TRStalker: an efficient heuristic for finding fuzzy tandem repeats

Marco Pellegrini¹,*, M. Elena Renda¹ and Alessio Vecchio²

¹CNR, Istituto di Informatica e Telematica, Via Moruzzi 1, 56124 Pisa and ²Univ. di Pisa, Dip. Ingegneria dell’Informazione, Via Ditosalvi 2, 56122 Pisa (Italy)

ABSTRACT

Motivation: Genomes in higher eukaryotic organisms contain a substantial amount of repeated sequences. Tandem Repeats (TRs) constitute a large class of repetitive sequences that are originated via phenomena such as replication slippage and are characterized by close spatial contiguity. They play an important role in several molecular regulatory mechanisms, and also in several diseases (e.g. in the group of trinucleotide repeat disorders). While for TRs with a low or medium level of divergence the current methods are rather effective, the problem of detecting TRs with higher divergence (fuzzy TRs) is still open. The detection of fuzzy TRs is preparadect to enrich our view of their role in molecular regulatory mechanisms and diseases. Fuzzy TRs are also important as tools to shed light on the evolutionary history of the genome, where higher divergence correlates with more remote duplication events.

Results: We have developed an algorithm (christened TRStalker) with the aim of detecting efficiently TRs that are hard to detect because of their inherent fuzziness, due to high levels of base substitutions, insertions and deletions. To attain this goal, we developed heuristics to solve a Steiner version of the problem for which the fuzziness is measured with respect to a motif string not necessarily present in the input string. This problem is akin to the “generalized median string” that is known to be an NP-hard problem. Experiments with both synthetic and biological sequences demonstrate that our method performs better than current state of the art for fuzzy TRs and that the fuzzy TRs of the type we detect are indeed present in important biological sequences.

Availability: TRStalker will be integrated in the web-based TRs Discovery Service (TREdS) at bioalgo.iit.cnr.it.

Contact: marco.pellegrini@iit.cnr.it

Supplementary information: Supplementary data are available at Bioinformatics online.

1 INTRODUCTION

Tandem Repeats (TRs) are multiple (two or more) duplications of substrings in the DNA that occur contiguousy, and may involve some base mutations (such as substitutions, insertions and deletions). TRs of several forms (satellites, microsatellites, minisatellites and others) have been studied extensively because of their role in several biological processes. In fact, TRs are privileged targets in activities such as fingerprinting or tracing the evolution of populations (Kelkar et al., 2008; Vogler et al., 2006). Several diseases, disorders and addictive behaviors are linked to specific TR loci (Wooster et al., 2005) and in relation to gene functions (Legendre et al., 2007). Large scale comparative studies on TRs of the human genome are described in Ames et al. (2008) and Warburton et al. (2008). Data Bases of repetitive elements such as RepBase (Jurka et al., 2005) and Tandem Repeats Database (TRDB) (Gelfand et al., 2007) are now available; and the detection of repetitive elements via library-based similarity matching, for example by using the tool Repeatmasker (Smit et al., 2004), is a popular practice. However, tools for ab initio detection of repetitive elements that are not based on prior knowledge accumulated in data bases are still important in order to extend our comprehension of the role of TRs in biological mechanisms. Existing ab initio tools are successful when the TR exhibits a low or medium level of divergence and when the TR is easily validated. However, there is an emerging need for new tools that are able to cope with higher levels of sequence divergence and/or TR computationally more difficult to validate. For example, Boeva et al. (2006) study so called Fuzzy TRs and their role in gene expression. The technique in Boeva et al. (2006) works well for the Hamming metric (only substitutions and no insertions/deletions allowed) and for short repeat units (from 3 to 24 bp) that are common in micro- and mini-satellite families.

Some of the most successful ab initio tools, such as TRF (Benson, 1999) and ATRHunter (Wexler et al., 2005), are based on a multi-stage filtering approach [see also (Peterlongo et al., 2009)]. In the first stage the input sequence is analyzed to detect, via statistical criteria, likely position and length of candidate subsequences. The final stage is the validation one in which a more expensive test is applied to candidate substrings passing the first stages, so to determine an output that matches the implicit definition of TR and the user-defined filtering parameters.

1.1 Our contribution

Our contribution is a novel multi-stage filtering algorithm, called TRStalker, for finding long fuzzy TRs under the edit distance, that introduces new techniques (w.r.t. previous TR finding algorithms) in all stages. For the first stage, where over-represented distances between probes are sought, we employ gapped q-grams (Burkhardt and Karkkainen, 2005) in place of the standard ungapped q-grams in order to collect evidence on the candidate substrings. Gapped q-grams have been used before in the context of textual and biological database searching, but less so in the area of TRs detection [with the exception of the system TEIRESIAS (Stohlovsky et al., 1999)]. Because of errors due to insertion/deletions, the period of a TR is subject to fluctuations, thus we employ a weighting scheme with exponential decay so to reinforce the signal even in presence of this smearing effect. Finally, we use ranking instead of thresholds when deciding the substrings to pass to the next phases, in order to concentrate the computational effort on the zones with candidates with higher weight. For the final validation stage we employ an NP-complete definition of TR involving the concept of generalized median string under edit distance (de la Higuera and Casacuberta, 2000; Sim and Park, 2003), together with an efficient heuristic for computing an approximation of such median string (Jiang et al., 2003) previously not used in a biological context.
By extensive experimental comparisons of TRStalker with two state-of-the-art tools, namely TRF and ATRHunter, we did find out that TRStalker has consistently better performance for a large range of error and length parameters for the class of fuzzy TRs under edit distance, with a recall ranging from 100 to 60%. Thus TRStalker improves the capability of TR detection for classes of TRs for which existing methods do not perform well. Tests performed on standard evolutionary TRs definitions (verifiable in polynomial time) also show recall performance close to 100%. Incidentally, this result confirms of the power of the new techniques developed for the initial filtering phase.

1.2 State of the art

We will briefly survey the state of the art in finding tandem repeats. First we will describe methods that for a given definition of TR are able to find all maximal substrings in the input that match the definition (exhaustive algorithms). Often exhaustive algorithms may not be available, or when available they may be too slow in practice. Thus, several heuristic algorithms have been developed which are shown experimentally to be able to detect a large fraction of TRs efficiently. Note that the time/precision trade-off is severely influenced by the allowed error thresholds. Performance often degrades quickly with increasing error levels.

1.2.1 Exhaustive algorithms

When we allow no error, it is possible to find all maximal exact TRs in a string of length n in time $O(n^2)$ (Gusfield and Stoye, 2004; Kolpakov and Kucherov, 1999). When we allow two consecutive repeats to differ by an amount at most $k$ (either in Hamming or in edit distance) Landau et al. (2001) give exhaustive algorithms running in time $O(nk\log(k))$ for Hamming distance, and $O(nk\log(k)\log(n/k))$ for edit distance. A simpler algorithm with the same asymptotic complexity for the edit distance is proposed by Sokol et al. (2007). Kolpakov and Kucherov (2003) improved the bound for the Hamming distance to $O(nk\log(k)+s)$ where s is the number of TRs found. For the Hamming distance, Krishnan and Tang (2004) give an exhaustive method running sequentially in time $O(n^3)$, that can be easily implemented onto a parallel architecture, since every possible pattern length is searched independently.

1.2.2 Heuristic algorithms

The algorithmic techniques in Kolpakov and Kucherov (1999, 2003) have been extended in the tool mpera (Kolpakov et al., 2003) so to be able to handle approximate TRs (ATRs) under edit distance, with some additional heuristic filtering steps.

The tool TRF (Tandem Repeat Finder) developed by Benson (1998, 1999), based on statistical filtering of zones of DNA likely to contain TRs, is currently one of the standard heuristic methods. ATRHunter by Wexler et al. (2004) is also based on a statistical filtering approach, placing greater emphasis in techniques for designing thresholds for the quantities of interest. Other proposed heuristics for finding TRs are REPuter (Kurtz and Schleiermacher, 1999; Kurtz et al., 2001), STRING (Paris et al., 2003), TEIRESIAS (Stolovitzky et al., 1999) and TandemSWAN (Boeva et al., 2006). A class of papers (see e.g. Brodzik, 2007; Buchner and Janjarasjitt, 2003; Gupta et al., 2007; Sharma et al., 2004) tackle the problem of finding TRs as a problem in signal processing theory and usually map the input string into a time-signal in a suitable numerical domain for which several spectral techniques can be used, such as the Periodicity Transform or the Fourier Transform. Other methods use data compression techniques to detect repetitive elements (Rivals et al., 1997).

The methods cited above are rather general since they aim at treating efficiently TRs in a wide range of length values. There is also a large class of methods that are aimed at handling particular or special classes of TRs such as: microsatellites [e.g. IMEx (Mudunuri and Nagarajaram, 2007)]; palindromic repeats [e.g. CRISPFinder (Grissa et al., 2007)]; Variable Length TRs (VLTR) and Multi-period TRs (MPTR) (Hauh and Joseph, 2002) and Variable Number TRs (VNTR) (Sammeth and Stoye, 2006). Since the focus of our research on TRs at present is on the more classical forms of TRs, we do not dwell longer on them. However, we just note that often methods for MPTR, VNTR, VLTR use standard TR finding as a subroutine, thus our proposed algorithm can increase also the ability to detect such higher order structures.

Systematic comparison among TR finding tools and algorithms operating ab initio, that is without support of specific biological data bases has been tackled in recent years (Leclercq et al., 2007; Saha et al., 2008). A survey of problems on TRs in the context of evolutionary mechanisms, such as the construction of TR Evolutionary Trees, is proposed in Rivals (2004); see also Elemento and Gascuel (2002).

1.3 Organization of the article

The article is organized as follows: in Section 2 we describe at a high level the principles guiding the different phases of the TRStalker algorithm. Section 3 gives a more technical description of key ingredients of TRStalker and discusses the formal definition of fuzzy TR employed. Section 4 describes the experiments devised to demonstrate the capacity of TRStalker in detecting fuzzy TRs, and a few interesting fuzzy TRs found in sequences of biological significance.

2 APPROACH

An example: To focus on the main ideas, let us consider the very simple case of Exact TR. Consider an alphabet $\Sigma = \{A, C, G, T\}$ of four symbols, and a string $X = x_1x_2…x_q$ formed by the concatenation of $t$ strings $x_i$, embedded in a random string $Y$, where $x_i = x_k$ for all $i$ and $|x_k| = k$, thus all replicas of $x_i$ are of the same length. An ungapped $q$-gram is a string of $q$ symbols from $\Sigma$ that appears as a consecutive sequence of $q$ symbols in $Y$. We aim at discovering $k$ just by looking at the distances between occurrences of homologous (i.e. identical) $q$-grams in $Y$. For $q$-grams in $X$, the period $k$ will appear at least $(k−q+1)(t−1)$ times as the distance between homologous probes. More generally the distance $hk$, an integer multiple of $k$, will appear at least $(k−q+1)(t−h)$ times for each value $h = 1, …, t−1$. A gapped $q$-gram is a sequence of $q$ characters from $\Sigma$ with additional ‘don’t care’ symbols, also called ‘gaps’, that appears as a consecutive sequence in $Y$. For gapped $q$-grams similar formulae hold. For values of $k$ and $t$ large enough, the period $k$ and its integer multiples will occur more frequently than the expected number of occurrences of any distance of homologous $q$-grams in a random string, thus the empirical number of occurrences of the value $k$ and its multiples will tend to be in the higher part of a ranking by frequency. This observation holds true as long as the length
of the super-string $Y$ is sufficiently limited so that the frequencies generated by the random portion of $Y$ do not overrun the frequencies generated by $X$. An exact characterization of such a distribution in terms of the parameters $k$, $l$, $q$ and $|Y|$ is complex since it can be characterized as the sum of non-independent random variables each with a negative binomial distribution. However we avoid the issue of characterizing exactly such a distribution by: (i) splitting the input string into blocks of predefined length and limiting the analysis to each block separately, providing mechanisms to deal with TRs stranded across the block boundaries; (ii) ranking the periods by weighted frequency and exploring only the top $L$ positions (for $L=50$ in our experiments). Note that in most cases the top ranking periods not corresponding to TRs will be discarded quickly when the positional density is considered, thus we can be very slack in choosing the positional density.

The final weight of the same blocks of length within a factor of up to 40 of the length of the TR will be affected by error and a match will be missed, thus reducing the frequency counts for the period $k$. The second filter computes a graphical compact output of the TR.

Post-processing: As a post-processing, we check for inclusion of the TRs found and we filter out those TRs completely enclosed in another one. For TRs in the same position and length but different period we report the TR with shorter period. Finally we align the approximate generalized median string with the TR units so to give a graphical compact output of the TR.

3 METHODS

3.1 Basic definitions

A TR in a DNA sequence is the repetition of two or more contiguous exact or approximate copies of a substring (called the motif) of the TR.

3.1.1 Exact TR

Formally, given an alphabet $\Sigma$, and a set of strings $x_i \in \Sigma^*$, consider the concatenation $X = x_1x_2 \ldots x_n$. The string $X$ is an exact TR (ETR) of period $k$ and repeat number $t$, when $|x_i| = k$ and $x_i = x_1$ for each $i \in \{1 \ldots t\}$. In general we may suppose there is a longer string $Y$ of which $X$ is a substring. The string $x_1$ that is repeated exactly is called the motif of the ETR. A TR $X$ is called maximal if it cannot be extended in $Y$ while still being a TR.

3.1.2 ATR

ETRs are sometimes found in biological sequences, but they tell us only part of the story, thus several notions of an ATR have been developed. Denote with $D_g(a,b)$ the hamming distance of two strings with equal length. If the length of $a$ and $b$ is different, we consider the smallest possible mismatch in an alignment of the two strings without gaps. Denote with $D_g(a,b)$ the edit distance of the two strings $a$ and $b$.

3.2 Our definitions of TR

We used two different definitions of TRs:

- **Neighboring TR (NTR)**: a string $X$, so that for each $i \in \{1 \ldots t-1\}$, $D_g(x_i, x_{i+1}) \leq \mu |x_i|$, for a user defined parameter $0 \leq \mu \leq 1$.

- **Steiner-STR with sum**: a string $X = x_1x_2 \ldots x_n$ for which two conditions hold for a user defined error parameter $0 \leq \mu \leq 1$, and constant $\epsilon$ with $1 \leq \epsilon \leq 2$:
  - (a) for each $i \in \{1 \ldots t-1\}$, $D_g(x_i, x_{i+1}) \leq \epsilon |x_i|$.
  - (b) there exists a Steiner string $k \in \Sigma^*$ so that $\sum_{i=1}^{t-1} D_g(x_i, x_{i+1}) \leq \mu |X|$.
Intuitively, in a Steiner-STR the TR consists of $t$ duplications of a single Steiner consensus string $x$ with $\mu$ mutations on average in each copy, so that consecutive copies do not diverge too much w.r.t. the average. Note that condition (a) is vacuous for $\mu \leq 1/\epsilon$. The choice for the constant $\epsilon$ depends also on the level of divergence. For low divergence $\epsilon = 2$ is a sensible choice since two copies at distance $\mu(x)$ from $x$ are also at distance at most $2\mu(x)$ from each other by the triangular inequality. Thus (a) is a necessary condition for (b). For higher level of divergence above 30%, the value $\epsilon = 2$ is too loose and we use a lower value $\epsilon = 1.5$, so as to maintain a good filtering ability of condition (a) and to avoid having as a possible solution a TR where the consecutive pairs may have a very irregular divergence.

### 3.3 Output of TRStalker

The aim of TRStalker is to produce a ranked list of ATR $\alpha_p$ with respect to one of the copies (called the string for (b)). For higher level of divergence above 30%, the value $\epsilon = 2$ is too loose and we use a lower value $\epsilon = 1.5$, so as to maintain a good filtering ability of condition (a) and to avoid having as a possible solution a TR where the consecutive pairs may have a very irregular divergence.

### 3.4 Other definitions

In Sokol et al. (2007) it is used the following definition: $X$ is called a $k$-edit ATR when $\sum x_k(x_i, x_{i+1}) \leq k$, where the last repeat $x_k$ might be incomplete so $D_k(x_i, x_{i+1})$ is computed as the minimum edit distance of $x_i$ and the prefixes of $x_{i+1}$. This definition is inspired by the evolutionary model of TRs in which it is assumed that TRs are generated by duplicating the last copy of a previous TR, possibly with duplication errors that truncate it. A k-ATR repeat is maximal if it cannot be extended either to the left or to the right without violating its definition.

In Wexler et al. (2004), for a similarity function $\phi$ that measures the alignment score of two sequences, it is defined a $\eta$-Simple ATR ($\eta$-SATR) a string $X = x_1...x_n$ such that: there exists a motif $\eta \in \Sigma^*$ so that for every $i \in [1,...,n]$, $\phi(x_i, \eta) \geq \eta$. In other words, the TR consists of $t$ duplications of a single consensus string $\eta$ with mutations. Such string $\eta$ is also called a Steiner motif if $\eta$ is not constrained to be equal to some repeat $x_i$. Often in practice $\eta$ is chosen as the repeat $x_i$ that minimizes the error function, and is called a pivot motif. The distinction is critical since, as mentioned before, Steiner motifs lead to NP-complete recognition problems, while pivot motifs do not.

The $\eta$-Neighboring ATR ($\eta$-NATR) is a string $X$, so that for each $i \in [1,...,n-1]$, $\forall x_i, x_{i+1}, \geq \eta$ (Wexler et al., 2004). The Pairwise ATR (PATR) is a string $X$, such that for every pair of indices $i, j \in [1,...,n]$ with $i \neq j$ we have $\phi(x_i, x_j) \geq \eta$. In other words, the TR consists of $t$ duplications of a single consensus string $\eta$ with mutations. Such string $\eta$ is also called a Steiner motif if $\eta$ is not constrained to be equal to some repeat $x_i$. Often in practice $\eta$ is chosen as the repeat $x_i$ that minimizes the error function, and is called a pivot motif. The distinction is critical since, as mentioned before, Steiner motifs lead to NP-complete recognition problems, while pivot motifs do not.

### 3.5 Gapped q-grams

Let $I$ be a finite subset of non-negative integers. We call $I$ an index set. The span of $I$ is span($I$) = $\max(|i| − |j|, |j|, i \in I$). The condition $I$ is post($I$) = min$I$ if and only if $I$ is a sequence. For higher level of divergence above 30%, the value $\epsilon = 2$ is too loose and we use a lower value $\epsilon = 1.5$, so as to maintain a good filtering ability of condition (a) and to avoid having as a possible solution a TR where the consecutive pairs may have a very irregular divergence.

Let $I$ be a finite subset of non-negative integers. We call $I$ an index set. The span of $I$ is span($I$) = $\max(|i| − |j|, |j|, i \in I$). The condition $I$ is post($I$) = min$I$ if and only if $I$ is a sequence. For higher level of divergence above 30%, the value $\epsilon = 2$ is too loose and we use a lower value $\epsilon = 1.5$, so as to maintain a good filtering ability of condition (a) and to avoid having as a possible solution a TR where the consecutive pairs may have a very irregular divergence.

Let $I$ be a finite subset of non-negative integers. We call $I$ an index set. The span of $I$ is span($I$) = $\max(|i| − |j|, |j|, i \in I$). The condition $I$ is post($I$) = min$I$ if and only if $I$ is a sequence. For higher level of divergence above 30%, the value $\epsilon = 2$ is too loose and we use a lower value $\epsilon = 1.5$, so as to maintain a good filtering ability of condition (a) and to avoid having as a possible solution a TR where the consecutive pairs may have a very irregular divergence.
with other constants fixed in TRSStalker, we have chosen \( b = 5 \) and \( H = 20 \) since they do well in our synthetic experiments for a large range of TR error and length values. A fine tuning of these parameters as a function of the characteristics of the TR sought is possible, but beyond the focus of this article.

### 3.7 Positional density

Let \( b \) be the period under investigation. Consider the set \( K \) of the positions of those \( q \)-grams (i.e., substrings of \( Y \)) that contribute to the weighting through the multiplicity weighting. In order to avoid double counting, we always take the position of the first of the two matching probes. Note that, if a position is shared by several pairs of probes it will be counted only once. Let \( f( |Y|) = \sum_{j=0}^{\text{DE} \cdot |Y|} f(j) \) that computes the \( k \)-smoothed density of the function \( f \), for \( k \in [1, \ldots, |Y| - k] \). Finally, we define a threshold \( r(k) \) proportional to the average \( k \)-density by a user-defined constant, and we consider as a candidate position set \( CPT(k) = \{(i_1, \ldots, i_k) | f(j) \geq r(k) \} \). The output of this positional density computation is a sequence of pairs \( (k, i) \) where \( k \) is a candidate period and \( i \) a candidate position.

### 3.8 Validation

The definition of Steinerr-STR is composed of two conditions that will be tested in cascade starting from the one less computationally demanding.

3.8.1 Testing condition (a) The WDP technique in Fischetti et al. (1993) solves the following problem. Given a string \( P \) of length \( m \) and a text \( T \) of length \( n \), find the best alignment of \( P \) (concatenation of \( n \) copies of \( P \)) in \( T \), in time and storage \( O(mn) \). Note that a naive application of the standard dynamic programming based optimal alignment of two strings would require \( O(mn^2) \) time/storage. We modify the WDP approach in order to (i) work with edit distance instead of similarity matrices, (ii) take as pattern an adjacent portion of the input string of size \( \alpha \cdot n \) (\( \alpha \) is obtained by applying those operations to the text length till the termination condition is met and (iii) we stop the matching as soon as the next adjacent iteration of the TR differ from the previous one by more than \( \epsilon \) in edit distance.

3.8.2 Testing condition (b) Let \( i_1, \ldots, i_b \) be the candidate TR to test for property (b) that passed the test for property (a). We incrementally compute the approximate generalized median string \( \bar{x}_i \). Initially \( \bar{x}_1 = x_1 \). Let \( k \) and \( h \) be two positive integers and \( K = \lceil k/2 \rceil \) be the set formed by \( k \) equally spaced real values between 0 and 1. For each value \( a \in K \), we determine up to \( h \) median strings between \( \bar{x}_i \) and \( \bar{x}_{i+1} \) with weight \( a \). This set of at most \( bh \) candidates is then searched for the string \( a \) that minimizes the function \( \sum_{j=0}^{\text{DE} \cdot |Y|} D_2(a, x_j) \). So we set \( i_0 = a \) and start the next iteration.

The median string of weight \( a \in [0, \ldots, 1] \) of two strings \( a \) and \( b \) is obtained as follows. Compute the edit distance of \( D_2(a, b) \) and record the set \( \text{As}(a, b) \) of edit transformations that transform \( a \) into \( b \). Pick any subset of size \( \lfloor |a|/2 \rfloor \) in \( \text{As}(a, b) \). The median weighted string \( c \) is obtained by applying those operations to the string \( a \). It is not difficult to show that it holds that \( D_2(c, a) = aD_2(a, b) \) and \( D_2(c, b) = (1-a)D_2(a, b) \). Note that depending on the value of \( a \) we have \( \lceil \frac{1}{2} \rceil \) different subsets of \( \text{As}(a, b) \) we can choose. In our algorithm we randomly select \( \min(k, \lceil \frac{1}{2} \rceil) \) of them.

### 3.9 Evaluation of recall in synthetic sequences

In order to measure the quality of the TRs reported by TRSStalker and by other benchmark algorithms in our synthetic experiments, we need to give a score to a pair of TRs. The higher the similarity of the two TRs, the higher should be the score. Since perfect equality is rare we need a more flexible score function. A TR can be characterized by the triple \((b, p, r)\), where \( b \) is the initial position, \( p \) the period, \( r \) the repetition number. Also, the same TR covers the positions in \( Y \) from index \( b \) to \( b + rp - 1 \). We identify the TR with the set of positions \( \text{Seg}(TR) = \{b, b + rp, \ldots\} \). Given two TRs \( TR_1 \) and \( TR_2 \) represented as sets of positions, the classical Jaccard coefficient measure of set similarity \( JC \) is:

\[
JC(\text{Seg}(TR_1),\text{Seg}(TR_2)) = \frac{|\text{Seg}(TR_1) \cap \text{Seg}(TR_2)|}{|\text{Seg}(TR_1) \cup \text{Seg}(TR_2)|}
\]

3.9.1 Modified Jaccard coefficient Let \( C \) be a TR embedded in \( Y \). Even if \( C \) is a TR according to the definition, when we embed \( C \) in a string \( Y \), it is well possible that \( C \) is not maximal in \( Y \), thus if an algorithm reports correctly \( C \) there will be a slight penalization in the IC measure. This phenomenon arose a number of times, thus we decided to use a modified version of the Jaccard coefficient, called IC, where the denominator is changed. The resulting measure is thus more robust w.r.t. this penalization:

\[
IC(\text{Seg}(TR_1),\text{Seg}(TR_2)) = \frac{|\text{Seg}(TR_1) \cap \text{Seg}(TR_2)|}{\max(|\text{Seg}(TR_1)|, |\text{Seg}(TR_2)|)}
\]

Given a TR \( T \) and a set of TRs \( T' = \{t_1, \ldots, t_n\} \) we define the best-match BM(T, T') as:

\[
BM(T, T') = \max_{T' \in T'} IC(T, T')
\]

and the best-match-score (BMS):

\[
BMS(T, T') = \max_{T' \in T'} IC(T, T')
\]

In our controlled experiments, the evaluation module knows the embedded TR \( T \) and receives the output of an algorithm \( T' \), giving back the BMS. For a series of experiments, we will report the average of the BMS. Note that BMS has values in the range \([0, \ldots, 1]\), and higher values correspond to better quality. At first sight one might consider this metric as overly generous. However, since we cannot rule out the existence of other TRs in \( Y \) besides the embedded ones, we do not want to penalize the presence in \( Y \) of valid TRs different from \( T \). Also, the set \( T \) will not contain nested TRs.

### 3.10 Evaluation of recall on biological sequences

The evaluation has been carried out according to the following procedure. Let \( T_{MAX}, T_{FREQ}, T_{TOP2} \) be the set of TRs found by TRSStalker, TRF and ATRHunter respectively. First, we remove from every set all the TRs that have a Jaccard coefficient greater than a threshold \( J \) when compared with another TR in the same set. In other words, we remove TR duplicates from every set of results, where two TRs are considered as duplicates when they cover the same region with an approximation \( J \). Since TRF and ATRHunter have been executed with options that discard all TRs having a score lower than a given threshold, we filtered TRF by removing all the TRs with a score under such value (this has been done to not penalize TRF and ATRHunter with respect to TRSStalker). More in detail, TRF has been executed with match, mismatch and indel score equal to 2, 3 and 3, respectively, maximum motif length equal to 200bp, and threshold equal to 50. ATRHunter has been executed with match, mismatch, gap and terminal gap score equal to 1 0 1 0, maximum motif length equal to 500bp and threshold equal to 30. For the TRs found by TRSStalker, the score is computed by using the same weights used by TRF and ATRHunter then we filtered the results using the same threshold. After the filtering phase, we computed the union of the TRs found by all algorithms, \( U = \bigcup \{T_{MAX}, T_{FREQ}, T_{TOP2}\} \). The removal of duplicates with threshold \( J \) is also applied to \( U \). Naturally the higher the value of \( J \) less filtering will be performed.

### 4 DISCUSSION

We have performed comparative experiments both with synthetic and with biological sequences. Here we describe the experimental...
set up, how the synthetic sequences are generated and the outcome of the comparison. For biological data, we briefly indicate the reason why that sequence has been selected, and the new TRs found by the application of TRStalker.

4.1 Synthetic data

4.1.1 Generation of synthetic data We carried out a first set of experiments by using synthetic data. This allows a fine grained control on the amount of mutations introduced within the regions covered by the TRs. The sequences we gave as input to the programs have been built according to the following steps:

1. The background sequence is generated by selecting the four bases A,C,G and T with equal probability;
2. a perfect TR is embedded within the previous sequence, the TR is generated as r repetitions of a motif with length l;
3. the region covered by the TR is mutated according to substitution, insertion and deletion probabilities ($p_s$, $p_i$ and $p_d$); the number of substitutions, insertions and deletion for every repetition of the motif is exactly equal to $|p_s|$, $|p_i|$ and $|p_d|$; and
4. if the TR is a Steiner-STR, mutations are introduced in every repeat with respect to the consensus motif; if the TR is a NTR, mutations are introduced with respect to the previous repeat.

The experiments have been carried out running ATRHunter with the parameters: match, mismatch, gap and terminal gap score equal to 1 0 1 0 (the most permissive setting on the website); maximum motif length equal to 500 bp (the maximum allowed by the tool). In order to select the definition of TRs among those allowed by ATRHunter, we performed a preliminary set of experiments: the definition that gave the best results was the third one (minimum alignment score). In this case, ATRHunter reports only the TRs that have a score higher than a given threshold. The value of the threshold has been set to 30.

For the web-based version of TRF, all the experiments have been carried out with these parameters: match, mismatch and indel score equal to 2, 3 and 5, respectively; maximum period equal to 500; minimum score equal to 30. For the binary version, we used the following ones: match, mismatch and indel score equal to 2, 3 and 5, respectively; match and indel probability equal to 0.75 and 0.20; and maximum period equal to 500; minimum score equal to 30. For the binary version, we used the following ones: match, mismatch and indel score equal to 2, 3 and 5, respectively; maximum period equal to 500; minimum score equal to 30. The parameters of the experiments have been set so as to make sure that the minimum allowed score for all the tools tested is attained on the input data. TRStalker runs with the error parameter $c = 1.5$.

Fig. 1. BMS as a function of copy number for NTR. Motif lengths 60 (a), 100 (b), 100 (c) and 300 (d). The total length of the input sequence is 10000 bp; the amount of substitutions, insertions and deletions are equal to 10% of the motif length each (thus with total error allowed of 30%). Every point is the average of 30 measurements and the 95% confidence intervals are shown.

4.1.2 Discussion of the comparative experiments For the experiments on NTR (Fig. 1), we tested TRs with motifs of length from 60 to 300, and a number of repeats from 2 to 8. TRStalker has recall always above 95%. TRF (binary) has always a recall above 80% except for TR with repeat number 2 for which the recall drops to 60%. ATRHunter has recall of a about 60%. These experiments confirm the effectiveness of the new techniques for the initial filtering steps.

Results on Steiner-STR with motifs of length from 60 to 300, and a number of repeats from 2 to 8 are shown in Figure 2. Here we notice that all methods have degraded performance for longer motifs (>200 bases) while TRStalker still manages to have recall above 60%. For shorter motifs (of <100 bases) TRF (binary) is able to match TRStalker only when the repeat number is above 6. Thus for a large range of values, TRStalker attains the best performance in recall, or a matching one, always above 80%.

The time performance of TRStalker has not been yet optimized. At the moment it is within an order of magnitude of TRF and ATRHunter. More details on the running time are in the Supplementary Materials.

4.2 Biological sequences

Testing of TRStalker on biological sequences has confirmed the potential of our method for finding very fuzzy TRs not detected by TRF and ATRHunter, and, to the best of our knowledge, not reported in literature. We tested the following sequences:

1. U43748 Homo sapiens frataxin gene, promoter region and exon—2465 bp long (FRDA).
2. L3609 Homo sapiens germ line T-cell receptor /chain, complete gene—684 973 bp long (HSBT).
3. NC_001133.8 Saccharomyces cerevisiae Chromosome I—230 208 bp long (YCh1).

4.2.1 Experimental settings The three algorithms have been run with the setting used in the synthetic experiments (thus with a very permissive acceptance policy). In general, none of the three algorithms generates all TRs found by the two others, and in Table 1 we show the percentage of the TRs found by each algorithm with respect to the union of the TRs found. In Table 2, we report some very long TRs that were detected by TRStalker but missed by the other two methods. We check the motif/repeat alignments using the tool jaligner [http://jaligner.sourceforge.net/] using the BLOSUM62 matrix.

4For TRF the maximum motif length has been raised to 2000 bp.
Table 1. Evaluation of recall for the three methods under evaluation

| Algorithm                  | Filter 90% | Filter 70% |
|----------------------------|------------|------------|
| Frataxin                   |            |            |
| TRStalker (TRF filter)     | 59 (56.2)  | 43 (56.5)  |
| TRStalker (ATR filter)     | 43 (41.0)  | 30 (39.4)  |
| TRF                        | 24 (22.9)  | 18 (23.6)  |
| ATRHunter                  | 24 (22.9)  | 23 (30.2)  |
| Union                      | 105 (100.0)| 76 (100.0) |
| Homo sapiens T-cell receptor β chain |          |            |
| TRStalker (TRF Filter)     | 22 557 (59.1)| 14 137 (60.2)|
| TRStalker (ATR Filter)     | 18 124 (47.5)| 11 427 (48.7)|
| TRF                        | 99 777 (26.1)| 85 521 (36.0)|
| ATRHunter                  | 7392 (19.3) | 7034 (29.6) |
| Union                      | 38 218 (100.0)| 23 743 (100.0)|
| Saccharomyces cerevisiae chromosome I |        |            |
| TRStalker (TRF Filter)     | 7168 (61.8) | 4656 (63.5)|
| TRStalker (ATR Filter)     | 5621 (48.4) | 3655 (49.9)|
| TRF                        | 2892 (24.9) | 2518 (34.1)|
| ATRHunter                  | 2037 (17.6) | 1958 (26.4)|
| Union                      | 11 616 (100.0) | 7407 (100.0)|

Each entry in the table gives the absolute number of unique TR found, and in the percentage of unique TR w.r.t the union of the three methods. For TRStalker, we used both a TRF-like and an ATRHunter-like filtering (more restrictive) on the TRs found.

4.2.3 Human beta T-cell receptor locus

The cellular immune system detects the presence of pathogens largely through the activation of T-cell receptor proteins (TCR) (Glusman et al., 2001), which come in four different families α, β, γ and δ. The complete DNA sequence of the human β T-cell receptor locus has been determined (Rowen et al., 1996) and it has been found that a large fraction of the locus sequence (about 47%) is formed by locus-specific repeats (Rowen et al., 1996). This sequence was selected as a test case for TRStalker because of its richness in repeating elements with the aim of highlighting the ability of TRStalker in finding repeats with high divergence among adjacent copies. Here, (Table 2) we could find a few such repeats apparently not recorded in the GenBank: L36902.2 record, nor found by TRF and ATRHunter (still set with very loose parameters).

4.2.4 Yeast chromosome 1

Saccharomyces cerevisiae (baker’s yeast) has been the focus of intensive study as the first eukaryotic organism whose genome was completely sequenced (Dujon, 1996), and serves as a model organism in basic genomic investigations. Chromosome 1 (Bussey et al., 1995) is the smallest of the 16 chromosomes present in yeast. It has been noticed that the yeast genome is remarkably poor in repeated elements (Dujon, 1996), thus finding new TRs in such organism is a challenging task for any algorithm. In Table 2 we report a TR in position [186168, 188347] of copy number 2 and motif length 1089. This TR is not reported in the TRDB database, while ATRHunter in the same region finds 15 shorter TR of length ranging from 50 to 180. This region, according to the NCBI record, is rich in genes of the DUP240 gene family (encoding membrane proteins). The presence of a fuzzy repeat in this region thus suggests a possible remote gene duplication event.

4.2.5 Performance on biological sequences

Reporting interesting single new TRs, as in Table 2, is useful to demonstrate that biological relevant TRs are still unknown. We give also an evaluation of the overall behavior of the three different methods on biological sequences. Thus, we compared TRStalker, TRF and ATRHunter by estimating their recall on the three biological sequences with the methodology described in Section 3.10. Table 1 reports (i) the number of unique TRs found by the different algorithms and (ii) the percentage of the union reported by a given algorithm, with two filtering thresholds at \( J = 90\% \) and \( J = 70\% \). For all the three sequences, TRStalker is able to find a large number of TRs that are not discovered by using the other methods. In practice, a better overall coverage can be attained by using all three methods and merging their results. Although lower \( J \) values imply...
Table 2. Examples of TRs found by TRStalker and missed by TRF and ATRHunter

| No. | Sequence | Seq. length | TR start | TR end | TR length | Consensus | Repetitions | Score | Norm. score |
|-----|----------|-------------|----------|--------|-----------|-----------|-------------|-------|-------------|
| 1   | HSBT     | 684,973     | 411,000  | 413,127| 2127      | 1061      | 2.00        | 2868  | 1.384       |
| 2   | HSBT     | 684,973     | 448,001  | 449,687| 1686      | 842       | 2.00        | 2310  | 1.370       |
| 3   | HSBT     | 684,973     | 636,116  | 638,622| 2506      | 1253      | 2.00        | 3323  | 1.326       |
| 4   | YCh1     | 230,208     | 186,168  | 188,347| 2179      | 1089      | 2.00        | 3055  | 1.401       |
| 5   | FRDA     | 2465        | 2029     | 2407   | 378       | 188       | 2.01       | 501   | 1.328       |

We report the original sequence name and length, the TR starting and ending positions, the TR length and the TR repeating unit length and copy number. The score is computed by assigning +2 to matches and −1 to mismatches and gaps w.r.t the consensus string. The normalized score is the score divided the TR length.

Table 3. Motif/repeats alignment scores computed by jaligner using the BLOSUM62 score matrix with gap open penalty set to 10.0 and gap extend penalty set to 0.5 for the TRs reported in Table 2

| Seq. | No. | Repeat | Length, N | Identity, n(%) | Gaps, n(%) | Score |
|------|-----|--------|-----------|----------------|------------|-------|
| HSBT | 1   | 1      | 1107      | 805/7(22)      | 91 (8.22)  | 3657.00|
|      | 1   | 2      | 1093      | 895/88.88      | 70 (6.60)  | 4291.00|
| HSBT | 2   | 1      | 878       | 638/72.67      | 85 (9.68)  | 3045.50|
|      | 2   | 2      | 866       | 716/82.68      | 52 (6.00)  | 3568.00|
| HSBT | 3   | 1      | 1300      | 1000/76.92     | 94 (7.23)  | 5206.00|
|      | 3   | 2      | 1313      | 1004/76.47     | 120 (9.14) | 5176.50|
| YCh1 | 4   | 1      | 1130      | 895/79.20      | 83 (7.35)  | 4280.50|
|      | 4   | 2      | 1123      | 901/80.23      | 77 (6.86)  | 4345.50|
| FRDA | 5   | 1      | 193       | 149/77.20      | 10 (5.18)  | 723.50 |
|      | 5   | 2      | 191       | 146/76.44      | 5 (2.62)   | 765.00 |

a more aggressive filtering, the percentage of the union attained by TRStalker is almost constant.

5 CONCLUSION

TRStalker is a novel efficient heuristic algorithm for finding Fuzzy TRs in biological sequences. TRStalker aims at improving the capability of TR detection for a class of fuzzy TRs for which existing methods do not perform well. Initial testing on biological data show that fuzzy TRs not previously reported are present in biologically relevant sequences. In the case of the Frataxin sequence, the fuzzy TR reported is associated with the known variable copy number breakpoint of Frederich’s ataxia. Future work will involve testing TRStalker on relevant families of repetitive elements such as centromeric α-satellites. An extension of TRStalker to handle amino acid sequences is under development.

Funding: European Community’s Seventh Framework Programme FP7/2007-2013 under grant agreement N 239392 (Virtual Physiological Human Network of Excellence) partially. The fuzzy TR reported is associated with the known variable copy number breakpoint of Frederich’s ataxia. Future work will involve testing TRStalker on relevant families of repetitive elements such as centromeric α-satellites. An extension of TRStalker to handle amino acid sequences is under development.

Conflict of Interest: none declared.

REFERENCES

Ames,D. et al. (2008) Comparative analyses of human single- and multisite tandem repeats. Genetics, 179, 1693–1704.

Benson,G. (1998) An algorithm for finding tandem repeats of unspecified pattern size. In Istrail,S., Pevzner,P. and Waterman,M. (eds), Proc. 6th Annual International Conference on Computational Molecular Biology (New York, New York, United States, March 22–25, 1998), RECOMB ’98 ACM, New York, NY, pp. 20–29.

Benson,G. (1999) Tandem repeats finder: a program to analyze DNA sequences. Nucleic Acids Res., 27, 573–580. Available at http://tandem.bu.edu/trf/trf.html (last accessed date April 27, 2010).

Bussey,H. et al. (2001) Comparative genomics of the human and mouse T cell receptor loci. Tandem and non-tandem repeat sequence. Trends Genet., 17, 169–174.

Bussey,H. et al. (2007) The CRISPRdb database and tools to display CRISPRs and to generate dictionaries of spacers and repeats. BMC Bioinformatics, 8, 172.
et al

Jurka, J., Kurtz, S. and Schleiermacher, C. (1999) Reputer: fast computation of maximal repeats. In *Proceedings of the Ninth International Conference on Intelligent Systems for Molecular Biology*, August 1-7, 2002, Edmonton, Alberta, Canada, pp. 31–37.

Jiang, X. et al. (2003) Dynamic computation of generalized median strings. *Pattern Anal. Appl.*, 6, 185–193.

Jurka, J. et al. (2005) Repbase update, a database of eukaryotic repetitive elements. *Cytogenet. Genome Res.*, 110, 462–467.

Kelk, Y.D.D. et al. (2008) The genome-wide determinants of human and chimpanzee microsatellite evolution. *Genome Res.*, 18, 30–38.

Kolpakov, R. and Kucherov, G. (2003) Finding approximate repetitions under Hamming distance. *Theor. Comput. Sci.*, 303, 135–156.

Kolpakov, R. et al. (2003) mreps: efficient and flexible detection of tandem repeats in DNA sequences. *Nucleic Acids Res.*, 31, 3672–3678. http://www.isds.hri.fmrps/ (last accessed date April 27, 2010).

Kolpakov, R.M. and Kucherov, G. (1999) Finding maximal repetitions in a word in linear time. In *Proceedings of the 40th Annual Symposium on Foundations of Computer Science* (October 17–18, 1999), FOCS. IEEE Computer Society, Washington, DC, pp. 596, 596.

Krishnan, A. and Tang, F. (2004) Exhaustive whole-genome tandem repeats search. *Bioinformatics*, 20, 2702–2710.

Kurz, S. and Schliep, A. (1999) Reputer: fast computation of maximal repeats in complete genomes. *Bioinformatics*, 15, 426–427.

Kurtz, S. et al. (2001) Reputer: the manifold applications of repeat analysis on a genomic scale. *Nucleic Acids Res.*, 29, 4653–4662.

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

Leclercq, R. et al. (2007) Detecting microsatellites within genomes: significant variation among algorithms. *Bioinformatics*, 23, 1–125.

Legendre, S. et al. (2007) Sequence-based estimation of minisatellite and microsatellite repeat variability. *Genome Res.*, 17, 1787–1796.

Morton, R. and Raglavan, P. (1995) Randomized Algorithm. *Cambridge University Press*, Cambridge, UK.

Musumeci, B. and Negrin, J.R. (2007) Imex: Imperfect microsatellite extractor. *Bioinformatics*, 23, 1181–1187.

Mudunuri, S.B. (1993) *Computational Geometry, an Introduction through Randomized Algorithms*. Prentice Hall, Englewood Cliffs, New Jersey.

O’Doherty, C. et al. (2005) Tandem repeat copy-number variation in protein-coding regions of human genes. *Genome Biology*, 6, R89.

Patnaik, V. et al. (2003) String: finding tandem repeats in DNA sequences. *Bioinformatics*, 19, 1733–1738.

Peterlongo, P. et al. (2009) Lossless filter for multiple repeats with bounded edit distance. *Algorithms Mol. Biol.*, 4, 1–3.

Rivals, E. et al. (2004) A survey on algorithmic aspects of tandem repeats evolution. *Int. J. Found. Comput. Sci.*, 15, 225–257.

Rivas, E. et al. (1997) Detection of significant patterns by compression algorithms: the case of approximate tandem repeats in DNA sequences. *Comput. Appl. Biosci.*., 13, 131–136.

Rosen, L. et al. (1996) The complete 685-kilobase DNA sequence of the human beta T Cell Receptor Locus. *Science*, 272, 1755–1762.

Sahu, S. et al. (2008) Empirical comparison of ab initio repeat finding programs. *Nucleic Acids Res.*, 36, 2234–2239.

Sammeth, M. and Stoye, J. (2006) Comparing tandem repeats with duplications and excisions of variable degree. *IEEE/ACM Trans. Comput. Biology Bioinform.*, 3, 395–407.

Sharma, D. et al. (2004) Spectral repeat finder (SRF): identification of repetitive sequences using Fourier transformation. *Bioinformatics*, 20, 1405–1412.

Sim, J.-S. and Park, K. (2003) The consensus string problem for a metric is NP-complete. *J. Discrete Algorithms*, 1, 111–117.

Smit, A.F.A. et al. (1996–2004). Repeatmasker open-3.0. http://www.repeatmasker.org/ (last access date April 27, 2010).

Sokol, D. et al. (2007) Tandem repeats over the edit distance. *Bioinformatics*, 23, 30–35.

Stodolitsky, G. et al. (1999) Tandem repeat detection using pattern discovery with applications to the identification of yeast satellites. *Theoretical Computer Science* 222,508.

IBM TJ Watson Research Labs.

Vissers, L.E. et al. (2009) Rare pathogenic microdeletions and tandem duplications are microhomology-mediated and stimulated by local genomic architecture. *Hum. Mol. Genet.*, 18, 3579–3593.

Wexler, Y. et al. (2009) Effect of repeat copy number on variable-number tandem repeat mutations in Escherichia coli. *J. Bacteriol.*, 188, 4253–4263.

Wuraba, P. et al. (2008) Analysis of the largest tandemly repeated DNA families in the human genome. *BMC Genomics*, 9, 533.

Wells, P.D. et al. (2008) DNA triplexes and Friedreich ataxia. *FASEB J.*, 22, 1625–1634.

Wexler, Y. et al. (2004) Finding approximate tandem repeats in genomic sequences. In *Proceedings of the Eighth Annual International Conference on Research in Computational Molecular Biology (RECOMB 2004)*, New York, NY, USA. ACM.

Wexler, Y. et al. (2005) Finding approximate tandem repeats in genomic sequences. *J. Comput. Biol.*, 12, 928–942.

Wooten, K. et al. (1994) Instability of short tandem repeats (microsatellites) in human cancers. *Nat. Genet.*, 6, 152–156.