BIOINFORMATICS

Vol. 19 no. 14 2003, pages 1733–1738
DOI: 10.1093/bioinformatics/btg268

STRING: finding tandem repeats in DNA sequences

Valerio Parisi[1,], Valeria De Fonzo[2] and Filippo Aluffi-Pentini[3]*

[1]Sez. INFM, [2]EuroBioPark, Univ. Roma 'Tor Vergata' Via della Ricerca Scientifica 1, 00133 Roma, Italy and [3]Dip. Metodi e Modelli Matematici, Univ. Roma 'La Sapienza' Via A. Scarpa 16, 00161 Roma, Italy

Received on November 4, 2002; revised on March 13, 2003; accepted on April 15, 2003

ABSTRACT

Motivation and results: The importance of Tandem Repeats in some genomes is now well established. We have reported elsewhere some interesting new results obtained by means of a preliminary program for finding Tandem Repeats in DNA sequences, together with a brief description of the basic ideas of the algorithm. We describe here a completely new program based only in part on those ideas, we briefly discuss the interpretation of the results, and, by way of example, we provide a few novel results relative to the parasites responsible of two re-emerging diseases, *Plasmodium falciparum* and *Mycobacterium tuberculosis*. Our program is portable, effective, powerful and fast: it can run on current desktop computers, and it finds all significant Tandem Repeats also in the longest segments of sequences in databases (up to millions of bases), in short times (minutes).

Availability: An academic version of the algorithm (full source listing in standard C language) can be freely downloaded (http://www.caspur.it/∼castri/STRING/).

Contact: valerio.parisi@roma2.infn.it

Supplementary information: Some illustrative figures and some sample results are provided as supplementary material at: http://www.caspur.it/∼castri/STRING/.

1 INTRODUCTION

The examination of DNA sequences, trying to make assumptions about how a given sequence might have been generated by successive mutations acting on a given or hypothetical starting sequence, is a scientifically and practically interesting problem. An important aspect of genomic sequences is the presence of repetitions, that can be scattered (as the Alu sequences) or consecutive, i.e. tandemly repeated. A Tandem Repeat (TR) is a tract of DNA where a unit is tandemly repeated exactly or nearly exactly a number of times. In this paper we shall deal only with TRs.

The importance of TRs in some genomes is well-established. It is moreover well known that in some places of the human genome one finds a Variable Number of Tandem Repeats (VNTRs), i.e. TRs where the number of repetitions is variable: variable within a population (polymorphism), variable from parent to offspring (genetic instability), variable within the same organism (mosaicism); the search for VNTRs is of growing importance: for example it is now commonly acknowledged that in many cases their excessive expansions cause several diseases (Richards and Sutherland, 1996). The best known cases are some nervous system diseases, where the repeated unit is often a triplet (Reddy and Housman, 1997).

We therefore consider of paramount interest to have powerful and efficient modern algorithms to effectively find all TRs that can be considered significant (for example according to a suitable figure of merit), especially in the cases of very long sequences (up to millions of bases), also in view of further (bioinformatic or experimental) studies about important TR features such as their instability (Jakupciak and Well, 2000; De Fonzo *et al*., 2000) or their propensity to build anomalous three-dimensional structures (Keniry, 2000; Shafer and Smirnov, 2000; De Fonzo *et al*., 2001).

We reported elsewhere (De Fonzo *et al*., 1998; Bersani *et al*., 2001) some interesting new results obtained by means of a preliminary program for finding tandem repeats in DNA sequences, together with a brief description of the basic ideas of the algorithm.

We describe here a completely new program (STRING: Search for Tandem Repeats IN Genomes) based only in part on those ideas, we briefly discuss the interpretation of the results, and, by way of example, we provide a few novel results relative to the genomes of *Plasmodium falciparum* and *Mycobacterium tuberculosis*.

We note that there exist a number of algorithms for finding tandem repeats. A detailed list is provided by the dedicated Web site by Professor Ramaswamy's Group (http://bic.jnu.ac.in/anju); a more restricted but critical examination of some important algorithms can be found in the paper of Benson (1999). It is clear from the above resources that most algorithms cannot be considered as competitive candidates for general purpose use, due to various kinds of design limitations. For example, some of them look only for TRs having

*To whom correspondence should be addressed.

a repeated unit of limited length (Sagot and Myers, 1998), or separated by error blocks of fixed length (Karlin *et al*., 1988). Other algorithms (Rivals *et al*., 1997, and the references contained therein) exploit the unconventional idea that sequences that contain more repeats are obviously more compressible by many compression algorithms, but limit themselves to only signal a suspected presence of TRs whenever a high compressibility is encountered. Of course the suspect must be then verified by other means.

Also from a more extensive survey of the references quoted in the above sources, it appears to us that the recent algorithm by Benson (1999) is the only good candidate for a general purpose use (in fact his program is used by several groups, and can be easily used online) and the scarcity of different tools is by itself, in our opinion, a good reason to justify the proposal of new algorithms, especially if based on completely different ideas.

2 METHODS

It is well known that genome sequences are subject to many kinds of modifications, such as point mutations, i.e. substitutions, insertions and deletions (*indels* for short), and expansions, i.e. exact tandem replications of some tracts. Situations that are more complicated may arise, such as an expansion followed by point mutations and further expansions, which may be called *nested* expansions.

In the present work we do not pursue the most general case, but we restrict ourselves to a simpler but practically more important aim: to assess if some tracts of a given sequence can be interpreted as follows. An *exact TR*, i.e. a single exact tandem repetition of a suitable word called *consensus*, underwent a (small) number of point mutations thus producing an *inexact TR*.

One must therefore compare a suitable prospective exact TR (*model TR*) with a suitable tract of the given sequence: and while if only point substitutions are present a simple one-to-one comparison is sufficient, indels produce misalignments. The standard tool in bioinformatics to compare misaligned sequences is an alignment procedure; in this case we use a local alignment procedure (Section 3.1), which, for a given proposed consensus, provides the best one-to-one correspondences between the model TR and some tracts of the sequence. The interpretation of a tract of a sequence as a TR is therefore fully described by the pair consisting of the hypothesized consensus and the associated alignment, and this pair will be called *Single-Expansion Interpretative Pattern*, or more briefly *SIP*.

We could think to perform all possible local alignment procedures between all possible model TRs and the given sequence, and to select only the best SIPs by means of a suitable optimization procedure. Unfortunately the number of possible local alignments for a given consensus is an exponential quantity (in the length of the tract to be examined), and the number of possible consensus words is also an exponential quantity (in the length of the consensus).

The exact algorithmic search of the best local alignments for a given model TR can be performed by means of a dynamic programming procedure as indicated in Section 3.2. Dynamic programming allows to lower the search time, for the best local alignments for a given consensus, from an exponential time to a polynomial time, but unfortunately as far as we know there exist no similar exact methods to reduce the exponential complexity of the number of possible consensus words. It is therefore necessary to resort to approximate methods.

Our approach has been to exploit some heuristically plausible criteria to reduce the size of the search space by considering, by means of a suitable pre-screening, only a reasonably restricted number of 'promising' sequence tracts and of 'promising' possible consensus words.

In the past (De Fonzo *et al*., 1998) we adopted the idea of searching tandem (and not scattered) repeats by means of a local autoalignment procedure (i.e. a search of local alignments of a sequence with itself, see Section 3.1) satisfying suitable conditions and of considering as TRs the autoaligned tracts. Of course it is not strictly correct to consider such autoaligned tracts as TRs: by their very nature such autoalignments compare every repeated unit only with the adjacent ones, possibly of different length (see phase 1 of http://www.caspur.it/∼castri/STRING/figure1.pdf), instead of correctly comparing every repeated unit with a same consensus word. Therefore, using such procedure, one may in some cases fail to detect the fact that many small changes in adjacent unit may produce a sort of 'drift' effect, so that far-away entirely different repeated units could be indicated as belonging to the same TR (for details see De Fonzo *et al*., 1998). We therefore consider the autoaligned tracts only as provisional TRs: they only suggest promising sequence tracts and possible consensus words, which will be more thoroughly examined later by the algorithm.

In more detail, our two basic heuristic criteria are:

(a) Instead of studying the whole sequence we examine only the tracts (*interesting zones*) that we consider to be the more promising ones, i.e. those including autoaligned bases, according to autoalignments that we consider significant since they have a score greater than a given significance threshold.

(b) Instead of studying all possible consensus words we examine only the words that we consider to be the more promising ones, suitably selecting them (in a way inspired by the autoalignments) only among those already present in the considered tract.

3 ALGORITHM

We give here a brief overview of the algorithm, while a more detailed description of the algorithm and of its software implementation will be given elsewhere.

The algorithm has two phases. The first phase obtains prospective TRs using the search for autoalignments used in a previous work (De Fonzo *et al*., 1998), and is described in Section 3.2; the second phase performs the final TR search, based on the above heuristic criteria, and is described in Section 3.3.

From a purely logical standpoint, the second phase can be thought as occurring after the end of the first one, although practical reasons (especially for very long sequences) suggested us a more intermingled use.

We first introduce some notation.

3.1 Alignments

Given two sequences $\underline{a}$ and $\underline{b}$ of respectively $L(\underline{a})$ and $L(\underline{b})$ bases

$$\underline{a} = (a_i)_{i\in[1,L(\underline{a})]} \quad \text{and} \quad \underline{b} = (b_i)_{i\in[1,L(\underline{b})]}$$

an *alignment* of the two sequences $\underline{a}$ and $\underline{b}$ is defined by two strictly increasing sequences $\underline{\alpha}$ and $\underline{\beta}$ of the same number $L(\underline{\alpha}) = L(\underline{\beta})$ of indices chosen respectively in $[1, L(\underline{a})]$ and in $[1, L(\underline{b})]$ that produces the pairs

$$(\alpha(k), \beta(k)), \quad \text{for } k = 1, \ldots, L(\underline{\alpha})$$

of corresponding (*aligned*) indices, and the pointed pairs

$$(a_{\alpha(k)}, b_{\beta(k)}), \quad \text{for } k = 1, \ldots, L(\underline{\alpha})$$

of aligned bases. If $a_{\alpha(k)} > b_{\beta(k)}$ we say that a point substitution occurred at the positions $\alpha(k)$ and $\beta(k)$.

For the sequence $\underline{\alpha}$ and for $k = 1, \ldots, L(\underline{\alpha}) - 1$, let

$$\Delta\alpha(k) = \alpha(k+1) - \alpha(k) - 1$$

If $\Delta\alpha(k) > 0$ we say that there is a gap in the sequence $\underline{\alpha}$ at location k, of length $\Delta\alpha(k)$, indicating a deletion of $\Delta\alpha(k)$ bases going from $\underline{a}$ to $\underline{b}$. Obviously there is no gap if $\Delta\alpha(k) = 0$. The same applies for the sequence $\underline{\beta}$. A usually implicitly assumed condition is that for each k at least one of the two sequences $\underline{\alpha}$ and $\underline{\beta}$ does not have a gap.

An *autoalignment* of a sequence $\underline{b}$ is simply an alignment of the sequence $\underline{b}$ with itself, and its use will be considered in Section 3.2.

For purposes of evaluation and comparison it is necessary to give a measure of the quality of an alignment. The measure of the quality is sometimes based on probabilistic criteria, but more often simple additive scores are used. The scores we shall use are as follows. We evaluate the pairs of aligned bases by giving a positive score p (say $p = 10$) if the bases are equal, and a negative score q (say $q = -30$) if the bases are different. The use of the above numerical values for the additive score stems from probabilistic considerations (De Fonzo *et al*., 1998). For a base pair (x, y) the score will therefore be

$$w(x, y) = \begin{cases} p & \text{if } x = y \\ q & \text{if } x \neq y \end{cases}$$

We shall evaluate the gaps in the sequences $\underline{\alpha}$ and $\underline{\beta}$ by giving a negative score W_0 (say $W_0 = -100$) for the very existence of a gap, and a negative score W_1 (say $W_1 = -10$) for any non-aligned base of the gap. For a gap of length g the score will therefore be

$$W(g) = \begin{cases} 0 & \text{if } g = 0 \\ W_0 + W_1 \cdot g & \text{if } g > 0 \end{cases}$$

and is usually called *affine gap penalty*: such a choice allows a great speed, as first pointed out by Gotoh (1982), in dynamic programming optimization.

The overall score given to an alignment will therefore be

$$S(\underline{a}, \underline{b}, \underline{\alpha}, \underline{\beta}) = \sum_{i=1}^{L(\underline{\alpha})} w(a_{\alpha(i)}, b_{\beta(i)}) + \sum_{i=1}^{L(\underline{\alpha})-1} W(\Delta\alpha(i)) + \sum_{i=1}^{L(\underline{\alpha})-1} W(\Delta\beta(i))$$

and higher scores will denote 'better' alignments.

We note that it may well be that there exist non-aligned bases that do not belong to the above gaps. Such bases are sometimes said to belong to the so-called *external* gaps. In such a parlance our formula is said to refer to a *local* alignment, while if a score is given also to external gaps one speaks of a *global* alignment. In this paper we shall not consider global alignments.

3.2 Phase 1

We briefly recall here the use of autoalignment of a sequence $\underline{b}$ to search for TRs (De Fonzo *et al*., 1998).

We define the offset between two aligned indices

$$\gamma(k) = \beta(k) - \alpha(k), \quad \text{for } k = 1, \ldots, L(\underline{\alpha})$$

We first consider a few ideal oversimplified cases that can be easily detected but have of course a limited practical interest, and are considered here only for clarity's sake.

If, for a given k, the aligned bases are equal, i.e. $b[\alpha(k)] = b[\beta(k)]$, there is a base exactly repeated with offset $\gamma(k)$. If for n consecutive values of k, $b[\alpha(k)] = b[\beta(k)]$ and $\gamma(k)$ is a constant, say γ_C:

- if $\gamma_C = n$ there is a unit of γ_C bases tandemly repeated exactly twice;
- if $\gamma_C < n$ there is a unit of γ_C bases tandemly repeated $(n/\gamma_C + 1)$ times. There is a partial superposition between the two aligned tracts and γ_C is the repetition period along the TR;
- if $\gamma_C > n$ we must distinguish: if γ_C is much greater than n there is a scattered repeat of a unit of n bases, with offset γ_C (a case not examined here), while if γ_C is only slightly greater than n, it may be better to speak of an exact TR that underwent some mutation, for example an insertion of length $(\gamma_C - n)$.

A more interesting and frequent case in practice is the case of an inexact TR where the aligned bases are equal not for all pairs but for almost all pairs, and $\gamma(k)$ is not constant but nearly constant and less than n (see phase 1 of http://www.caspur.it/∼castri/STRING/figure1.pdf); in this case $\gamma(k)$ can be called a *pseudoperiod*.

Our algorithm for searching autoalignments uses the dynamic programming of Bellman (1957), along the lines first suggested by Needleman and Wunsch (1970) for the global alignment between two sequences, where the optimization is performed with respect to the score defined in Section 2. In more detail our algorithm is analogous to the algorithm of Waterman and Eggert (1987), which was devised, like the algorithm of Smith and Waterman (1981), for searching the best local alignments of two different sequences. The algorithm of Waterman and Eggert performs a sequence of dynamic programming optimizations, with the progressive condition of non-intersection between pairs of alignments: each pair of aligned indices, involved in any of the higher scoring alignments, must not be considered in the current optimization.

The basic differences of our algorithm with respect to the algorithm of Waterman and Eggert are those needed to make it apt to detect TRs, i.e. to look for auto-alignments (instead of alignments), and to add two important constraints aimed at detecting only TRs. In more detail our constraints can be written:

$$1 \leq \gamma(k) \leq \gamma_{\max}, \quad k = 1, \ldots, L(\underline{\alpha})$$

where $\gamma_{\max}$ is a fixed given maximum value (say $\gamma_{\max} = 100$). The left constraint is needed to avoid the trivial identical autoalignment, while the right constraint is needed to restrict the search to Tandem Repeats (of course TRs with a pseudoperiod greater than $\gamma_{\max}$ are not detected). Moreover the last constraint is extremely advantageous from the standpoint of both memory requirement and computing time: both quantities, as functions of the length $L(\underline{b})$ of the considered sequence, grow linearly, $O[\gamma_{\max} \cdot L(\underline{b})]$, instead of quadratically, $O[L^2(\underline{b})]$. When considering sequence lengths of the order of 10^6 bases (as in Section 5), such gains are absolutely necessary.

In phase 1 our algorithm provides local auto-alignments, and therefore tracts consisting of words consecutively repeated more or less identically (inexact TRs): we shall use such words as preliminary candidates to be considered consensus words, and such tracts as 'interesting' tracts to be further explored.

For future reference we define the *extension* of the considered autoalignment (or of the tract of prospective TR) the interval starting from the first index of $\underline{\alpha}$ and ending with the last index of $\underline{\beta}$, i.e.

$$e(\underline{\alpha}, \underline{\beta}) \equiv [\alpha(1), \beta(L(\underline{\alpha}))]$$

and its *augmented extension* the interval

$$\tilde{e}(\underline{\alpha}, \underline{\beta}) \equiv \left[\alpha(1) - \tfrac{1}{2}D_0, \beta(L(\underline{\alpha})) + \tfrac{1}{2}D_0\right] \cap [1, L(\underline{b})]$$

i.e. the interval obtained by symmetrically increasing the length of the extension by a given quantity D_0 (say $D_0 = 100$), but without overflowing outside the sequence.

3.3 Phase 2

The second phase can be schematized as follows.

We group in suitable clusters all the autoalignments obtained in phase 1, by putting in the same cluster all the autoalignments whose augmented extensions are not disjoint. For each cluster we perform the following actions.

We define the extension of the cluster as the union of all the augmented extensions of the autoalignments of the cluster. This extension will be used as the *interesting zone* within which we shall restrict any further search for TRs, according to the heuristic criterion (a) in Section 2.

Instead of studying all possible consensus words we suitably extract some segments ('words') of the interesting zone that we consider as the more promising ones and that we shall examine as prospective consensus words, i.e. as *candidates* to be later declared consensus words. To this end we could consider, for each autoalignment of the cluster, all the $L(\underline{\alpha})$ words defined by the autoalignment, i.e. the tracts of the interesting zone containing all the bases pointed by the indices from $\alpha(k)$ to $\beta(k) - 1$, and having therefore length $\gamma(k)$, for $k = 1, \ldots, L(\underline{\alpha})$.

The words we shall use as candidates are not all the above words, for example since the same word may appear many times, or since a word may be a TR of a shorter word. For each candidate we perform a local alignment procedure between the corresponding model TR and the interesting zone (with the same score function defined in phase 1, possibly with different values for p, q, W_0, W_1). Our algorithm, as in the first phase, is similar to the Waterman and Eggert algorithm, with an important difference: the model TR sequence is treated as a single consensus word cyclically addressed. This is not a simple coding detail, but is essential since it reduces the required memory and the computing time from quadratic to linear, as in phase 1. Each local alignment procedure produces, for each candidate, a number of local alignments having a score greater than a given threshold, and therefore a number of SIPs: we remind here that a SIP is the couple of a consensus and of an alignment, for a graphic example see the phase 2 of http://www.caspur.it/∼castri/STRING/figure1.pdf. For each cluster we compare pairwise each SIP with each one of all other SIPs, in order to assess if one of them can be considered a trivial and less significant version of the other, and therefore must be discarded: an example is provided in http://www.caspur.it/∼castri/STRING/figure2.pdf

We perform the comparison by first taking into account the amount of overlapping of the two tracts of the original sequences involved in the two alignments. If there is no overlapping, neither SIP is discarded by this comparison. If there is overlapping, the comparison takes suitably into account the amount of overlapping, the two scores, and the length of

the two consensus words. If there is complete overlapping, the SIP with lower score is discarded, unless it has a sufficiently shorter consensus, thus suggesting a likely nested expansion that is beyond the scope of our search, but is nevertheless indicated in the results. A partial overlapping is treated as complete or absent, according to the amount of overlapping.

We finally consider as a *good consensus* the candidate word associated with each one of the remaining SIPs and we consider as a *good TR* the tract of the original sequence involved in the alignment of the SIP. We note that with different values for the score parameters, the optimal alignments may differ: for example it may well happen that a single TR may split into two TRs if one increases the weight of the gap penalty.

Such *good SIPs* will be the core of the output of our program.

4 IMPLEMENTATION

The algorithm has been implemented in standard C language and has undergone extensive preliminary testing, with satisfactory results, on many available sequences containing VNTRs. The program name, STRING, stands for 'Search for Tandem Repeats IN Genomes'.

For each accepted SIP, the output of the program consists of

- the length of the consensus word;
- the first and the last position of the TR;
- the number of repeated units, defined as the (possibly non-integer) ratio between the length of the model TR and the length of the consensus;
- the score;
- the consensus word;
- the flanking sequences;
- the alignment between the model TR and the given sequence;
- the number of insertions and deletions;
- the number of matches and mismatches;
- the TR base composition percentages;
- a flag indicating a likely nested expansion.

5 RESULTS

While a systematic exploitation of our program to produce novel results is deferred to a future occasion, we report here, by way of example, a few novel results obtained by running STRING on the genomes of two interesting unicellular parasites, a prokaryote and a eukaryote, namely *Mycobacterium tuberculosis* and *Plasmodium falciparum*.

The study of these parasites is important since they are responsible for two re-emerging diseases. It is important to note that in their genomes many VNTRs are present (Frothingham and Meeker-O'Connell, 1998; Frontali, 1994) and that these VNTRs appear to be useful for the parasite in escaping the host immune system. Since future endeavours of contrasting such diseases will most likely require a detailed understanding of the dynamics of the involved VNTRs, it is most likely of strategic importance to detect all the TRs in order to perform deeper analyses.

We have examined the complete genome of the strain H37Rv of *M.tuberculosis* and the complete sequence of chromosomes 2 and 3 (the only ones already completely sequenced) of *P.falciparum*.

The runs have been performed setting a threshold score equal to 150 for the first phase and equal to 200 for the second phase (below these values, too many non-significant statistical fluctuations occur). We have restricted our search to consensus length between 3 and 100, discarding TRs having less than 2.5 repeated units.

For the other parameters we have used the following values:

$$\gamma_{\max} = 100 \quad \text{(phase 1)}$$
$$D_0 = 100 \quad \text{(phase 2)}$$
$$p = 10, \quad q = -30, \quad W_0 = -100, \quad W_1 = -10 \quad \text{(both phases)}$$

We have obtained the following results:

- 226 TRs for the complete genome of *M.tuberculosis* H37Rv (GenBank accession: NC_000962, 4411529 bp).
- 555 TRs for the chromosome 2 of *P.falciparum* (GenBank accession: NC_000910, 947103 bp).
- 708 TRs for the chromosome 3 of *P.falciparum* (GenBank accession: NC_000521, 1060106 bp).

The detailed results, for the best 50 TRs of each run, are reported at http://www.caspur.it/∼castri/STRING/tables.pdf

The meaning of the column headings is as follows:

- start, end: the start and the end of the TR;
- length: the length of the consensus word;
- repeat: the number of repetitions;
- score: the total score of the alignment between the TR and the model TR;
- ins, del, match, mism: the number of insertions, deletions, matches and mismatches (of the alignment between the TR and the model TR);
- A%, C%, G%, T%: the percentage of bases in the TR;
- consensus: the consensus word (only the first 10 bases);
- nesting: a flag that indicates a probably nested expansion.

A comparison of the results of the runs for the two organisms indicate that *M.tuberculosis* has a lower ratio TRs/bp, longer TRs with more uniform lengths, and lower number of cases of likely nested expansions.

6 CONCLUSION

STRING has been designed to be portable since it has been written in a portable subset of standard C, by deliberately avoiding risky constructs; it is therefore usable without problems on any standard UNIX system, and its portability has also been verified on some desktop operating systems (Windows, MacOS). Moreover, possible incompatibilities with peculiar compilers should be easily fixed by the user with minimal adjustments in the source code, which is available, structured, readable and with many comments.

STRING is remarkably effective since it apparently finds all significant Tandem Repeats, and it is powerful, since it has successfully coped also with the longest segments of sequences in databases (up to millions of bases).

As for the required computing time we have observed that it slowly increases as a function of the sequence length, while it increases more quickly as a function of the number of TRs. To give a rough order of magnitude we can say that the program goes through the longest sequences in times of the order of minutes, and can be therefore considered satisfactorily fast.

We now consider briefly the problem of the comparison of algorithms.

As for a theoretical comparison, we note that till now, as far as we know, no accepted objective figure of merit for the biological importance of a TR is available, and therefore there is no easy way, from an objective standpoint, to perform a satisfactory evaluation (e.g. to declare false positives or negatives and to assess the significance of the results).

As for an experimental comparison, we think that, following a common fairness practice, a thorough comparative benchmark should be best systematically performed by a third party, on a hopefully accepted set of test problems (obviously taking in due account the possible presence of different tunable input parameters, that can drastically affect the performance).

We feel that the non-existence, as far as we know, of such set of test problems and the scarcity of available good algorithms, indicates that the situation is still far from a mature state, and therefore the proposal of new algorithms (especially if based on quite different ideas) may be useful.

As a purely preliminary comparison, we ran online (simply with default parameters) version 2.02 of Benson's algorithm (http://c3.biomath.mssm.edu/trf.html), on some test sequences, including the above sequence of *M.tuberculosis*; the results were practically the same (except for a few borderline cases).

REFERENCES

Bellman,R.E. (1957) *Dynamic Programming*. Princeton University Press, Princeton, NJ.

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

Bersani,E., De Fonzo,V., Aluffi-Pentini,F. and Parisi,V. (2001) On new hypotheses about Autosomal Dominant Polycystic Kidney Disease type 1. *Med. Hypotheses*, **57**, 754–758.

De Fonzo,V., Bersani,E., Aluffi-Pentini,F. and Parisi,V. (2000) A new look at the challenging world of tandem repeats. *Med. Hypotheses*, **54**, 750–760.

De Fonzo,V., Bersani,E., Aluffi-Pentini,F. and Parisi,V. (2001) DNA quadruplexes and dynamical genetics. *Med. Hypotheses*, **57**, 103–111.

De Fonzo,V., Bersani,E., Aluffi-Pentini,F., Castrignanò,T. and Parisi,V. (1998) Are only repeated triplets guilty? *J. Theor. Biol.*, **194**, 125–142.

Frontali,C. (1994) Genome plasticity in Plasmodium. *Genetica*, **94**, 91–100.

Frothingham,R. and Meeker-O'Connell,W.A. (1998) Genetic diversity in the *Mycobacterium tuberculosis* complex based on variable numbers of tandem DNA repeats. *Microbiology*, **114**, 1189–1196.

Gotoh,O. (1982) An improved algorithm for matching biological sequences. *J. Mol. Biol.*, **162**, 705–708.

Jakupciak,J.P. and Well,R.D. (2000) Genetic instabilities of triplet repeat sequences by recombination. *IUBMB Life*, **50**, 355–359.

Karlin,S., Morris,M., Ghandour,G. and Leung,M.-Y. (1988) Efficient algorithms for molecular sequence analysis. *Proc. Natl Acad. Sci. USA*, **85**, 841–845.

Keniry,M.A. (2000) Quadruplex structures in nucleic acids. *Biopolymers*, **56**, 123–146.

Needleman,S.B. and Wunsch,C.D. (1970) A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.*, **48**, 443–453.

Reddy,P.S. and Housman,D.E. (1997) The complex pathology of trinucleotide repeats. *Curr. Opin. Cell Biol.*, **9**, 364–372.

Richards,R.I. and Sutherland,G.R. (1996) Repeat offenders: simple repeat sequences and complex genetic problems. *Hum. Mutat.*, **8**, 1–7.

Rivals,É., Delgrange,O., Delahaye,J.-P., Dauchet,M., Delorme,M.-O., Hénaut,A. and Ollivier,E. (1997) Detection of significant patterns by compression algorithms: the case of approximate tandem repeats in DNA sequences. *Comput. Appl. Biosci.*, **13**, 131–136.

Sagot,M.-F. and Myers,E.W. (1998) Identifying satellites and periodic repetitions in biological sequences. *J. Comput. Biol.*, **5**, 539–554.

Shafer,R.H. and Smirnov,I. (2000) Biological aspects of DNA/RNA quadruplexes. *Biopolymers*, **56**, 209–227.

Smith,T.F. and Waterman,M.S. (1981) Identification of common molecular subsequences. *J. Mol. Biol.*, **147**, 195–197.

Waterman,M.S. and Eggert,M. (1987) A new algorithm for best subsequence alignments with application to tRNA–rRNA comparisons. *J. Mol. Biol.*, **197**, 723–728.