs of tandem repeats. We assume that the number of perfect tandem repeats X found by chance in a random sequence database follows a binomial distribution $X \sim B(n,p)$ approximated by a Poisson Distribution with parameter $\lambda = np$ where $p = (1/20)^{l*(m-1)}$, $n = 1.5 \times 10^8$ residues (size of the SwissProt database) and $l \times m$ is the total length of the tandem repeat region. Based on this approximation, occurrences of tandem repeats with $l \times m$ equal or longer than 14 residues is $\lambda = 0.1$ and the probability not to find such repeats $P(X = 0)$ is 0.896 (Fig. 3). Therefore, we considered 14 residues as the minimal length of the true repeat run with potential biological meaning. Within T-REKS, homorepeats are treated differently, so we found it reasonable to fix their minimal length separately. In accordance with our calculations [$\lambda = 0.005$ and $P(X = 0) = 0.994$], it is equal to 9 residues.

3 RESULTS AND DISCUSSION

3.1 Performance of T-REKS depending on its parameters and options

T-REKS was tested against the GRDs having different levels of similarity P_{sim}. Figure 4a shows decrease of the number of the undetected repeats with the increase of the repeat similarity. During this test the allowed similarity threshold $P*_{\text{sim}}$ of T-REKS was set to 0.7. Another test of T-REKS against the random sequence databank by varying its $P*_{\text{sim}}$ threshold showed that the number of false positive results (that are hits of total length more than 14 residues) drastically drops to 0 when $P*_{\text{sim}}$ becomes equal or more than 0.7 (Fig. 4b). We selected $P*_{\text{sim}} \geq 0.7$ as the default value for the web version of T-REKS.

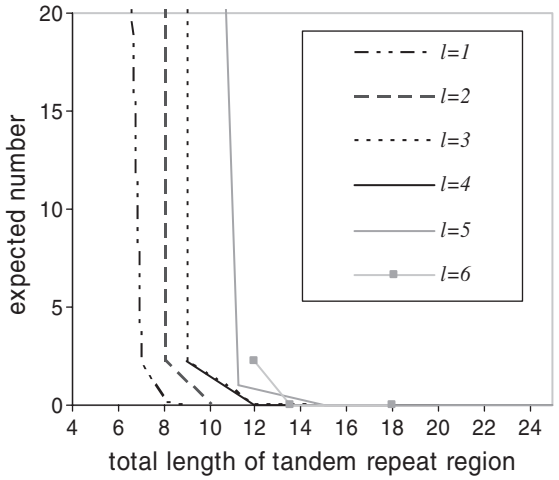

Fig. 3. Distribution of expected number (λ) of tandem repeats with different l found in a random protein sequence of 1.5×10^8 residues depending on total length of the repetitive region.

The test also revealed that at a given $P*_{\text{sim}}$ value of the program a MSA mode which uses one by one external programs CLUSTALW and MUSCLE find the biggest number of tandem repeats.

3.2 T-REKS performance compared with the other programs

Tests of the other existing programs for identification of protein repeats, such as TRED, INTREP and XSTREAM demonstrated that they, similarly to T-REKS (Fig. 4) face difficulties in correct determination of repeats with the decrease of the repeat similarity level. At the same time, all four programs passed successfully the test against the random sequence databank, since no false positives repeats could be detected.

To benchmark T-REKS and these programs we used databank of repeats TRIPS and SwissProt (Release of January, 2009) taken from the repository of NCBI. It is worth mentioning that the comparison of the programs is complicated by differences in the tandem repeat definitions. T-REKS and XSTREAM have the closest definitions. To match the definitions given by these two programs we set the similarity filtering parameters $P*_{\text{sim}}$ of T-REKS and I of XSTREAM to 0.7. The minimal total length of tandem repeat has been set to 14 residues for both T-REKS and XSTREAM. Our tests show that T-REKS finds more tandem repeats in protein sequences than other tested programs (Table 1). (Supplemental Data show examples of the tandem repeats of TRIPS databank that were found only by T-REKS and not by XSTREAM which is the most rapid program with similar definition of the repeats). At the same time, the performance of T-REKS was one of the most rapid. Although XSTREAM is faster than T-REKS, speed is becoming an uncritical issue when both programs need only minutes to analyse one genome. For example, T-REKS needs 17 min to analyse a medium size

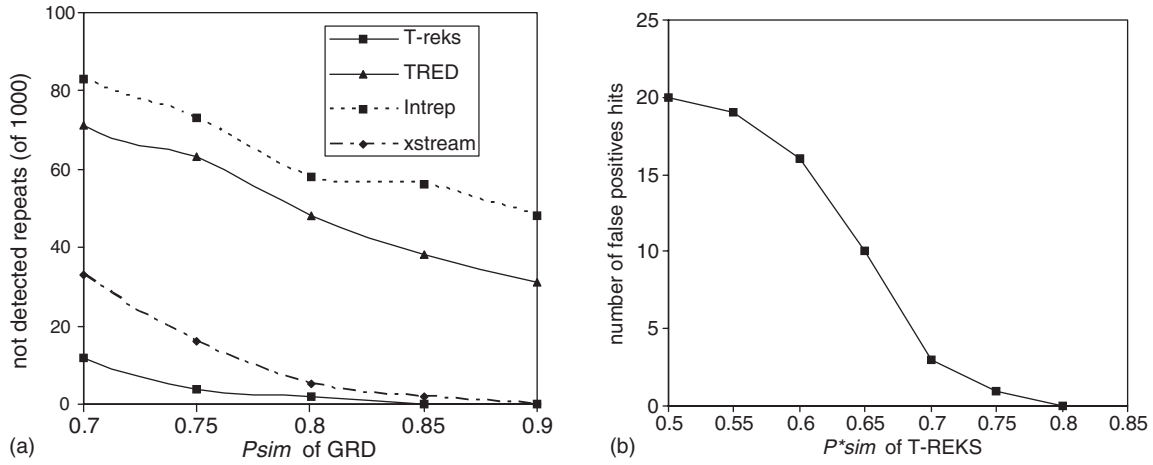

Fig. 4. (**a**) Number of sequences not detected by programs among the 1000 ones of each GRD. $P*_{sim}$ = 0.7 for T-REKS and I = 0.7 for XSTREAM. (**b**) Number of false positive sequences found in the Random sequence databank by T-REKS depending on its $P*_{sim}$ treshold.

Table 1. Benchmark of T-REKS, INTREP, TRED and XSTREAM programs executed on two databanks of protein sequences

	TRIPS (890 sequences with tandem repeats)		SWISSPROT (356 232 sequences)	
	Sequences identified[a]	Execution time	Sequences identified[a]	Execution time
T-REKS[b]	888	4 min	33 780	4 h 30 min
INTREP[c]	863	25 min	20 607[d]	23 h
TRED[e]	865	4 min	15 274	13 h 25 min
XSTREAM[f]	872	50 s	15 204	25 min

Benchmark has been performed with a Personal Computer Pentium 4 3.00 GHz and 2 Gb of RAM.

[a] Sometimes, the number of identified tandem repeats exceeds the number of sequences due to ability of programs to find several tandem repeats in the same sequence.

[b] This work, tandem repeats are defined as having the minimal length of 14 residues (nine residues for homorepeats) and $P_{sim} \geq 0.70$.

[c] Marcotte *et al.* (1999) repeats are defined as having P-value $< 10^{-3}$ (default value).

[d] INTREP results include both tandem and interspersed repeats.

[e] Sokol and Benson (2007) to match with T-REKS, 14 residue minimal length of tandem repeat region was selected, other parameters were set to their default values.

[f] Newman and Cooper (2007) to match the definition of tandem repeats with T-REKS the following parameters were used: $I = i = 0.7$, $minP = 1$, $minC = 2$, $mind = 14$, sort seed length = 2.

genome of *Drosophila melanogaster* (by using a Personal Computer Pentium 4 3.00 GHz and 2 Gb of RAM), while XSTREAM uses 2 min.

3.3 Implementation

T-REKS has been implemented in Java language and has a built-in GUI to allow the user to set parameters needed for the identification of tandem repeats. Although, T-REKS is tuned to explore protein sequences, the same version of the program can also be used for nucleotidic sequences or ones based on another alphabet. A standalone version of the program can be downloaded from our web page (http://bioinfo.montp.cnrs.fr/?r=t-reks). The program can also be used through a web interface at the same webpage. The web version, in addition to the basic features, is adjusted to treat large-scale protein sequence databanks such as Swissprot, NR or PDB taken from the NCBI repository.

Several parameters and options of the program can be defined by users. Among them are: Δl—allowed percentage of length variability (default value, which is fixed in the web version, is equal to 20% of l. It was chosen based on the analysis of the known repeats of biological importance.), $P*_{sim}$—similarity threshold (default value is equal to 0.7) and an option to allow/disallow overlaps of different tandem repeats that can be detected in the same region of a given sequence. In the case of the overlapping tandem repeat regions, priority was given to the longer one with the higher P_{sim}. In the standalone version, it is possible to choose between three MSA modes:

(1) a built-in mode which uses a 'center-star' algorithm,

(2) an external mode that uses two external programs CLUSTALW and MUSCLE one after another or

(3) a hybrid mode which uses in sequential order the 'center-star', CLUSTALW and MUSCLE programs.

The build-in MSA mode 1 was developed to make the standalone version of T-REKS more convenient for downloading. The combination of CLUSTALW and MUCLE yields the best alignment results and is the only mode available in the web version to favour accuracy at the expense of rapidity. This MSA mode 2 is applied to obtain the most reliable results for our protein repeat database called PRDB (see next section). The hybrid mode 3 represents the most optimal version in terms of accuracy and rapidity.

3.4 Protein Repeats DataBase (PRDB)

T-REKS output is used to fill our PRDB. A pilot version of PRDB can be found on our website http://bioinfo.montp.cnrs.fr/?r=repeatDB. At this moment, the PRDB contains about 1105 entries of tandem repeats from PDB, 50 789 from Swissprot. The information about the identified repeats is displayed in a table with characteristics such as organism, repeat length, number of copies, level of the repeat similarity, consensus sequence, position in the sequence, subcellular

Table 2. Comparison of repeats found by our program and Tandem Repeats Finder in the Human Frataxin gene intron 1

T-REKS[a]/TRF[b]			
Start	End	Copy length	Copy number
163/–	**188/–**	**12/ –**	**2/ –**
822/822	856/854	7/14	5/2.4
1786/1787	1912/1874	44/44	3/2
2167/–	**2184/–**	**1/–**	**18/–**
2185/2183	2210/2211	3/3	9/9.7
2387/–	**2410/–**	**6/–**	**3/–**

Additional repeats identified by T-REKS are indicated in bold.
[a]This work.
[b]Benson (1999).

localization and complete alignment of the copies. It allows users to choose repeats based on their organism, the consensus pattern, amino acid composition, tendency to be unstructured and the other parameters. We plan to improve, regularly update and maintain this database.

3.5 Application of T-REKS to the identification of tandem repeats in DNA sequences

T-REKS can, in principle, be also applied to nucleotide sequences because its algorithm is not linked to any particular alphabet. However, depending on the number of the characters in the alphabet, T-REKS may require optimization of certain parameters. The probability to find tandem repeats by chance in DNA sequences is higher than in protein sequences with correspondingly 4- and 20-letter alphabet. For example, comparing random 'Bernoulli' DNA and protein sequences of the same size, the expected (mean) number of repeats of l-residue length, copied m-times in DNA is $(20/4)^{*}l^{*}(m-1)$ times higher than in proteins. Thus, DNA sequences have a higher background noise in the distribution of the SS lengths and this can hamper the tandem repeats detection. To improve performance of T-REKS on DNA sequences we increase the length of SS to four residues. This modification reduced the gap between the expected values of tandem repeats found by chance in proteins and DNA. In addition, the number of clusters, K, was increased from 10 (for proteins) to 20 (for DNA). Similarly to protein sequences, T-REKS splits the DNA sequences that are longer than 1500 residues before and concatenates after the analysis. After these modifications, T-REKS yields better results on DNA sequences than its version with protein-specific parameters. For example, our tests of T-REKS on the Human Frataxin Gene (Friedreich's ataxia) intron 1 sequence, showed that it detects not only tandem repeats previously obtained by Tandem Repeats Finder (TRF) (Benson, 1999) but also some additional repeats (Table 2). Thus, T-REKS can also be used for detection of tandem repeats in DNA sequence databases. We are aware that DNA sequences of 1500 nucleotides may, in principle, contain more than 20 different lengths of tandem repeats (maximum number of clusters at $K = 20$) and that the splitting step may lead to the failure to detect some repeats. More accurate tests and optimization of the T-REKS parameters for DNA is a subject of our further studies.

4 CONCLUSIONS

In this article, we described a new program for *ab initio* identification of tandem repeats in protein sequences called T-REKS. It is based on K-means clustering of putative lengths of tandem repeats. T-REKS finds more tandem repeats in protein sequences than other tested programs. At the same time, it demonstrates one of the most rapid performances. Thus, this approach is well-suited for large scale analysis of tandem repeats.

T-REKS has been developped in a dual mode: a standalone mode with a user-friendly graphical interface for local use, and a Web-interface version. The latter version of the program is connected to our webserver and is able to store the results in a database of protein tandem repeats called PRDB. Both versions of T-REKS and the database PRDB are available to public via our webpage at http://bioinfo.montp.cnrs.fr/?r=t-reks. We plan to use this database for systematic large scale analysis of protein tandem repeats in genomes in order to obtain a global view on the structure, function and evolution of these motifs. T-REKS can be also used for detection of tandem repeats in DNA. Its further optimization for DNA sequences will be a subject of our future studies.

ACKNOWLEDGEMENTS

The authors thank Dr M. Anisimova, Dr J. Arunachalam and Dr S.A. Kondratov for critical reading of the manuscript and suggestions.

Funding: Ministère de l'Education Nationale, de la Recherche et de la Technologie (MENRT) grant to J.J.

Conflict of Interest: none declared.

REFERENCES

Andrade,M.A. *et al*. (2000) Homology-based method for identification of protein repeats using statistical significance estimates. *J. Mol. Biol.,* **298**, 521–537.

Baxa,U., *et al*. (2006) Structure, function, and amyloidogenesis of fungal prions: filament polymorphism and prion variants. *Adv. Protein Chem.*, **73**, 125–180.

Benson,G. (1999) Tandem repeats finder: a program to analyze DNA sequences. *Nucleic Acids Res*., **27**, 573–580.

Delgrange,O. and Rivals,E. (2004) STAR: an algorithm to search for Tandem Approximate Repeats. *Bioinformatics*, **20**, 2812–2820.

Edgar,R.C. (2004) MUSCLE: multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Res.,* **32**, 1792–1797.

Gasteiger,E. *et al*. (2003) ExPASy: The proteomics server for in-depth protein knowledge and analysis. *Nucleic Acids Res*., **31**, 3784–3788.

George,R.A. and Heringa,J. (2000) The REPRO server: finding protein internal sequence repeats through the Web. *Trends Biochem Sci*., **25**, 515–517.

Gribskov,M. *et al*. (1987) Profile analysis: detection of distantly related proteins. *Proc. Natl Acad. Sci. USA*, **84**, 4355–4358.

Hamming,R. (1950) Error detecting and error correcting codes. *Bell System Technical J.*, **29**, 147–160.

Heger,A. and Holm,L. (2000) Rapid automatic detection and alignment of repeats in protein sequences. *Proteins*, **41**, 224–237.

Johnson,S. (1967) Hierarchical clustering schemes. *Psychometrika*, **32**, 241–241.

Kajava,A.V. *et al*. (2006) Beta-structures in fibrous proteins. *Adv. Protein Chem.*, **73**, 1–15.

Kajava,A.V. and Steven,A.C. (2006) The turn of the screw: variations of the abundant beta-solenoid motif in passenger domains of Type V secretory proteins. *J. Struct. Biol.*, **155**, 306–315.

Kajava,A.V. *et al*. (1995) Modeling of the three-dimensional structure of proteins with the typical leucine-rich repeats. *Structure*, **3**, 867–877.

Katti,M.V. *et al*. (2000) Amino acid repeat patterns in protein sequences: their diversity and structural-functional implications. *Protein Sci*., **9**, 1203–1209.

Kolpakov,R. *et al*. (2003) mreps: efficient and flexible detection of tandem repeats in DNA. *Nucleic Acids Res*., **31**, 3672–3678.

Landau,G.M. *et al*. (2001) An algorithm for approximate tandem repeats. *J. Comput. Biol.*, **8**, 1–18.

Lupas,A. *et al*. (1997) A repetitive sequence in subunits of the 26S proteasome and 20S cyclosome (anaphase-promoting complex). *Trends Biochem Sci*., **22**, 195–196.

MacQueen (1967) Some methods for classification and analysis of multivariate observations. *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*. University of California Press, Berkeley.

Marcotte,E.M. *et al*. (1999) A census of protein repeats. *J. Mol. Biol*., **293**, 151–160.

Nelson,R. and Eisenberg,D. (2006) Structural models of amyloid-like fibrils. *Adv. Protein Chem*., **73**, 235–282.

Newman,A.M. and Cooper,J.B. (2007) XSTREAM: a practical algorithm for identification and architecture modeling of tandem repeats in protein sequences. *BMC Bioinformatics*, **8**, 382.

Rice,P. *et al*. (2000) EMBOSS: the European Molecular Biology Open Software Suite. *Trends Genet*., **16**, 276–277.

Sokol,D. *et al*. (2007) Tandem repeats over the edit distance. *Bioinformatics*, **23**, e30–e35.

Szklarczyk,R. and Heringa,J. (2004) Tracking repeats using significance and transitivity. *Bioinformatics*, **20**(Suppl. 1), i311–i317.

Thompson,J.D. *et al*. (1994) CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Res*., **22**, 4673–4680.