Detecting Fuzzy Amino Acid Tandem Repeats in Protein Sequences

Marco Pellegrini
Istituto di Informatica e Telematica
CNR – Consiglio Nazionale delle Ricerche
56124 Pisa, Italy
marco.pellegrini@iit.cnr.it

M. Elena Renda
Istituto di Informatica e Telematica
CNR – Consiglio Nazionale delle Ricerche
56124 Pisa, Italy
elena.renda@iit.cnr.it

Alessio Vecchio
Dipartimento di Ingegneria dell'Informazione
Università degli Studi di Pisa
56122, Pisa, Italy
a.vecchio@ing.unipi.it

ABSTRACT

Tandem repetitions within protein amino acid sequences often correspond to regular secondary structures and form multi-repeat 3D assemblies of varied size and function. Developing internal repetitions is one of the evolutionary mechanisms that proteins employ to adapt their structure and function under evolutionary pressure. While there is keen interest in understanding such phenomena, detection of repeating structures based only on sequence analysis is considered an arduous task, since structure and function is often preserved even under considerable sequence divergence. In this paper we present *PTRStalker*, a new algorithm for *ab-initio* detection of *very fuzzy tandem repeats* in protein amino acid sequences. In the reported results we show that by feeding PTRStalker with amino acid sequences from the UniProtKB/Swiss-Prot database we detect novel tandemly repeated structures not captured by other state-of-the-art tools.

Categories and Subject Descriptors

H.3.3 [**INFORMATION STORAGE AND RETRIEVAL**]: Information Search and Retrieval; J.3 [**LIFE AND MEDICAL SCIENCES**]: Biology and genetics

General Terms

Algorithms

Keywords

Tandem repeats, amino acid tandem repeats, protein sequence analysis, repetitive structures

1. INTRODUCTION

In a seminal paper of 2001 M.A. Andrade and co-authors [1] observe that repetitive subsequences that appear tandemly often form integrated assemblies when viewed in their three-dimensional corresponding conformation, as they often confer multiple binding opportunities and play a structural role in the protein. Moreover, tandemly repeated structures constitute a different class from *domains* and *motifs* that appear singly in each protein (while they can be repeated across families of protein). M.A. Andrade and co-authors also remark that repeats in protein sequences are usually hard to detect because of their relatively (on average) short length and because of their considerable -inherent- sequence divergence.

A study of 1999 by Marcotte et al. [17] indicate that internal subsequence repetitions in proteins primary structure are quite widespread, they have been noticed in about 14% of all the known proteins, with eukaryotic proteins being three times more as likely to have internal repeats than prokaryotic ones. The distribution of TRs in protein families and the mechanisms of protein TR formations are discussed in detail in [4].

From a theoretical point of view almost any approach that works for a 4-characters alphabet (DNA) could also be applied to the 20-characters amino acid alphabet and indeed some methods are developed for "generic" biological sequences (e.g. [6]). In practice, there are differences between protein sequences and DNA sequence, and the way they can be analyzed. Protein sequences tend to be shorter (at most of the order of 10^4 amino acids) than DNA sequences (often of the order of 10^6 nucleotides and more). In proteins assigning a score to an amino acid substitution is important, and finally repetitions in amino acid sequences tend to be very divergent, since the same structural role can be maintained even with little sequence conservation. Also, empirically, we observe that the tools for detecting (tandem) repeats for DNA and for proteins constitute two rather distinct families, often employing completely different characterization and algorithmic approaches.

In [24] we presented TRStalker, an algorithm designed, developed and tuned to detect fuzzy tandem repeats in DNA sequences. In view of the biological relevance of tandemly repeated subsequences in proteins, we worked on an evolution of TRStalker, modified and tuned to work with protein sequences in order to detect amino acid (aa) tandem repeats, by incorporating amino acid similarity information deep into the algorithmic framework. Amino acid substitution matrices (such as the PAM and BLOSUM families

ACM-BCB '11, August 1-3, Chicago, IL, USA

of matrices, based on known evolutionary changes of amino acids in proteins) are usually expressed in terms of a *similarity score* between amino acids. In the framework we present here, we use the notion of weighted edit *distance* between sequences, thus converting standard similarity scores into distance weights. Interest in metric-space search and indexing in bioinformatics [19] has been recently rekindled since one can tap on a vast array of efficient metric-based generic methods developed over the years, thus reducing the need for developing ad-hoc searching and indexing algorithms for each new similarity measure. A technique for turning BLOSUM log-likelihood similarity matrices into corresponding metric space was proposed in [9], while a second method to turn PAM log-likelihood similarity matrices into a corresponding metric space was proposed in [31]. It is also possible to define new metrics over amino acids by first mapping each amino acid into a real vector space (see e.g. [2]) and then applying one of the many possible metrics associated to a real vector space. For this reason we generalize the definition of tandem repeat in order to incorporate this larger class of metrics in the proposed algorithm.

PTRStalker, the novel algorithm we developed, has been compared against competing state-of-the-art methods over large collections of proteins (UniprotKB/Swiss-Prot) as well as on membrane channels proteins and on selected families of proteins known to contain long fuzzy TRs. We found out that 19.58% of the protein listed in UniPtotKB/Swiss-Prot have significant fuzzy TRs. We could find fuzzy TRs, not detected by competing methods, in Human Titin (striated muscles). For the Chlorine channel protein ClC-0 (membrane transport) we show that PTRStalker can detect known symmetries while competing methods do not. The output of our analysis of protein sequences has been collected into a publicly available database, that will be continuously updated in order to provide useful insight to researchers interested in protein sequence analysis.

The outline of the paper is the following: after a short introduction on the algorithms developed for detecting tandem repeats in protein sequences (Section 2), in Section 3 we introduce and describe PTRStalker procedural approach in detail. Section 4 describes the experiments we performed to evaluate the metric chosen, and the results our algorithm obtained, while Section 5 concludes.

2. STATE OF THE ART

The first methods developed for finding TRs in proteins are based on detecting sub-optimal alignments in the self-alignment matrix generated by the Smith-Waterman algorithm. Some examples are Internal Repeat Finder [17, 25], RADAR [10], REPRO [11, 26] and TRUST - Tracking Repeats Using Significance and Transitivity [29], even if they often detect both tandem and interspersed repeats. More recently Newman and Cooper proposed XSTREAM [22], which uses a seed expansion approach, Jorda and Kajava proposed T-REKS [13], which uses a clustering approach based on k-means, while Sokol et al. [28] proposed TRED, which is based on an estimation of edit distance between strings. The systems HHrep [27] and HHRepID [3] are instead based on building and matching Hidden Markov Models for the repeating substrings to be sought (not necessarily tandem). Some approaches based on neuronal networks aim at detecting particular repetitive structures. For example, in [23], Palidwor et al. develop a classification techniques for detecting alpha-rods repeats, a specific repetitive structure. A meta searching approach that combines the output of different algorithms has been also proposed, for example in the tool REPPER [8]. With a few notable exceptions (see e.g. [6]) all these methods specialize on protein sequences. Among the -few- databases reporting protein repetitive subsequences we recall ProtRepeatsDB [14], a relational database of perfect and mismatch repeats, which also provides cross species comparisons of different types of amino acid repeats.

3. PTRSTALKER

In this Section we describe PTRStalker, an evolution of the algorithm TRStalker [24] we developed for finding TRs within DNA sequences. Whereas TRStalker and PTRStalker use the same overall procedural approach to detect tandemly repeated subsequences, PTRStalker has been opportunely tuned for detecting TRs in protein sequences. In particular, the main differences are in the formal definition of a tandem repeat, in the metrics used, and in the q-grams matching policy (see below). However, the overall idea still holds: TRs are detected by *(i)* first finding a set of candidate periods, *(ii)* then finding a set of candidate pairs (period, starting position), and *(iii)* finally verifying if in a particular position there exists a TR according to the precise mathematical definition adopted (see below). To make this paper self contained, after introducing a few working definitions we will recall the general structure of the algorithm presented in [24], describing the innovative parts PTRStalker has.

3.1 Preliminaries

In order to extend the methodology used in [24] by incorporating amino acid similarity information encoded in similarity matrices, we need to perform two tasks: define a *weighted edit distance*, and suitably modify the tandem repeat definition so to obtain a *Weighted Steiner Simple Tandem Repeat* (Weighted Steiner STR). We give here details for the BLOSUM metric space as defined in [9]. The extension to other metrics is straightforward.

BLOSUM-weighted edit distance. We incorporate the definition of BLOSUM-derived edit distance between two strings as follows; the cost of indel operations is equal to 1, while the cost of substitutions depends on the involved characters. Since BLOSUM matrix entry M_{ij} defines the level of similarity between characters i and j, and not their distance, the cost C_{ij} of a substitution between characters i and j is computed, according to [9], as the ratio:

$$C_{ij} = \frac{D_{ij}}{D_{max}}$$

where

$$D_{ij} = M_{ii} + M_{jj} - 2M_{ij} \ ,$$

$$D_{max} = \max_{i \in \Sigma, j \in \Sigma} D_{ij}$$

and Σ is the amino acid alphabet. Since D_{ij} represents the non-normalized distance between two characters, we use D_{max}, representing the maximum distance between two characters, as the normalization value so to obtain the normalized weight $C_{ij} \in [0, 1]$. In [9] it is shown that these weights are consistent, since they almost always satisfy the triangular inequality.

BLOSUM-weighted Steiner-STR. Let $D_B(a, b)$ be the BLOSUM-derived edit distance between strings a and b computed by using the BLOSUM-derived weights defined above. In order to give meaningful limits to the divergence of strings under this metric we need to define its behavior for random pair of amino acid strings. For the expected edit distance of two strings no analytic formula is known even in the unweighted case. Even if the theory of Karlin and Altshul [15] could give some insight, the issue of extending this methodology to the case of tandem sequences is not trivial. Thus we decided to estimate $D_B(a, b)$ with an experimental approach, by generating a suitable number of random strings in which the probability of amino acid substitution is consistent with the BLOUSM data, and by measuring the weighted edit distance for a given error level. Alternatively, a rough estimate can be obtained as follows. We approximate the expected BLOSUM-derived edit distance of two random amino acid strings with its correspondent BLOSUM-derived Hamming distance (ie. we disregard insertions and deletions). We define with $E(C)$ the expected substitution cost among two amino acids due to the cost matrix C. Thus two random amino acid stings of length $p = |a| = |b|$ are at expected distance $pE(C)$. If we allow only a percentage μ of substitutions we obtain an estimated weighted distance of $\mu pE(C)$.

A *BLOSUM-weighted Steiner-STR* is defined as a string $X = x_1x_2..x_t$ for which two conditions hold, for a user defined error parameter $0 \leq \mu \leq 1$, and constant c with $1 \leq c \leq 2$:

(a) for each $i \in [1, .., t-1]$, $D_B(x_i, x_{i+1}) \leq c\mu|x_i|E(C)$

(b) there exists a Steiner string $\bar{x} \in \Sigma^*$ so that for each $i \in [1, .., t]$, $D_B(\bar{x}, x_i) \leq \mu|x_i|E(C)$

Intuitively, in a BLOSUM-weighted Steiner-STR the TR consists of t duplications of a single Steiner consensus string $\bar{x}$ with at most μ times the number of mutations one would expect from random strings of the same length (condition (b)). Moreover consecutive copies of the mutated string do not diverge too much w.r.t. each other, at most $c\mu$ times the number of mutations one would expect from random strings of the same length (condition (a)). Note that condition (a) is vacuous for $\mu \geq 1/c$. The choice for the constant c depends also on the level of divergence. For protein repeats with low divergence $c = 2$ is a sensible choice since two copies at distance $\mu|x_i|E(C)$ from $\bar{x}$ are also at distance at most $2\mu|\bar{x}|E(C)$ from each other by the triangular inequality. Thus (a) is a necessary condition for (b). For the higher level of divergence we are interested in ($\mu = 0.3$), the value $c = 2$ is too loose and we use a lower value $c = 1.5$, so 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. Note that the standard TR definition for the unweighted edit distance corresponds to a matrix with cost 1 for each substitution.

Homologous q-grams. Let I be a finite subset of non-negative integers. We call I an *index set*. The *span* of I is $span(I) = max\{i - j | i, j \in I\}$, the position of I is $pos(I) = \min i \in I$, and the *shape* of I is $shape(I) = \{i - pos(I) | i \in I\}$. When $|I| = q$ and $span(I) = s$, the shape of I belongs to the class of (q, s)-shapes. Any set of non-negative integers Q containing 0 is a shape. For an alphabet Σ (the 20 amino acid letters in our case), a string $S \in \Sigma^*$ of length n can be seen as a function defined over $[0, .., n-1]$ with values in Σ, and for any subset $I \subset [0, .., n-1]$ the restriction of S to I, denoted by $S[I]$, is a substring of S. Given any shape Q in the class of (q, s)-shapes, all sets $I \subset [0, ..n-1]$ such that $shape(I) = Q$, form the set of $Indexes(Q, n)$. We can use elements from the $Indexes(Q, n)$ to generate restrictions for the string S. An index set I such that $|I| = q$ and $span(I) = q - 1$ is called an *ungapped q-gram* since its shape is $shape(I) = [0, ..q-1]$. If we have an index set J with $|J| = q$ and $span(J) = s \geq q$ we have a *gapped q-gram* since its shape is formed of non consecutive integers. As noted in [5] gapped q-grams are strictly more powerful than ungapped ones. In order to generate a population of candidate periods we consider now all possible (q, s)-shapes with $q = 3$ and $s = 4, 3, 2$. Denoting with $-$ the gaps and with $\#$ symbols from Σ, (the first and last positions must be always $\#$), we have the $(3, 4)$-shapes $\#\#--\#$, $\#-\#-\#$, and $\#--\#\#$; the $(3, 3)$-shapes $\#-\#\#$, $\#\#-\#$; and the $(3, 2)$-shape $\#\#\#$.

Gapped q-grams (also called *spaced seeds*) have been used in [12, 16, 30] to speed up homology search. In these papers much larger values of q and s are used in order to attain sensitivity. Therefore the key problem for them becomes how to select one (or a few) effective seed out of an exponentially large family (for s and q). Since we use small values of (q, s) we do not have this seed selection problem and we can afford to use a complete family, formed by 6 seeds only.

In defining the notions of *homologous q-grams* for amino acid sequences we take into account the fact that functional properties of proteins are preserved even under considerable sequence substitutions. The BLOSUM similarity matrices give a quantitative definition of the allowed amino acid substitutions. Given two index sets $I_1, I_2 \in Indexes(Q, n)$, we call them *matching* (or *homologous* in S, if $S[I_1]$ and $S[I_2]$ have two identical symbols in corresponding positions while the symbol in the remaining position can be different in the two strings but within ranking z of each other as given by the BLOSUM similarity matrix (that is, the one symbol is among the top z most similar symbols to the other one, and vice versa). The slackness parameter is $z = 1$ when we want exact match, and $z = 3$ and $z = 5$ when we allow more substitutions. The value $|pos(I_1) - pos(I_2)|$ is called the *period* of the match.

Note that, by considering the z amino acids closer to a given amino acid, we introduce a discrete ranking in the metric space. Alternatively one could choose a fixed radius and consider all amino acids within that radius. We performed dedicated experiments in order to choose the right approach and the results (not shown here) indicate that there is almost

no difference in the two approaches. We decided to adopt the discrete ranking approach because choosing suitable radii implies ad-hoc dependance on the specific metric.

Anti-smear weighting. The anti-smear weighting technique allows to cope with the fluctuations in the period of matching q-grams introduced by insertion and deletions of amino acids in a sequence. If q_1 and q_2 are occurrences of homologous q-grams in X at distance k, before the implant of mutations, the effect of insertion and deletions on the positions of the string X between q_1 and q_2 is to alter their distance so that a different period k' is detected. The difference $k - k'$ is equal to the algebraic sum of number of insertions and deletions in the positions between q_1 and q_2. Assuming that any such position can be an insertion or a deletion independently with the same probability, the random variable $k - k'$ is distributed as a sum of independent r.v. with values in $\{+1, -1, 0\}$ with mean value 0, thus, by a Chernoff bound argument, its tail distribution decays exponentially [20, 21]. Also near-by probes in X have small variations in the value of the shift $k - k'$. Inspired by the above observation we devise a weighting scheme that increments the total weight of period k if another period of value $\bar{k}$ is discovered in a near-by position, with weights that decay exponentially with $|k - \bar{k}|$.

Let q be a q-gram in the input string Y at position i. Let $q_1, .., q_h$ be the next h occurrences of q in Y following the occurrence at position i. The h corresponding detected distances are $k_g = j_g - i$, for $g \in [1, ..h]$. For the period k_g, we increment its weight:

$$w_0(k_g) = w_0(k_g) + 1 + \sum_{k \in Q} 2^{-|k_g - k|}, \qquad (1)$$

where Q is a queue holding the last H detected distances in the sequential scan of the input string Y. This way, the final weight $w_0(k)$ for a given period k is the sum of the individual anti-smear weights computed above for probes at distance k. After the weight update, we enqueue all h values k_g in the queue Q, and we dequeue an equal number h of items.

Multiplicity weighting. The goal os this technique is to strengthen the signal when the TR is made by more than two repeating units. Let $w_0(k)$ be the weight of the period k as assigned by the anti-smear weighting procedure. As observed before for a TR with a large number of copies we will find also integer multiples of k with a relatively high frequency. We take advantage of this fact and compute new weights:

$$w_1(k) = \sum_{h \geq 1} w_0(hk). \qquad (2)$$

The candidate periods are then sorted by the weight $w_1(.)$, and processed in decreasing order.

Positional k-density. We further exploit the property of TRs for which the same period is detected by probes in near-by positions. The *positional k-density* is the density of probes that contribute to the counter for the candidate period k. Let k be the period under investigation. Consider the set K_k of the positions of those q-grams (i.e. substrings of Y) that contribute to the weighting of k 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 $f : [1, .., |Y|] \to \{0, 1\}$ the characteristic function that for each position in Y denote the membership of that position to K_k. Consider the k-window smoothing of f: $F(i) = \sum_{j=i}^{i+k} f(j)$ that computes the k-smoothed density of the function f, for $i \in [1, .., |Y| - k]$. Finally we define a threshold $t(k)$ proportional to the average k-density by a user-defined constant, and we consider as a candidate position set $CP(Y, k) = \{i \in [1, .., |Y| - k] | F(i) \geq t(k)\}$. 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.

TR validation. We take each candidate pair (p, i) and explicitly test whether there is a TR of period p starting in position i according to the definition used, by using an alignment technique based on the well-known Smith-Waterman algorithm. In this phase, besides validating the TRs, we discover the (fractional) repetition number of the TRs eventually extracted. Finally, we check for inclusion 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.

Let Y be the input string:
1. $L = 50$, $K, T = \{\}$
2. **for each** Y_j **in** *block*(Y)
3. **for each** P^i **in** *findGappedQGrams*(Y_j)
4. $K' =$ *periods*(P^i)
5. **for each** k **in** K'
6. $w(k) =$ *updateWeight*(k, i)
7. **if** $!((k, i) \in K)$
8. $K = K \cup (k, i)$
9. $\widetilde{K} =$ *posDensity*(K)
10. $\widetilde{K}_L =$ *getTopPeriods*$(\widetilde{K}, L)$
11. **for each** (k, i) **in** $\widetilde{K}_L$
12. $t =$ *getTR*(k, i)
13. **if** *verifyTR*(t)
14. $T = T \cup t$
15. **for each** $t \in T$
16. **if** $(!$*maximal*(t) || $!$*minp*$(t))$
17. $T = T \setminus t$
18. **return** T

Figure 1: PTRStalker algorithm scheme.

3.2 The algorithm

In the following we will go through the high level pseudocode of PTRStalker, reported in Figure 1), explaining each phase and function in detail.

Step 1. The number L of candidate periods to examine is empirically set to 50, while the set of candidate pairs K, and the set of TRs T to be given as output are initialized as empty sets.

Step 2. The function $block(Y)$ splits the input sequence Y into n blocks Y_j, $1 \leq j \leq n$, of predefined length (we used a value of 2000 aa). We limit our computations within a block so to avoid counting q-grams when they are too far to be involved in a TR local to the block. For TRs stranding across blocks we adopt mechanisms to carry over useful information from one block to the next one.

Steps 3–4. The candidate periods are found by detecting the distance (counted in amino acid positions) among *homologous q-grams*. Because of the presence of substitutions, insertions, and/or deletions, many instances of q-grams will be probably affected by error and a match could be missed, thus reducing the frequency counts for the candidate period k. In order to cope with this effect, for each block Y_j, function $findGappedQGrams(Y_j)$ records for each occurrence of a *gapped q-gram* P^i in Y its distances K' to the next 5 occurrences (candidate periods) and its starting position i in Y. Note that the candidate periods will be processed later in order of cumulative weight.

Steps 5–8. For each period k detected by a q-gram at position i, the function $updateWeight(k, i)$ increments the weight $w(k)$ of the period $k \in K'$ by applying the two weighting techniques presented before: the anti-smear weighting technique to cope with the fluctuations in the period of matching q-grams introduced by insertion and deletions of amino acids in a sequence (see Equation 1), and the multiplicity weighting (see Equation 2) to strengthen the signal when the TR is made by more than two repeating units. For computing the anti-smear weight as shown in Equation 1 we empirically set $h = 5$ and $H = 20$. The candidate pair (k, i) is then added to the set K, if it is not already present.

Step 9. PTRStalker further exploits the property of TRs that the same period is detected by probes in near-by positions through the positional k-density. Thus, the function $posDensity(K)$ computes the density of probes that contribute to the counter for the candidate period K, and then cuts off for position with low density, returning those candidates with higher positional density.

Step 10. Function $getTopPeriods(\widetilde{K}, L)$ ranks the periods by *weighted frequency* and returns only the top L positions in the set $\widetilde{K}_L$.

Steps 11–14. For each candidate pair (k, i), functions $getTR()$ computes a candidate TR of period k starting at position i and $verifyTR()$ verifies whether there is a tandem repeat t of period k starting at position i according to the definition of TR given, and if so t is added to the set T. Recall that for a BLOSUM-weighted Steiner-STR we set $\mu = 0.3$ and $c = 1.5$. In this validation phase, the expected BLOSUM-derived edit distance of two random amino acid strings has been roughly approximated by its correspondent BLOSUM-derived Hamming distance.

Steps 16–18. Finally, for each candidate tandem repeat $t \in T$ the function $maximal()$ verifies whether t is included in a longer TR, and possibly removes t from T, while $minp()$ for TRs in the same position and length but different period maintains only the TR with shorter period. The procedure returns the set T as result.

Metric	$z = 1$	$z = 3$	$z = 5$
BLOSUM 90	26.5	30.8	34.3
BLOSUM 70	31.8	38.7	40.7
BLOSUM 50	34.5	41.7	43.8
Edit	18.2	-	-

Table 1: UniProtKB/Swiss-Prot database: percentage of protein sequences that contain at least a tandem repeat with length ≥ 14.

Metric	$z = 1$	$z = 3$	$z = 5$
BLOSUM 90	17.7	20.5	23.2
BLOSUM 70	21.6	25.6	26.1
BLOSUM 50	22.5	27.4	29.1
Edit	4.3	-	-

Table 2: UniProtKB/Swiss-Prot database: percentage of protein sequences that contain at least a tandem repeat with length ≥ 20.

The elements of T can be visualized and listed according to different properties of the TR found: initial position, final position, repeating unit size, number of repetitions, total length, absolute divergence, mean divergence, etc.

4. EXPERIMENTS

In this section we report PTRStalker performance with the use of a BLOSUM-based metric and the unweighted edit distance. Furthermore we report a detailed comparison of PTRStalker versus some state of the art algorithms. For some of the experiments reported below, PTRStalker has been run to analyze the UniprotKB/Swiss-Prot database of protein sequences. More in detail, the version we used in our experiments was UniProtKB/Swiss-Prot Release 57.15 of 02 March 2010 -henceforth called Swiss-Prot database, that contains 515, 203 sequence entries.

4.1 Metric Evaluation

We performed a set of experiments to evaluate the effects of BLOSUM-based metrics and q-gram similarity. For this reason PTRStalker has been run on the first one thousand sequences of the database, using the definition of BLOSUM-weighted Steiner-STR. We also evaluated the influence of the q-gram similarity matching parameter z. Table 1 reports the percentage of sequences that contain at least a TR with length ≥ 14 when using edit distance and BLOSUM-based distance (three different matrices have been used: BLOSUM 90, BLOSUM 70, and BLOSUM 50). The table also show the results obtained for different levels of q-gram similarity. When $z = 1$ similarity is not considered: a q-gram is only similar to itself. When $z = 3$, for every q-gram also the two other more similar q-grams have been considered (the total number is three), and so on. Tables 2, 3, 4 show the results obtained when considering TRs with length ≥ 20, ≥ 30, and ≥ 40. The maximum sensitivity is attained for almost all length classes with $z = 5$ and the BLOSUM70 matrix. The use of BLOSUM metrics more than doubles the sensitivity for TR of length above 20 aa w.r.t the unweighted edit distance.

Metric	*z = 1*	*z = 3*	*z = 5*
BLOSUM 90	5.7	6.4	7.8
BLOSUM 70	7.0	7.0	8.7
BLOSUM 50	7.1	8.4	8.8
Edit	2.2	-	-

Table 3: UniProtKB/Swiss-Prot database: percentage of protein sequences that contain at least a tandem repeat with length ≥ 30.

Metric	*z = 1*	*z = 3*	*z = 5*
BLOSUM 90	4.3	4.9	4.9
BLOSUM 70	4.9	5.5	5.4
BLOSUM 50	4.9	5.7	5.7
Edit	1.8	-	-

Table 4: UniProtKB/Swiss-Prot database: percentage of protein sequences that contain at least a tandem repeat with length ≥ 40.

These experiments show that the use of BLOSUM based metrics is effective in detecting longer fuzzy TRs.

4.2 Proteins With Very Long Tandem Repeats

In this experiments we compared PTRStalker with two state of the art algorithms: XSTREAM [22] and T-REKS [13], for the task of detecting very long tandem repeats (spanning more than 4000 aa). HHRep and HHRepID are less suitable for this task since they report pairs of homologous substrings, thus failing to report multi-repeating units. We have selected 12 proteins from Swiss-Prot database for which a tandem repeat of length $\geq$ 4000 aa has been detected by PTRStalker. The data in Table 5 show that T-REKS with the standard parameter setting for these long proteins does not detect fuzzy TRs longer than 100 aa in 6/12 cases (marked *), and, even when longer TRs are found, these are often sub-TR of those found by the other to methods. PTRStalker and XSTREAM have a remarkable consistence in detecting the location of the longest fuzzy TR in 11/12 cases. Sometimes they differ in the periodicity, since XSTREAM gives priority to higher repeat number (and consequently shorter period), while PTRStalker prefers TR with longer span (often attained with a lower copy number and longer period). One notable difference in the output of PTRStalker and XSTREAM is for the Human Titin sequence (involved in the functioning of vertebrate striated muscles). This protein is one of the longest and most complex human proteins. Here PTRStalker is able to detect a much longer TR (4-repeat, 1082-period) completely missed by the other two methods.

4.3 Membrane Proteins

Membrane proteins perform a wide range of biological functions including signal transduction and molecular transport. Here we report the behavior of PTRStalker in detecting fuzzy tandem repeats in two families of well known membrane proteins: the *Urea Transport Channels* and the *Chloride Channels*.

4.3.1 Urea Transport Channels

Transmembrane transport protein biology is key to the understanding of many critical biological process such as absorption and distribution of drugs within the human body. Transport proteins are basic component of transmembrane channels and often exhibit interesting symmetries at the structural level. Since experimental structural resolution of membrane proteins is difficult one would like to extract as much information as possible from the analysis of sequence data. In a set of experiments we tested the ability of PTRStalker in detecting blindly known repetitive structures in proteins of the Urea Transporter (UT) family [18]. Many members of the UT family are known to have a dimeric structure but the level of amino acid identity between homologous subsequences is rather poor. Here we employ the metric based on the BLOSUM30 substitution matrices. For the purpose of detecting long fuzzy dimeric structures in protein the tools XSTREAM and T-REKS proved unsuitable, therefore we tested PTRStalker against the tools HHRep and HHRepID. Results in Table 6 show that all three methods could detect the some dimeric structure of the four UT proteins under examination. In the case of UT-A1 [Mus musculus] PTRStalker attains a notably longer alignment w.r.t those reported by the two alternative methods, therefore giving a result more consistent with the known structure of that protein.

4.3.2 Chloride Channel ClC-0

CLC Channels are a family of membrane proteins whose major action is to translocate chloride ions across cell membranes. They have been subject to intense study since the cloning and identification of this protein in the species Torpedo marmorata (marbled electric ray) in 1990 [7]. This first identified protein, denoted CLC-0 (UniProtKB locus CICH_TORMA, accession P21564), is 805 aa long. It is divided into a pore domain in the region [49 – 507] and a cytoplasmatic domain in the region [508 - 805]. The pore domain has a diadic structure made of 18 alpha-helices organized in two symmetric groups of 9 alpha-helices each. The cytopalsmatic domain contains two CSB subdomains in positions [543 – 601] and [719 – 776]. Our hypothesis is that by analyzing just the amino acid sequence we can gain insight into the symmetries present in the structure of the protein (at least at a high level). We submitted the ClC-0 sequence to PTRStalker, XSTREAM, T-REKS, HHrep and HHrepID. As shown in Table 7, HHrep identifies two homologous regions in positions [542 – 597] and [718 – 770] which coincide almost exactly with the CSB subdomains and two (short) partially overlapping domains that do not correspond to the global symmetry of the pore domain. Instead, PTRStalker discovers a tandem structure in positions [8 – 289] [291 – 575] that covers most of the pore domain respecting its symmetry, and a tandem structure in positions [517 – 627] [628 – 800] that extends the two CSB subdomains. XSTREAM, T-REKS and HHrepID could not find any repetitive structure.

4.4 Analysis of UniprotKB/Swiss-Prot

PTRStalker has been run to analyze the database in order to report Steiner Tandem Repeats with edit distance where the level of similarity between the repeated copies and the motif was greater than or equal to 0.7. The output of PTRStalker

Prot acc #	Prot ID	Prot Len	PTRStalker	XSTREAM	T-REKS
Q8IVF2	AHNK2_HUMAN	5795 aa	165-x-24 [720 - 4666]	165-x-23 [774 - 4617]	*
Q9N4M4	ANC1_CAEEL	8545 aa	915-x-6 [3000 - 8491]	903-x-4.27 [4342 - 8199]	58-x-4 [2336 - 2567]
P08519	APOA_HUMAN	4548 aa	1495-x-3 [0 - 4486]	114-x-37 [7 - 4220]	114-x-24 [1501 - 4125]
P20930	FILA_HUMAN	4061 aa	1339-x-3 [32 - 4051]	323-x-11 [268 - 3902]	*
Q54CU4	COLA_DICDI	11103 aa	433-x-17 [1175 - 8554]	430-x-17 [1257 - 8691]	*
Q8R0W0	EPIPL_MOUSE	6548 aa	515-x-8 [2000 - 6548]	515-x-8 [2067 - 6529]	*
Q9Y6R7	FCGBP_HUMAN	5405 aa	1367-x-3 [1000 - 5102]	1201-x-3 [1100 - 4811]	*
P05790	FIBH_BOMMO	5263 aa	1049-x-5 [1 - 5247]	168-x-30 [152 - 5221]	8-x-19 [3362 - 3495]
Q9UKN1	MUC12_HUMAN	5478 aa	1548-x-3 [74 - 4719]	1557-x-2 [446 - 3569]	25-x-8 [2049 - 2280]
Q8WXI7	MUC16_HUMAN	22152 aa	156-x-61 [12038 - 21555]	156-x-61 [12047 - 21567]	156-x-17 [12420 - 15000]
Q6PZE0	MUC19_MOUSE	7524 aa	652-x-9.6 [1071 - 7372]	163-x-36.4 [1281 -7214]	*
Q8WZ42	TITIN_HUMAN	34350 aa	1082-x-4 [22186 - 26525]	28-x-6 [11428 - 11596]	10-x-26 [11445 - 11686]

Table 5: Analysis of the 12 proteins from the UniProtKB/Swiss-Prot database for which a tandem repeat of length ≥ 4000 aa has been detected by PTRStalker. For each protein (row) and each algorithm that returned at least a result (column) we report the longest TR found by each algorithm above the threshold of 100 aa. For each TR we report: the period -x- repeat number and the [interval spanned]. Fail to report is marked with '*'. Note that HHRep and HHRepID are not listed here because they fail to report multi-repeating units, since they only report pairs of homologous substrings.

Prot ID	Prot acc #	Prot Len	PTRStalker	HHRep	HHRepID
dvUT [D. vulgaris DP4]	ABM28909	337 aa	[14 - 165] [177 - 323]	[11 - 91] [174 - 254]	[2 - 138] [141 - 286]
apUT [A. pleuropneumoniae AP76]	YP_001969475	300 aa	[17 - 136] [153 - 284]	[2 - 140] [156 - 288]	[2 - 138] [156 - 286]
UT-A1 [Mus musculus]	AAM00357	930 aa	[4 - 452] [467 - 918]	[63-337] [532-800]	[65 - 493] [533 - 916]
UT-A1 [Homo sapiens]	AAL08485	920 aa	[4 - 320] [403 - 763]	[50 - 338] [519 - 800]	[87 - 490] [548 - 906]

Table 6: Analysis of proteins belonging to the Urea Transporter (UT) family. For each protein (row) and for each algorithm that returned at least a result (column) we report the longest fuzzy repeated subsequence detected by each algorithm by giving start and end position of homologous segments. Note that XSTREAM and T-REKS are not listed here because they are unsuitable for detecting long fuzzy dimeric structures.

Prot ID	Prot acc #	Prot Len	PTRStalker	HHRep
CLC-0 [CICH_TORMA]	P21564	805 aa	[8 – 289] [291 – 575] [517 – 627] [628 – 800]	[119 – 220] [174 – 260] [542 – 597] [718 – 770]

Table 7: Analysis of the CIC-0 protein belonging to the Chloride Channel family. For each algorithm that returned at least a result we report the fuzzy repeated subsequences detected by giving its start and its end position. Note that XSTREAM, T-REKS, and HHrepID are not listed here because they could not find any repetitive structure.

has been stored in a web-accessible database[1]. The website takes as input the accession number of a sequence and shows a table containing all the TRs found in such a sequence. For every TR it displays the start and end position, the length of the motif, the number of repeats, the total length and the consensus string.

This analisys provides an indirect evaluation of PTRStalker with respect to other tools: in [13] T-REKS and other programs are compared by analyzing an old version of the Swiss-Prot database (Release of January 2009), which contained $356,232$ sequences. Among the four evaluated tools (T-REKS, Internal Repeats Finder [17], XSTREAM [22], and TRED [28]), T-REKS is the one that provided the best results by identifying $33,780$ sequences as containing TRs. The definition of TR used by T-REKS is the following: total length of TRs greater or equal to 14 residues (nine residues for homorepeats) and $P_{sim} \geq 0.7$ (average level of similarity between copies and consensus). A direct comparison between PTRStalker and T-REKS would require the analisys of the same version of the Swiss-Prot database. Nevertheless, an indirect comparison can be performed by counting the percentage of sequences containing TRs (found with a given program) against the total number of analyzed sequences (we assume that such percentage is independent from the specific version of the used database). Let us call P_{TR} such value. T-REKS, according to the numbers reported above and derived from [13], obtains $P_{TR} = 9.48\%$, while PTRStalker registers a value of P_{TR} equal to 19.53% (for PTRStalker, we counted only the sequences for which a TR not shorter than 14 residues has been found, independently from the nature of the TR, i.e. homorepeats or not).

[1]The database is freely available at the following address: http://vecchio.iet.unipi.it:8080/PTRStalkerDB.

5. CONCLUSIONS AND FUTURE WORK

Discovering fuzzy tandem repeating units in amino acid sequences gives precious hints as to the internal protein organization and symmetries. However due to high level of protein sequence divergence this task is considered challenging even for relatively short sequences. In this paper we presented a new algorithm, PTRStalker, opportunely tuned for detecting amino acid tandem repeats within protein sequences. We proved that PTRStalker pushes forward the state of the art. Indeed, feeding PTRStalker with sequences from the UniProtKB/Swiss-Prot database did allow us to detect novel repetitive structures not captured by other state-of-the-art tools. In particular we could find a new notable long fuzzy TR in Human Titin. For Chlorine channel protein ClC-0 we showed that PTRStalker can detect general symmetries not detected by competing methods.

Future work will aim at comparing the relative power of different amino acid metric spaces within the PTRStalker framework. In particular we are interested in those based on PAM matrices [31] and those based on the vector space mapping approach [2].

6. ACKNOWLEDGMENTS

We wish to thank Michael Pusch and Laura Gioiosa for their insights on membrane proteins. This work was partially supported by the European Community's Seventh Framework Programme FP7/2007-2013 under grant agreement N 223920 (Virtual Physiological Human Network of Excellence).

7. REFERENCES

[1] Miguel A. Andrade, Carolina Perez-Iratxeta, and Chris P. Ponting. Protein repeats: Structures, functions, and evolution. *Journal of Structural Biology*, 134(2-3):117 – 131, 2001.

[2] William R. Atchley, Jieping Zhao, Andrew D. Fernandes, and Tanja Drüke. Solving the protein sequence metric problem. *Proceedings of the National Academy of Sciences of the United States of America*, 102(18):6395–6400, 2005.

[3] A. Biegert and J. Soding. De novo identification of highly diverged protein repeats by probabilistic consistency. *Bioinformatics*, 24(6):807–814, 2008.

[4] Åsa K Björklund, Diana Ekman, and Arne Elofsson. Expansion of protein domain repeats. *PLoS Comput Biol*, 2(8):e114, 08 2006.

[5] Stefan Burkhardt and Juha Kärkkäinen. Better filtering with gapped q-grams. *Fundam. Inform.*, 56(1-2):51–70, 2003.

[6] E Coward and F Drablos. Detecting periodic patterns in biological sequences. *Bioinformatics*, 14(6):498–507, 1998.

[7] Raimund Dutzler, Ernest B. Campbell, Martine Cadene, Brian T. Chait, and Roderick MacKinnon. X-ray structure of a clc chloride channel at 3.0[thinsp][angst] reveals the molecular basis of anion selectivity. *Nature*, 415(6869):287–294, 2002.

[8] Markus Gruber, Johannes Soding, and Andrei N. Lupas. REPPER-repeats and their periodicities in fibrous proteins. *Nucl. Acids Res.*, 33(suppl_2):W239–243, 2005.

[9] E. Halperin, J. Buhler, R. Karp, R. Krauthgamer, and B. Westover. Detecting protein sequence conservation via metric embeddings. *Bioinformatics*, 19(suppl_1):i122–129, 2003.

[10] Andreas Heger and Liisa Holm. Rapid automatic detection and alignment of repeats in protein sequences. *Proteins: Structure, Function, and Genetics*, 41(2):224–237, 2000.

[11] J. Heringa and Argos P. A method to recognize distant repeats in protein sequences. *Proteins Struct. Func. Genet.*, 17:391 – 411, 1993.

[12] Lucian Ilie and Silvana Ilie. Multiple spaced seeds for homology search. *Bioinformatics*, 23(22):2969–2977, 2007.

[13] Julien Jorda and Andrey V. Kajava. T-REKS: identification of Tandem REpeats in sequences with a K-meanS based algorithm. *Bioinformatics*, 25(20):2632–2638, 2009.

[14] Mridul Kalita, Gowthaman Ramasamy, Sekhar Duraisamy, Virander Chauhan, and Dinesh Gupta. Protrepeatsdb: a database of amino acid repeats in genomes. *BMC Bioinformatics*, 7(1):336, 2006.

[15] S. Karlin and S.F. Altschul. Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes. *Proc. Nat'l Academy of Science USA*, 87(6):2264–2268, 1990.

[16] Bin Ma, John Tromp, and Ming Li. Patternhunter: faster and more sensitive homology search. *Bioinformatics*, 18(3):440–445, 2002.

[17] Edward M. Marcotte, Matteo Pellegrini, Todd O. Yeates, and David Eisenberg. A census of protein repeats. *Journal of Molecular Biology*, 293(1):151 – 160, 1999.

[18] Ranjeet Minocha, Keith Studley, and Milton H. Saier. The urea transporter (ut) family: Bioinformatic analyses leading to structural, functional, and evolutionary predictions. *Receptors and Channels*, 9(6):345–352, 2003.

[19] Daniel P. Miranke. Metric-space search in bioinformatics. *SIGSPATIAL Special*, 2:32–35, July 2010.

[20] Rajeev Motwani and Prabhakar Raghavan. *Randomized Algorithms*. Cambridge University Press, 1995.

[21] K. Mulmuley. *Computational Geometry, an Introduction through Randomized Algorithms*. Prentice Hall, 1993.

[22] Aaron Newman and James Cooper. Xstream: A practical algorithm for identification and architecture modeling of tandem repeats in protein sequences. *BMC Bioinformatics*, 8(1):382, 2007.

[23] Gareth A. Palidwor, Sergey Shcherbinin, Matthew R. Huska, Tamas Rasko, Ulrich Stelzl, Anup Arumughan, Raphaele Foulle, Pablo Porras, Luis Sanchez-Pulido, Erich E. Wanker, and Miguel A. Andrade-Navarro. Detection of alpha-rod protein repeats using a neural network and application to huntingtin. *PLoS Comput Biol*, 5(3):e1000304, 03 2009.

[24] Marco Pellegrini, M. Elena Renda, and Alessio Vecchio. TRStalker: an efficient heuristic for finding fuzzy tandem repeats. *Bioinformatics*, 26(12):i358–366, 2010.

[25] Matteo Pellegrini, Edward M. Marcotte, and Todd O.

Yeates. A fast algorithm for genome-wide analysis of proteins with repeated sequences. *Proteins: Structure, Function, and Genetics*, 35(4):440–446, 1999.

[26] George R.A. and Heringa J. The repro server: finding protein internal sequence repeats through the web. *Trends in Biochemical Sciences*, 25:515–517, 2000.

[27] Johannes Soding, Michael Remmert, and Andreas Biegert. HHrep: de novo protein repeat detection and the origin of TIM barrels. *Nucl. Acids Res.*, 34(suppl_2):W137–142, 2006.

[28] Dina Sokol, Gary Benson, and Justin Tojeira. Tandem repeats over the edit distance. *Bioinformatics*, 23(2):e30–35, 2007.

[29] Radek Szklarczyk and Jaap Heringa. Tracking repeats using significance and transitivity. *Bioinformatics*, 20(suppl_1):i311–317, 2004.

[30] Jinbo Xu, Daniel G. Brown 0001, Ming Li, and Bin Ma. Optimizing multiple spaced seeds for homology search. *Journal of Computational Biology*, 13(7):1355–1368, 2006.

[31] Weijia Xu and Daniel P. Miranker. A metric model of amino acid substitution. *Bioinformatics*, 20(8):1214–1221, 2004.