Annals of West University of Timişoara, ser. Biology, 2017, vol. 20 (2), pp.159-166

A REVIEW OF THE CURRENT METHODS FOR COMPUTATIONAL ANALYSIS OF TANDEM REPEATS

Done STOJANOV*, Aleksandra MILEVA
Faculty of Computer Science, University Goce Delcev, Krste Misirkov nn – Štip, Republic of Macedonia
**Corresponding author e-mail: done.stojanov@ugd.edu.mk*
Received 31 August 2017; accepted 4 December 2017

ABSTRACT
This paper considers some of the most important methods for computational tandem repeat analysis. The problem of repeats analysis is far from trivial due to the fact that tandems tend to be highly polymorphic motifs, i.e. or types of mutations within repeats has to be considered. The computational analysis of all types of mutations within repeats increases the time of execution, especially if chromosomes or whole genomes are subject of an analysis. On the other the time complexity significantly improves if only exact tandem repeats are considered, but this has less practical application. There are pros and cons of the methods being considered and maybe the most suitable solutions is a compromise of the opposed conceptions.
KEY WORDS: *tandem repeats, computational, ETR, ATR, TRF, STAR.*

INTRODUCTION

A part of the genomic material consists of motifs of repeated DNA patterns. It is assumed that 10% of the human genome is consisted of short repeating sequences. Usually repeated DNA sequences are part of non-coding DNA, but they can also be found in coding DNA regions in form of repeated codon (three nucleotide repetitions). As far as known, the excessive one-codon repetition in some human genes has been associated with disorders, such as: huntington's disease, spinocerebellar ataxia and dentatorubropallidoluysian atrophy. Depending of the length of the repeating pattern there are three types of tandem repeats: microsatellites (the length of the repeating pattern is less than 6 bp), mini-satellites (the length of the repeating pattern ranges from 7 to 100 bp) and satellites (longer than 100 bp).

Simple Sequence Repeats (SSRs) or Microsatellites are patterns of repetitions of mono (one), di (two), tri (three), four (tetra) or penta (five) nucleotides. Since microsatellites are polymorphic, i.e. they have higher mutation rate than other regions (the number of repetitions varies from one individual to other, but not the structure of the pattern that is repeated), they are most commonly used as genetic markers. In spite of the variability of the number of repetitions, some types of microsatellites are found more frequently than others. According to the study performed by Gabor, Zoltan and Jerzy (Tóth *et al.,* 2000), the most frequent tetra SSRs in the human chromosome 22 are: AAAT, AAAG and AAAC, while the most frequent penta SSRs found in the second smallest human chromosome are: AAAAC and AAAAT. In terms of the length of the repetition per mebgabase, they found that: AAAT is the longest repeating tetranucleotide SSRs (378 nucleotides per chromosome megabase), while AAAAC is the longest repeating pentanucleotide SSRs (285 bases per chromosome magabase). The same types of SSRs can be found in other species, but the number of repetitions differs.

This paper considers conceptual frameworks, complexities and all pros and cons of computational methods for microsatellite and tandem repeats identification. Many of them such as: Tandem Repeats Finder (TFR) (Benson, 1999) and STAR: an algorithm to Search for Tandem Approximate Repeats (Delgrange *et al.,* 2004) have an on-line available web implementations based on the client – server model that allows an identification and an analysis of tandem repeats in DNA data submitted in FASTA format by the scientific community.

MATERIALS AND METHODS

Pioneering algorithms developed in this field are typical straight forward implementation of computer science logic, with less practical application in biology, searching for two string duplications or exact tandem repeats (ETR). In this group of algorithms we can enumerate: Main *et al.* (1984), Kolpakov *et al.* (1999), Stoye *et al*. (2002).

The algorithm of Main and Lorentz (Main *et al.,* 1984) identifies all duplications of substrings (xx-duplicated substrings, such as: AAATAAAT, AAAGAAAG) in $O(nlogn)$ time, where n is the length of the sequence. Kolpakov and Kucherov also proposed linear time algorithm, which is able to find maximal repeated substrings (Kolpakov *et al.,* 1999). Maximal repeated substring is a substring that repeats, but it is not a part of any longer repeating pattern.

Suffix tree is the most common data structure employed for this purpose. Suffix trees are memory and time efficient. They require linear space (memory) and they are constructed and searched in linear time (time proportional to the length of the sequence). These features allow an efficient analysis of long genetic sequences, such as whole genomes. Stoye and Gusfield discuss how suffix tress can be applied in order to find all repeating patterns (Stoye *et al.,* 2002). The proposed algorithm identifies branching and primitive tandem repeats in $O(nlogn)$ time and all tandem repeats in $O(nlogn+z)$ time, where: z is the number of occurrences. Once we have constructed the suffix tree, internal nodes correspond the repeating substrings, while leafs (leafs' numbers) correspond to the positions where repetitions occur.

We will discuss the application of suffix tree for this purpose on the short sequence s: GAGAG. First all suffixes in s are extracted: {GAGAG$(1), AGAG$(2), GAG$(3), AG$(4), G$(5), $(6)}. The symbol $ means end of the string and the number in parenthesis (n) identifies the staring position of the suffix in s. Suffixes: GAGAG$(1), GAG$(3) and G$(5) have common staring nucleotide (G) that corresponds to G–labeled edge in the suffix tree Fig. 1. GAGAG$(1) and GAG$(3) have in common the second and the third nucleotide (A and G) that correspond to A and G-labeled edges prior AG$(1) and $(3) branching, Fig. 1. Suffixes AGAG$(2) and AG$(4) have the first two elements in common (A and G) that correspond to A and G-labeled edges prior AG$(2) and $(4) branching, Fig. 1.

Leafs numbered as 2 and 4 follow (A) Adenine and Adenine Guanine (AG) labeled edges that means there are A and AG repetitions at positions 2 and 4, Fig. 1. On the other hand, leafs following G labeled edge are 1, 3 and 5 that means there is G repetition at the previous positions. Leafs following GA and GAG are 1 and 3 that means that there are GA and GAG repetitions at starting positions 1 and 3, Fig. 1.

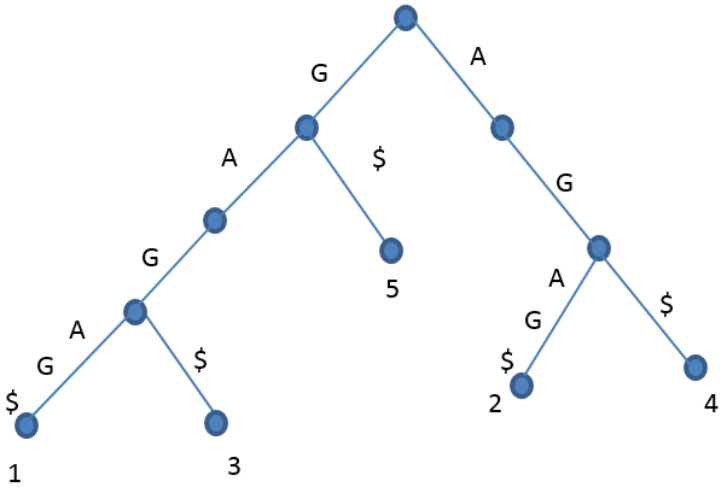

FIG. 1. Suffix tree for the sequence: GAGAG

There are algorithms, such as (Kolpakov *et al.,* 2003), searching for adjacent copies of a pattern that may mismatch, with more practical implementation to biology. On Fig. 2a) the second copy AAAT differs from the previous AGAT at position 2. The hamming distance between these copies is the number of substitutions (mismatching elements) and in this case it equals 1. Applying these approach results in good time complexity, since un-gapped alignment is performed, but (insertion/deletions within repeats are not considered). Using Hamming distance or measuring the number of mismatching nucleotides between two adjacent copies, the second algorithm proposed by Kolpakov and Kucherov (Kolpakov *et al.,* 2003) is able to detect tandem repeats with k possible substitutions in $O(nklogk + S)$ time where S is the size of the output.

On the other hand, (Landau *et al.,* 2001) proposed an upgraded algorithm which is not only able to detect adjacent copies that mismatch at most k elements, but it can also detect copies for which the edit distance is at most k. The edit distance is defined as a number of operations (deletions or insertion or substitution) in order to convert one pattern into the other. For instance the edit distance for p1: AAAT and p2: AAT equals 1, since p1 can be transformed in p2 by deleting nucleotide at position 2 in p1 or p2 can be transformed in p1 by inserting A (Adenine) just after the first nucleotide in p2, Fig. 2b). Applying Landau's algorithm copies with at most k substitutions are identified in $O(nklog(\frac{n}{k}))$ time and those for which the edit distance is at most k are found in $O(nklog(k)log(\frac{n}{k}))$ time.

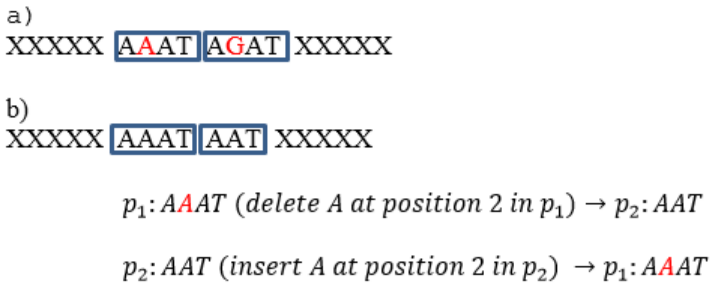

FIG. 2. Polymorphic repeats

The issue of the polymorphic nature of tandem repeats and the difficulty of their computational identification has been also considered by: (Rivals *et al.,* 1997), (Sagot *et al.,* 1998) and Tandem Repeat Finder (Benson, 1999, 2005).

The method of (Rivals *et al.,* 1997) is a compression algorithm that considers insertions among two repetitions of a motif in an approximate tandem repeat and its main drawback is that is limited to relatively small motifs of length less than 4 bp. On the other hand, Sagot *et al*. (1998) detects all type of tandem repeats (micro, mini and satellite repeats). This algorithm runs in two phases. In the first phase all regions that are unlike to contain satellite data are eliminated, while in the second phase the rest of the space is explored for all possible models of repeating units. Due to its combinatorial nature (computation of different combination of likely motifs and alignments for approximate tandem repeat) this is a time-exhausting algorithm.

Tandem Repeats Finder (TRF): a program to analyze DNA sequences (Benson, 1999, 2005) is an implementation of an algorithm, which is able to detect DNA repeats without have to prior specify the structure of the pattern and its length.

At first, Benson's program searches for exact repeated motifs, such as for relatively small integer k, a set of 4^k different strings (also referred as probes) over the DNA alphabet $\sum = \{A, C, T, G\}$ is generated. By sliding each probe through the sequence, each position i where the probe occurs is subtracted from position j of a previous occurrence of the probe, i.e. the distance $d = i - j$ is calculated.

By computing the distance d and applying statistical test, based on: the size of the probe, the number of repetitions of the pattern and the probability of insertion/deletion, tandem repeats are localized. In order to report a tandem repeat, the score of the alignment of the region that contains ETRs (exact tandem repeats) must be greater than user defined threshold.

Benson's program is freely accessible at: https://tandem.bu.edu/trf/trf.html and it allows the user to submit sequences in FASTA format for repeats' analysis, Fig. 3. The results of the analysis are summarized into a report that contains: the repeating pattern, the number of repetitions, and the indices of repetition. If using the advanced version: https://tandem.bu.edu/trf/trf.advanced.submit.html the output depends of user-specified parameters.

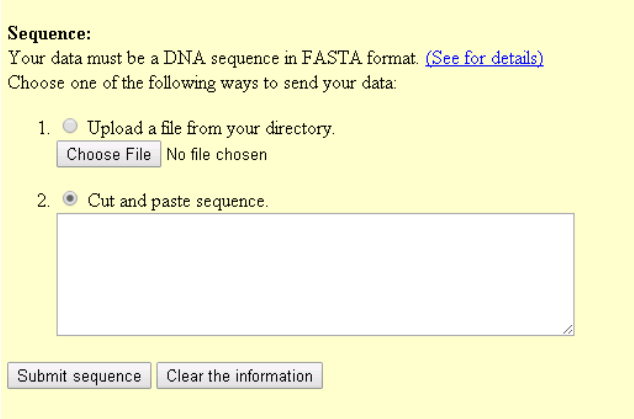

FIG. 3. Tandem Repeats Finder interface

As similar or upgraded TRF implementations can be considered: Adplot (Taneda, 2004), VNTRfinder (O'Dushlaine *et al.,* 2006) and TRAP (Sobreira *et al.,* 2006). Adplot (Auto Dot PLOT) can visualize approximate local repeats, while VNTRfinder analyzes variation in the length between arrays of tandem repeats. TRAP is implemented in PERL and its main advantage is that output results are outputted according user's requirement. This means that identified tandem repeats are selected, quantified and stored according user's preference.

Since results generated by tandem repeats finder depend of the user-defined threshold and the length of the motif, STAR: an algorithm to Search for Tandem Approximate Repeats (Delgrange *et al.,* 2004) detects approximate tandem repeats regardless any user-defined threshold. This algorithm is able to detect duplications of a motif with point mutations included in., by employing a criteria called minimum description length that computes the number of point mutations allowed per approximate tandem repeat compared to the exact tandem repeat of the best possible size.

Just as Tandem Repeats Finder, STAR also employs Wraparound Dynamic Programming (WDP) (Fischetti *et al.,* 1993) and its computations complexity is $O(n \times |m|)$, such as: n is the length of the sequence and $|m|$ is the length of the motif.

When these algorithms are compared in terms of sensitivity and efficiency, STAR is more sensitive than TRF (it can detect approximate tandem repeats which TRF can't), but it is less effective. STAR is also available online at: http://www.atgc-montpellier.fr/star/, Fig. 4 and just as TRF sequences in FASTA format are accepted for analysis, but the results are returned to the user via mail.

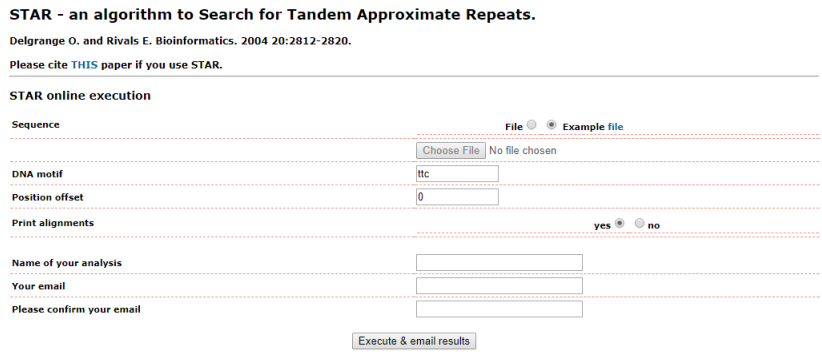

FIG. 4. STAR user interface

In order to facilitate and speed up the process, (Reneker *et al.,* 2004) employ hash function. Web server implementation which is available at: http://acmes.rnet.missouri.edu allows search for exact hits of query sequences ranging between 3 and 10 kbp.

In the preprocessing stage, short DNA words of k nucleotides are hashed (converted) into integers employing hash function (equation (1), (2)) and stored into indexed file on the hard drive. Prior the search the query is also converted into integer (key) and during the search process only those pages that contain the corresponding has bin are retrieved in the memory. This approach can be used to identify dispersed or tandem repeats, but its main drawback is that the pattern must be specified in advance.

$$f(A) \to 0, f(C) \to 1, f(G) \to 2, f(T) \to 3 \quad (1)$$

$$f(w: b_1 b_2 \dots b_{k-1} b_k) = \sum_{i=1}^{k} f(b_i) \times 4^{i-1} \quad (2)$$

The model of evaluative tandem repeats has been also exploited for this purpose. This model (Groult *et al.,* 2014), (Hammock *et al.,* 2015) assumes that each copy of a repeat within tandem repeat resembles its precursor and if the succeeding copy is a part of the tandem repeat this must be also satisfied. Sokol (Sokol *et al.,* 2001) addressed the problem of *k*-edit repeat. Theoretically *k*-edit repeat is a DNA substring such as the edit distance (the number of mismatches or insertions or deletions) between two consecutive copies does not exceed *k*. Furthermore the *k*-edit repeat is maximal if can't be further extended to the left. Sokol (Sokol *et al.,* 2001) solved this problem, but in order to improve the time complexity, the straight forward application of dynamic programming for computing the edit distance between two consecutive copies has been avoided, that results in $O(nk \log k \log(n/k))$ time complexity, such as n is the length of the DNA sequence. The sequence shown on Fig.5 is a typical example for 1-edit repeat. The second copy (AATC) mismatches the previous (AAGC) at position 3, while the third copy (AAC) can be obtained from the previous AATC by deleting the nucleotide T at position 3.

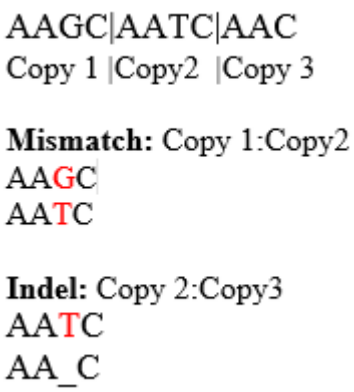

FIG. 5. 1-edit repeat

RESULTS AND DISCUSSIONS

Table 1 summarizes the key features of the methods and algorithms being discussed in the previous section.

TABLE 1. Key features of the algorithms for tandem repeats analysis

Algorithm	Year	Key features
Main et al.,	1984	*Suitable for identification of exact tandem repeats, does not considers mutations within repeats*
Rivals et al.,	1997	*Applicable only for microsatellite identification (<4 base pairs)*
Sagot et al.,	1998	*Applicable for all types of tandem repeats, unfavorable time complexity due to its combinatorial nature*
Kolpakov et al.,	1999	*Detects maximal repeated substrings*
Tandem Repeat Finder (Benson)	1999	*Detects tandems of k-long copies, output depends on user specified threshold*
Landau et al.,	2001	*Considers mutations within copies, at most k mutations per copy*
Sokol et al.,	2001	*Detects tandems based on a solution of the k-edit repeat problem without using dynamic programming*
Kolpakov et al.,	2003	*Mismatching elements are only considered in repeats*
Stoye et al.,	2002	*Employs suffix tree data structure for repeats analysis*
STAR (Delgrange et al.,)	2004	*Detects tandems regardless user specified threshold, less sensitive than Tandem Repeat Finder*
Reneker et al.,	2004	*Fast search due to hash function, main drawback is that the template must be specified in advance*

CONCLUSION

Some of the current and most commonly used computational methods for tandem repeats analysis have been considered. Considered methods mutually differ in terms of time complexity and conceptual framework. Methods that identify exact tandem repeats or repeats that include mismatches only, have favorable time complexity due to the fact that indels within repeats are not considered. On the other hand, methods that consider all types of mutations

within repeats have worse time complexity than the previous, but are more suitable for practical analysis due to the fact that tandem repeats tend to be more polymorphic than other DNA motifs.

REFERENCES

- Benson G.1999.Tandem repeats finder: a program to analyze DNA sequences. *Nucleic acids research27*(2):573-580.
- Benson G. 2005.Tandem cyclic alignment. *Discrete applied mathematics*146(2):124-133.
- Delgrange O., Rivals E. 2004. STAR: an algorithm to search for tandem approximate repeats. *Bioinformatics.* 20 (16):2812-2820.
- Fischetti VA., Landau GM., Sellers PH., Schmidt JP.1993. Identifying periodic occurences of a template with applications to protein structure. *Information Processing Letters.* 45(1):11-18.
- Groult R., Léonard M., Mouchard L. 2004. Speeding up the detection of evolutive tandem repeats. *Theoretical computer science.* 310:309-328.
- Hammock EA., Young LJ. 2005. Microsatellite instability generates diversity in brain and sociobehavioral traits. *Science.* 308(5728):1630-1634.
- Kolpakov R., Kucherov G. 1999. Finding maximal repetitions in a word in linear time, pp. 596-604. In: Foundations of Computer Science, 40th Annual Symposium on. IEEE.
- Kolpakov R., Kucherov G. 2003. Finding approximate repetitions under Hamming distance. *Theoretical Computer Science*303(1):135-156.
- Landau GM., Schmidt JP., Sokol D. 2001. An algorithm for approximate tandem repeats. *Journal of Computational Biology*8(1):1-18.
- Main MG., Lorentz RJ. 1984. An O(nlogn) algorithm for finding all repetitions in a string. *Journal of Algorithms*5(3):422-432.
- O'Dushlaine C., Shields DC. 2006.Tools for the identification of variable and potentially variable tandem repeats. *BMC genomics*7(1):290.
- Rivals E., Delgrange O., Delahaye JP., Dauchet M., Delorme MO., Hénaut A., Ollivier E. 1997. Detection of significant patterns by compression algorithms: the case of approximate tandem repeats in DNA sequences. *Bioinformatics.* 13(2):131-136.
- Reneker J., Shyu CR., Zeng P., Polacco JC., Gassmann W. 2004. ACMES: fast multiple-genome searches for short repeat sequences with concurrent cross-species information retrieval. *Nucleic acids research* 32(suppl_2):W649-W653.
- Sagot MF., Myers EW.1998. Identifying satellites and periodic repetitions in biological sequences. *Journal of Computational Biology.* 5(3):539-553.
- Stoye J., Gusfield D. 2002. Simple and flexible detection of contiguous repeats using a suffix tree. *Theoretical Computer Science.* 270(1-2):843-56.
- Sobreira TJ., Durham AM., Gruber A. 2005. TRAP: automated classification, quantification and annotation of tandemly repeated sequences. *Bioinformatics.* 22(3):361-362.
- Sokol D., Benson G., Tojeira J. 2007. Tandem repeats over the edit distance. *Bioinformatics*23(2):e30-e35.
- Taneda A. 2004. Adplot: detection and visualization of repetitive patterns in complete genomes. *Bioinformatics.* 20(5):701-708.
- Tóth G., Gáspári Z., Jurka J. 2000. Microsatellites in different eukaryotic genomes: survey and analysis. *Genome research.* 10(7):967-981.