ITAT 2016 Proceedings, CEUR Workshop Proceedings Vol. 1649, pp. 18–25
http://ceur-ws.org/Vol-1649, Series ISSN 1613-0073, © 2016 M. Macko, M. Králik, B. Brejová, T. Vinař

OB-Fold Recognition Combining Sequence and Structural Motifs

Martin Macko, Martin Králik, Broňa Brejová, and Tomáš Vinař

Faculty of Mathematics, Physics and Informatics, Comenius University in Bratislava
Mlynská dolina, 842 48 Bratislava, Slovakia
martin.macko1@gmail.com, brejova@dcs.fmph.uniba.sk, vinar@fmph.uniba.sk

Abstract: Remote protein homology detection is an important step towards understanding protein function in living organisms. The problem is notoriously difficult; distant homologs can often be detected only by a combination of sequence and structural features.

We propose a new framework, where important sequence and structural features are described by the user in the form of a descriptor, and the descriptor is then used to search a database of protein sequences and score potential candidates. We develop algorithms necessary to support such search using support vector machines and discrete optimization methods. We demonstrate our approach on the example of the telomere-binding OB-fold domain, showing that not only we can distinguish between Telo_bind family members and negatives, but we also identify proteins from related protein families carrying similar OB-fold domains.

Prototype implementation of the descriptor search software is available for Linux operating system at http://compbio.fmph.uniba.sk/descal/

1 Introduction

Remote homology detection is a key to understanding the role of individual proteins in living organisms. This problem is notoriously difficult; the most commonly used tools build profiles from groups of related proteins, representing preferred amino acids at individual loci (e.g. [1, 6, 24]). However, distant homologs are difficult to detect by sequence alone, since the function of a protein is largely determined by its 3D structure. Methods combining structural and sequence-based elements can therefore achieve higher sensitivity [29, 19, 17, 4].

A similar problem is encountered in search for RNA genes, where considering secondary RNA structure is essential to finding distant homologs of known genes. In addition to fully-automated systems for such tasks [9], success was achieved by tools allowing expert human users to handcraft motif descriptors representing the most important features of the target RNAs [8, 5, 22, 28, 21]. Such descriptors specify restrictions on the base-pairing structure of the target RNA (characterizing important secondary structure features), as well as sequence constraints in the form of regular expressions (characterizing important conserved functional sites).

In this work, we propose to extend such descriptor-based approach to protein homology search. However, proteins do not have an equivalent of the simple deterministic rules for RNA base-pairing, and sequence constraints are more naturally written in the form of profiles rather than simple regular expressions. For these reasons, our approach combines techniques from machine learning (support vector machines), probabilistic modeling (sequence profiles), and manual selection of important structural features.

In particular, as the first step, the user creates a descriptor characterizing the most important sequential and structural features of a given protein or a protein family. In the second step, we use our algorithm to score individual proteins (e.g., all proteins in a particular organism) based on how well the descriptor fits these proteins; the score combines sequential features, secondary structure features, and interactions between individual structural elements. Finally, the candidate proteins can be ordered based on this score and the highest scoring candidates will be considered as homologs of the original protein.

Consider an example of the telomere binding OB-fold protein CDC13 in *Saccharomyces cerevisiae*. The important structural elements of this protein have been well characterized [18] and are outlined in Fig. 1. Secondary structure of the telomere-binding OB-fold domain is composed of five β-strands and two α-helices. The β-strands form a typical β-barrel structure. Even though large sequence divergence is typical for this domain, several sequence sites are strongly conserved. To search for putative CDC13 homologs in various species, we propose to describe all these features in a single descriptor, as shown in Fig. 2. By screening a protein database and scoring individual proteins based on this descriptor, we can see that relevant homologs (those containing telomere binding domain) are scored the highest, the proteins from related families have moderate scores, and unrelated proteins have generally low scores (Fig. 5). Thus, the highest scoring proteins are potential candidates for functional homologs of CDC13 in other species.

The paper is organized as follows. First, we describe general framework of descriptors characterizing sequence and structural features of a protein domain and illustrate it on the telomere-binding OB-fold domain. An important feature of these descriptors is identification of potential bonded β-strands. We have developed a support vector machine based classifier for this task. Next, we describe two algorithms for descriptor search in protein sequences. Finally, we evaluate our method on the example

of telomere-binding OB-fold proteins, as outlined in the previous paragraph.

2 Methods

2.1 Protein Domain Descriptors

To search for occurrences of a known protein domain, we propose to characterize the domain in the form of a descriptor inspired by descriptors used in RNA structure search [10, 21]. The main idea is to divide the whole domain into segments corresponding to secondary structure elements; these segments have fixed order along the sequence. Each segment is characterized by the minimum and maximum allowed length and the secondary structure class (α-helix, β-strand or coil). For each segment, it is also possible to provide a short sequence motif in the form of a position-specific scoring matrix (PSSM). An important aspect is the ability to specify interactions between distant segments of the protein. In our descriptor, we allow specification of hydrogen bonds between individual β-strand segments which can be parallel or anti-parallel; we also specify the minimum number of hydrogen bonds.

Most constraints specified by the descriptor are soft; we allow arbitrary consecutive placement of descriptor segments on the query protein subject only to the length constraints. Each such alignment of the descriptor to the protein obtains a score according to the scoring scheme described below, and the score of the protein is the score obtained by the best alignment. All examined proteins are then ranked by their scores, and the user can examine selected proteins from the top of the list, or choose a suitable cutoff score for protein classification.

The scoring scheme consists of three components which are combined to the overall score by a linear combination with suitable weights. The first component measures the agreement of the desired secondary structure elements with the predicted secondary structure of the query protein. The score s_j of segment j placed at positions $k \dots \ell$ is the sum $s_j = \sum_{i=k}^{\ell} \ln(p_i + 0.5)$, where p_i is the posterior probability of the desired secondary structure type at position i. In this way, we prefer alignments that agree with the predicted secondary structure, while at the same time we tolerate unavoidable errors in the secondary structure prediction.

The methods for estimating posterior probabilities of each position in the protein being either α-helix, β-sheet, or a coil have been previously developed, and we use PSIPRED [13] to estimate them.

The second component of the score evaluates the agreement of the sequence with the specified sequence motif in each segment. The motif is given by a PSSM containing a log-odds score for every amino acid at each position within the motif. We use PSSMs extracted from strongly conserved regions of PFAM profiles. In general, the motif is shorter than the minimum segment length, and we use the score of the best-scoring ungapped alignment of the PSSM within the particular sequence segment.

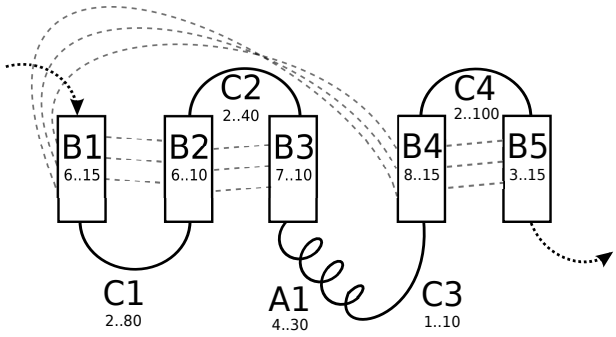

Figure 1: Cartoon representation of the descriptor for the telomere-binding OB-fold domain from Fig.2. Boxes represent β-strands, dotted lines are the required hydrogen bonds. Each segment is annotated with its minimum and maximum allowed length.

Finally, the third score component characterizes the propensity of two β-strands to form hydrogen bonds (we call such β-strands *interacting*). In the next section, we describe a sequence-based classifier that estimates whether two amino acids are likely to form a hydrogen bond in the context of two interacting β-strands; we denote the resulting score for positions i and j as bond(i, j). For a pair of segments required to interact through k parallel hydrogen bonds, we find positions i and j within the segments such that the score $m_{i,j} = \sum_{\ell=0}^{k-1} \text{bond}(i+2\ell, j+2\ell)$ is maximized. (We proceed similarly for anti-parallel interacting β-strand segments.) Note that orientation of amino acids in a β-strand typically alternates, and therefore we skip one position between adjacent bonds. Even though a β-strand as a whole can interact with two other β-strands, we require that each amino acid is involved in at most one hydrogen bond.

Figures 1 and 2 show an example of a descriptor for the telomere binding OB-fold domain. The descriptor contains ten segments, out of which five are β-strands and one is an α-helix. Six of the segments contain sequence motifs corresponding to strongly conserved sections of the Pfam model for the Telo_bind domain. The descriptor also specifies anti-parallel interactions between β-strands forming a β-barrel.

2.2 SVMs Recognizing Interacting β-strands

An important part of the descriptor language is the ability to specify the pattern of hydrogen bonding between individual β-strands that are possibly quite distant in the primary sequence. The use of β-strand interactions has been shown to improve the accuracy of fold recognition in β-strand rich proteins [17, 4]. Several grammar-based methods for recognition of hydrogen bond structure of β-sheets were proposed in the context of protein structure prediction [16, 27, 3]. Another approach to predicting

```
B1|C1|B2|C2|B3|A1|C3|B4|C4|B5
B1: 6 15 B1_LOGO
C1: 2 80
B2: 6 10 B2_LOGO
C2: 2 40 C2_LOGO
B3: 7 10 B3_LOGO
A1: 4 30
C3: 1 10
B4: 8 15 B4_LOGO
C4: 2 100 C4_LOGO
B5: 3 15
- B1 B2: 3
- B2 B3: 3
- B4 B5: 3
- B1 B4: 3

**********LOGO_DEFINITIONS*************
B1_LOGO: [5]
1.4737  0.5703  1.4760  1.1420  ...
1.5272  0.8311  1.2353  1.2990  ...
-0.6192 -0.2024 0.5459  -1.0728 ...
1.1771  0.8718  2.0237  -0.0035 ...
1.0375  0.8607  1.8582  1.3712  ...
B2_LOGO: [3]
-0.7001 -0.7778 -0.9399 0.4729  ...
0.4238  1.1190  0.6647  -0.3823 ...
-1.1356 -0.6612 -0.3739 -1.4812 ...
C2_LOGO: [6]
-1.2004 -1.2746 -1.0686 -1.5443 ...
...
B3_LOGO: [6]
1.4816  1.4929  0.9900  1.1481  ...
...
```

Figure 2: A descriptor of the telomere binding OB-fold protein domain (see also Fig.1). The first line shows the order of individual segments along the sequence and the desired secondary structure of the segments (B for β-strands, A for α-helices, C for coils). The second section contains for each segment its length constraints and an optional PSSM identifier. The third section describes interactions between individual β-strands. Finally, the last section describes PSSMs (each line corresponds to one position and contains 20 log-odds scores of individual amino acids; most numbers were omitted).

the topology of β-sheets and interstrand β-residue pairings uses neural networks [2].

We have created two classifiers to determine if two putative β-strands are likely to form hydrogen bonds, one for parallel and one for anti-parallel strands. The input to each classifier consists of two sequence windows of length 5. The classifier estimates whether the middle amino acids in these windows are likely to form a hydrogen bond with each other. The classifier has the form of a support vector machine (SVM) [26]. We first convert the two sequence windows to a numerical feature vector of length 201. Each sequence position is represented by 20 binary features. One of these features is always set to one and the remaining 19 features are set to zero, depending on the encoded amino acid.

The last feature is the log-odds score for the interaction of the two middle amino acids. In particular, by using our training set, we have estimated frequencies $f_{a,b}$ with which pairs of amino acids a and b occur among hydrogen bonds between interacting β-strands, and frequencies f_x with which amino acids x occur in β-strands individually. If the two amino acids in the middle of the evaluated windows are a and b, then the log-odds score will be defined as $s_{a,b} = \log \frac{f_{a,b}}{f_a f_b}$.

To create a training set, we clustered sequences in the PDB database [23] to clusters with 90% sequence similarity by software CD-Hit [15]. From each cluster we have selected only one sequence for further processing, thus obtaining a representative sample of the proteins in the database. (Without this preprocessing, we would over sample from few large clusters of very similar proteins.)

In addition to the sequence and to the secondary structure annotations (all of which is contained in the PDB database), we also need to determine hydrogen bonds in selected sequences. These were calculated using Jmol Viewer [11]. A positive sample is a pair of sequence windows of length 5 taken from two beta strands of the same protein that have their middle amino acids connected by a hydrogen bond and that have at least one other hydrogen bond between endpoints of the two windows. Parallel or anti-parallel orientation of the windows is determined based on this second bond. A negative sample is a pair of windows from two different β-strands in the same protein that are not connected by any hydrogen bond.

We have selected a random subset of 160,000 positive and 766,239 negative samples for the anti-parallel model and 86,887 positive and 434,435 negative samples for the parallel model. Testing sets for both models contained 15,000 positive and 75,000 negative samples and did not overlap the training set.

By using a small validation set that did not overlap the training or testing set, we have explored a variety of kernels for the SVM by using software SVM-light [12]. Fig.3 shows the accuracy of the models with different SVM kernels. Our final choice was the polynomial kernel of degree 7.

2.3 Descriptor Alignment as an Integer Linear Program

We will call the task of finding the best placement of individual descriptor segments on the query protein sequence the *descriptor alignment problem*. Since the scoring scheme includes long-range interactions between segments, and the positions of interacting segment pairs within the descriptor are not constrained, this problem is NP-hard, similarly to protein threading [14] and RNA descriptor search with pseudoknots [20].

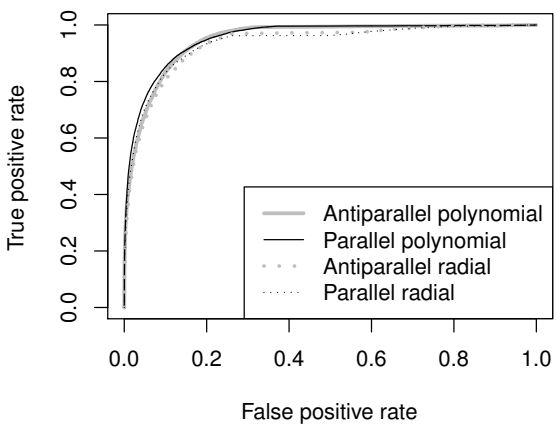

Figure 3: Accurracy of SVM models hydrogen bonds between parallel and antiparallel β-strands. Polynomial kernel of degree 7 used in this work is compared with radial kernel with parameter 0.3.

We therefore formulate the problem as an integer linear program (ILP) and use existing ILP solvers (CPLEX) to find the optimal solution. For simplicity, we show only formulation for parallel interacting β-strands; antiparallel strands are analogous. Our formulation uses the following binary variables:

- Variable x_{is} indicates whether position i is covered by segment s.
- Variable m_{is} indicates whether position i is the starting position of the motif alignment within segment s.
- Variable y_{isjt} indicates whether positions i and j are the first in a chain of hydrogen bonds between (parallel) interacting segments s and t.
- Variable p_{ist} indicates whether position i is involved in a hydrogen bond between segments s and t.

We add hypothetical segments 0 and $m+1$ that fill the gap at the beginning and at the end of the sequence, but do not contribute to the score. The goal of the optimization is to maximize the score for the alignment given by the variables:

$$\sum_{i,s}(e_{is}x_{is}+f_{is}m_{is})+\sum_{i,j,s,t}g_{isjt}y_{isjt}$$

Coefficients e_{is}, f_{is}, and g_{isjt} are precomputed according to the scoring function, including the weights of individual components. The optimization is subject to linear constraints shown in Figure 4. Constraints P_1-P_3 ensure that all the segments are placed consecutively and in the correct order onto the protein sequence. The length of segment is constrained by L_1. The motif occurences must be placed within their corresponding segments (constraints M_1-M_3). The hydrogen bonds between a pair of interacting segments s and t must also lie within these segments (constraints B_1-B_3). Finally, each amino acid can be involved in at most one hydrogen bond (constraints S_1, S_2).

$$\begin{array}{ll}
(P_1) & \forall i : \sum_s x_{is} = 1 \\
(P_2) & \forall i \geq 1, s \geq 1 : x_{is} - x_{i-1,s} \leq x_{i-1,s-1} \\
(P_3) & \forall i \geq 1 : x_{i,0} \leq x_{i-1,0} \\
(L_1) & \forall s : \mathrm{lower}_s \leq \sum_i x_{is} \leq \mathrm{upper}_s \\
(M_1) & \forall s : \sum_i m_{is} = 1 \\
(M_2) & \forall i,s : m_{is} \leq x_{is} \\
(M_3) & \forall i,s : m_{is} \leq x_{i+\mathrm{mot_len}_s-1,s} \\
(B_1) & \forall i,s,j,t, 0 \leq \ell < b_{st} : y_{isjt} \leq x_{i+2\ell,s} \\
(B_2) & \forall i,s,j,t, 0 \leq \ell < b_{st} : y_{isjt} \leq x_{j+2\ell,t} \\
(B_3) & \forall s,t : \sum_{i,j} y_{isjt} = 1 \\
(S_1) & \forall i,s,t : p_{ist} = \sum_{\ell=0}^{b_{st}-1} \sum_j y_{i-2\ell,s,j,t} \\
(S_2) & \forall i : \sum_{s,t} p_{ist} \leq 1
\end{array}$$

Figure 4: Linear constraints in the integer linear program for the descriptor alignment problem. In these constraints, s and t are segments, i and j are positions in the sequence, lower_s and upper_s are bounds on the segment length, $\mathrm{mot\_len}_s$ is the motif length, b_{st} is the number of hydrogen bonds between s and t.

We have tested variants of this ILP formulation for several proteins. Short positive examples can be typically solved within minutes. However, the running time for negative examples was usually quite high, and therefore we were not able to complete more extensive tests with this approach. Instead, we propose a dynamic programming algorithm for a slightly simplified version of the problem as described in the next section.

2.4 Efficient Dynamic Programming Algorithm

Since the descriptor alignment problem is NP-hard for general conformation of interacting segment pairs, we will solve a special case where the interactions within the descriptor are limited. In particular, if segments s_1 and s_2 interact, we do not allow any interactions for segment s such that $s_1 < s < s_2$. Interacting pairs form chains of the form (s_1,s_2), (s_2,s_3),..., (s_{k-1},s_k), and different chains occupy disjoint regions of the descriptor. The descriptor in Fig.2 does not satisfy this restriction because segments B_2 and B_3 interact, and they lie within interacting pair (B_1,B_4). If we remove pair (B_1,B_4), the remaining interactions satisfy the restriction and form two chains $[(B_1,B_2),(B_2,B_3)]$ and $[(B_4,B_5)]$.

We will show that under this restriction, the alignment problem can be solved optimally in polynomial time by dynamic programming. First, we will consider two simpler problems. If there are no interactions in the descriptor, we can use straightforward dynamic programming as follows. Let $A[s,t]$ be the score of the best alignment of the first s segments of the descriptor, where last of them ends at the sequence position t. This value can be computed using the values for the first $s-1$ segments ending

at positions $t' < t$: $A[s,t] = \max_f A[s-1, f-1] + S[s,f,t]$. In this formula, $S[s,f,t]$ is the segment score for segment s extending from position f to t. This score includes the secondary structure and sequence motif scores, combined with appropriate weights.

We will now extend this dynamic programming to the case where we allow restricted interaction configurations as described above. However, we will not enforce the condition that a single amino acid can only be used in a single hydrogen bond. To accommodate interactions, we need to include the score for the best placement of hydrogen bonds between interacting segments. Let $B[s,f',t',f,t]$ be the interaction score between segment s and its interaction partner $s' < s$ if s extends from f to t and s' extends from f' to t'. This score is precomputed by finding the highest scoring position of hydrogen bonds within the two segments. In order to incorporate such interaction scores to our dynamic programming, we need to increase the dimension of matrix A to keep track of the position of s'.

If segments s_1 and s_2 interact and s is a segment such that $s_1 \leq s < s_2$, we say that s_1 is an *open segment* for s. Under our restriction on descriptors, each segment s has at most one open segment. We can define the subproblem of the dynamic programming $A[s,t,f',t']$ as the score of the best alignment of the first s segments of the descriptor, where segment s ends at position t and the open segment for s starts at f' and ends at t'. We compute this value from values for $s-1$, distingushing the following four cases.

In the first case, s interacts with two other segments s_1 and s_2, where $s_1 < s < s_2$. Then s is its own open segment, and therefore s starts at f' and ends at $t' = t$. We maximize over possible values f'' and t'' that represent start and end of segment s_1 (which is the open segment for $s-1$): $\max_{f'',t''} A[s-1, f'-1, f'', t''] + S[s,f',t'] + B[s,f'',t'',f',t']$.

In the second case, s interacts with one segment s_1, where $s_1 < s$. Then s does not have an open segment, and we only consider values $f' = t' = \perp$. We maximize over all possible values of f, f'' and t'', where f'' and t'' represent start and end of segment s_1, and f is the start of segment s: $\max_{f'',t'',f} A[s-1, f-1, f'', t''] + S[s,f,t] + B[s,f'',t'',f,t]$.

In the third case, s interacts with one segment s_2, where $s_2 > s$. Again, s is its own open segment, and therefore we require $t = t'$. On the other hand, $s-1$ does not have an open segment, and thus we do not need to maximize over any values, obtaining the equation $A[s,t,f',t'] = A[s-1, f'-1, \perp, \perp] + S[s,f',t]$.

The last case occurs when s does not interact with any segment. It may have an open segment $s' < s$, which is then also the open segment for $s-1$, or it does not have any open segment, which means that $f' = t' = \perp$. We maximize over all possible starts f of segment s: $\max_f A[s-1, f-1, f', t'] + S[s,f,t]$.

Finally, we further extend our algorithm to enforce that each amino acid is involved in at most one hydrogen bond. Let (s_1,s_2) and (s_2,s_3) be two interacting pairs of segments sharing segment s_2. When choosing bond positions for (s_2,s_3), we need to know which positions were already used for bonds in (s_1,s_2) and thus cannot be used again. To do this, we introduce new parameter b' into our table $A[s,t,f',t',b']$. Parameter b' is the position of the first hydrogen bond within the open segment s_1 of segment s. Other positions in s_1 used by bonds can be determined based on the required number of bonds and orientation specified in the descriptor. Computation of values in table A needs to distinguish six cases depending on the type of segment s, similarly as before. The two extra cases arise from the need to keep track whether the open segment for segment $s-1$ has restricted positions or not. We omit the full recurrence, which can be derived by carefully extending the formulas above.

The running time of the algorithm is $O(nmf\ell^4)$, where n is the length of the protein sequence, m is the number of segments in the descriptor, ℓ is the maximum segment length, and f is the maximum flexibility of interacting pairs defined as the difference between the smallest and the largest possible distance between their ends t and t'.

2.5 Implementation Details

To compute interaction scores, we need to run the SVM predictor for all pairs of windows of length 5 that can be covered by interacting β-strands. In the worst case, the number of such pairs grows quadratically with the protein length, although in practice the flexibility of the descriptor is limited, thus bounding achievable distance of these pairs. Nonetheless, computation of all required SVM values was the most time-consuming part of the dynamic programing solution. Therefore we have added a heuristic rule which allows hydrogen bonds only between amino acids that have posterior probability of β-strand secondary structure from PSIPRED at least 0.5. This rule has dramatically lowered the computation time. As we will see in the next section, many negative examples do not have enough potential placements for hydrogen bonds, and as a result, no alignment of the descriptor is possible. On the other hand, this situation happens only very rarely for positive examples.

Our scoring scheme allows us to assign different weights to the three components. Ideally, these weights would be optimized to improve the prediction accuracy. For simplicity, we have used weight 1 for secondary structure and sequence motifs and weight 2 for interactions. The weight of interactions was increased because values produced by the SVM were relatively small compared to the overall score.

In order to comply with constrains imposed by the dynamic programming, we omit the interaction between segments B_1 and B_4 from the descriptor of the OB-fold protein domain shown in Fig.2. After computing solution for the reduced descriptor with the dynamic programming, we simply try every possible position for the B_1-B_4 interaction and include the best one in the overall score.

3 Results

To evaluate our descriptor approach, we have used the descriptor in Fig.2 and our dynamic programming algorithm to recognize proteins containing the telomere-binding OB-fold domain. Note that in the dynamic programming, we omit some of the interacting pairs to make the problem tractable. Even though part of the score corresponding to missing interactions is later added to the final score, the alignment of the descriptor obtained by the dynamic programming may not be optimal.

We have randomly selected 50 proteins with telomere-binding OB-fold domain annotated in Pfam (Pfam domain PF02765 Telo_bind). We have also randomly chosen 50 SWISS-PROT proteins not associated with the PF02765 family as a negative sample. Finally, we have selected four other families from the OB-fold clan: RNA polymerase Rpb8 family (PF03870 Rpb8), single-strand binding protein family (PF00436 SSB), tRNA binding domain (PF01588 tRna_bind), and eukaryotic elongation factor 5A hypusine (PF01287 elF-5a). From each of these four families, we have randomly chosen 20 proteins.

The results are summarized in Fig.5. Our descriptor can reliably recognize Telo_bind proteins from the negative samples. Only three negative samples had score higher than 20, with the largest score 38.9. Two positives scored less than 35, additional two proteins were filtered out in the secondary-structure filtering. HMMer [7] achieved perfect separation between Telo_bind and negatives (Fig.5c), but this comparison is not fair since the annotation of protein domains in Pfam is based on the same profile HMM which was used in this test, and therefore it is not surprising that it achieves perfect classification.

Our goal is, however, to search for distant homologs that cannot be reliably recognized by Pfam profiles. The descriptor search is able to recognize proteins from related families that also contain OB-folds, and yet at the same time, it is possible to distinguish them quite successfully from Telo_bind proteins. On the other hand, HMMer results cannot distinguish these four additional families from negatives (compare Fig.5b and c). These results suggest that it is sensible to use the descriptor search to locate distant homologs, examining the resulting candidates in order of the assigned scores. This can be especially beneficial when sequence-based methods (such as HMMer) fail to find any matches.

Pot1 and CDC13 are OB-fold telomere binding proteins that bind single-stranded telomere overhang and are key players in telomere maintenance. It has been a long standing question, which protein performs this crucial role in the pathogenic yeast *Candida albicans* and related species. No homolog could be found by common sequence-based methods [25]. Yu *et al.* [30] have demonstrated that a short protein (Uniprot ID Q5AB98) associates with telomere DNA and regulates telomere lengths. They postulate that this is the missing ortholog of CDC13. Search with our descriptor against this protein produced score of 32.8, which is on the low end of the range for Telo_bind and well within range of other OB-fold containing families. Note that search for Pfam domains in this protein does not return any significant matches.

4 Conclusion

In this paper, we have introduced a framework of combining sequence and structural information in search of distant protein homologs. Important sequence and structural features of a given protein or a protein family are first manually selected and described in the form of a descriptor which is then used to search a database of protein sequences and score potential candidates.

We have demonstrated the use of our framework on the telomere-binding OB-fold domain. Based on the description of the *S. cerevisiae* CDC13 protein by Mitton-Fry *et al.* [18], we have created a descriptor that includes the information on the secondary structure elements, interaction of individual β-strands, and highly conserved sequence motifs. We have developed an algorithm that allowed us to score individual proteins with this descriptor, and we have demonstrated that not only the descriptor search was able to distinguish between Telo_bind family members and negative examples, but it also identified proteins from related families containing similar domains.

There are many avenues for further research in this area. First, even though our algorithms are universal, we have mostly targeted the features required to support CDC13 distant homolog search. There are many other features that could be included within the same algorithmic framework (e.g., more flexible sequence motifs, irregular hydrogen bond configurations, flexible distances between individual elements), while others would require development of new algorithms (e.g., more complex interaction models between segments).

The experience from a similar RNA search framework suggests that writing sensitive and specific descriptors is a long iterative process. Our Telo_bind descriptor is only the first attempt at this task, further examination of results could suggest which features are perhaps less important and could be omitted, and which new features should be included instead. Continuing this work could lead to a discovery of telomere binding OB-fold proteins in species where these proteins are yet unknown, and also to greater understanding of importance of individual features of this protein. Development of additional tools supporting such research would be of great interest.

The scoring function of the descriptor alignment to a protein is a linear combination of several components. The overall score is optimized globally, however, the weights controlling individual contributions of the components were chosen ad hoc. Systematic choice of these constants, perhaps through machine learning methods, could lead to higher accuracy.

One obstacle to a wider deployment of our current

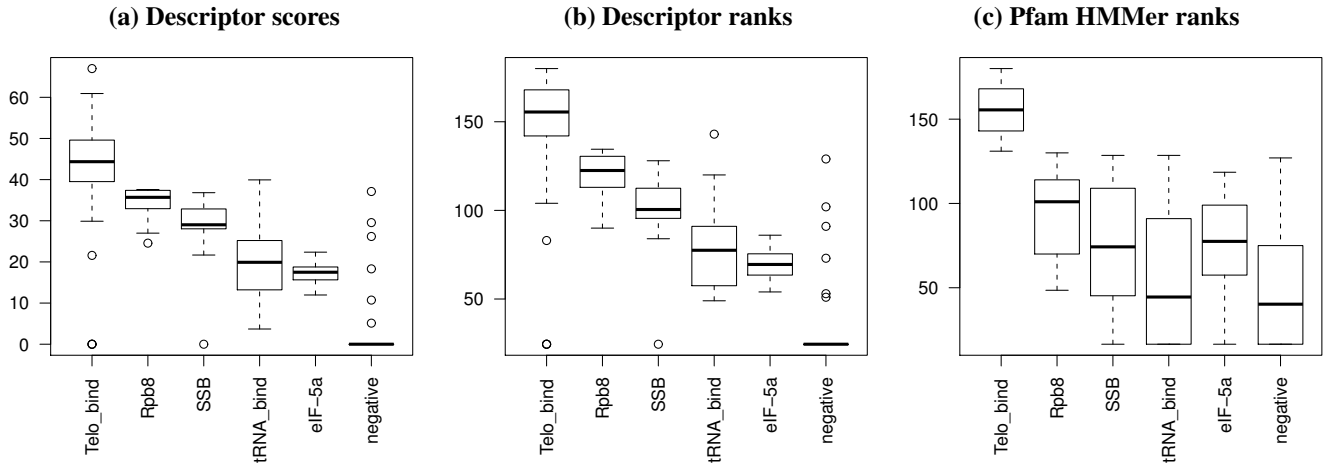

Figure 5: Recognizing a variety of OB-fold families using the OB-fold descriptor and HMMer search with the Pfam model of the Telo_bind domain. Each box plot shows score or rank distributions for one method within different OB-fold families and negative samples. Both the descriptor and the Pfam model can reliably distinguish between telomere binding OB-fold proteins and negative samples. Descriptor approach can, however, also identify proteins from related protein families that also contain an OB-fold domain.

search tool is its running time. The alignment of the descriptor to a single protein requires anything between couple of seconds to several hours. In the dynamic programming, we have sacrificed information provided through one of the β-strand interactions, and we have further restricted the search space by discarding segment positions that did not match the secondary structure constraints well. Yet, we believe that these relaxations changed the final result very little. Perhaps further heuristic relaxations and approximations could lead to a faster search tools.

Finally, one could imagine that efforts towards assembling a database of descriptors characterizing common protein functions could lead to a better and faster functional annotation of newly sequenced species.

Acknowledgements. This research was funded by APVV grant APVV-14-0253 and VEGA grants 1/0719/14 (TV) and 1/0684/16 (BB).

References

[1] Altschul, S. F., Madden, T. L., Schaffer, A. A., Zhang, J., Zhang, Z., Miller, W., and Lipman, D. J. (1997). Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res*, **25**(17), 3389–3392.

[2] Cheng, J. and Baldi, P. (2005). Three-stage prediction of protein β-sheets by neural networks, alignments and graph algorithms. *Bioinformatics*, **21**(suppl 1), i75–i84.

[3] Chiang, D., Joshi, A., and Searls, D. (2006). Grammatical representations of macromolecular structure. *Journal of Computational Biology*, **13**(5), 1077–1100.

[4] Daniels, N. M., Hosur, R., Berger, B., and Cowen, L. J. (2012). SMURFLite: combining simplified Markov random fields with simulated evolution improves remote homology detection for beta-structural proteins into the twilight zone. *Bioinformatics*.

[5] Eddy, S. R. (1996). RNABob: a program to search for rna secondary structure motifs in sequence databases. unpublished.

[6] Eddy, S. R. (2009). A new generation of homology search tools based on probabilistic inference. *Genome Inform*, **23**(1), 205–211.

[7] Eddy, S. R. (2011). Accelerated Profile HMM Searches. *PLoS Comput Biol*, **7**(10), e1002195.

[8] Gautheret, D., Major, F., and Cedergren, R. (1990). Pattern searching/alignment with RNA primary and secondary structures: an effective descriptor for tRNA. *Comput Appl Biosci*, **6**(4), 325–331.

[9] Griffiths-Jones, S., Bateman, A., Marshall, M., Khanna, A., and Eddy, S. R. (2003). Rfam: an RNA family database. *Nucleic Acids Res*, **31**(1), 439–441.

[10] Jimenez, R. M., Rampasek, L., Brejova, B., Vinar, T., and Luptak, A. (2012). Discovery of RNA motifs using a computational pipeline that allows insertions in paired regions and filtering of candidate sequences. *Methods Mol Biol*, **848**, 145–148.

[11] Jmol (2012). Jmol: an open-source Java viewer for chemical structures in 3D. www.jmol.org.

[12] Joachims, T. (1999). Making large-scale SVM learning practical. In B. Schölkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*. MIT-Press.

[13] Jones, D. T. (1999). Protein secondary structure prediction based on position-specific scoring matrices. *J Mol Biol*, **292**(2), 195–202.

[14] Lathrop, R. H. (1994). The protein threading problem with sequence amino acid interaction preferences is NP-complete. *Protein Eng*, **7**(9), 1059–1068.

[15] Li, W. and Godzik, A. (2006). Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences. *Bioinformatics*, **22**(13), 1658–1659.

[16] Mamitsuka, H. and Abe, N. (1994). Predicting location and structure of beta-sheet regions using stochastic tree grammars. *ISMB-94*, pages 276–284.

[17] Menke, M., Berger, B., and Cowen, L. (2010). Markov random fields reveal an N-terminal double beta-propeller motif as part of a bacterial hybrid two-component sensor system. *Proc Natl Acad Sci U S A*, **107**(9), 4069–4074.

[18] Mitton-Fry, R. M., Anderson, E. M., Theobald, D. L., Glustrom, L. W., and Wuttke, D. S. (2004). Structural basis for telomeric single-stranded DNA recognition by yeast Cdc13. *J Mol Biol*, **338**(2), 241–245.

[19] Nielsen, M., Lundegaard, C., Lund, O., and Petersen, T. N. (2010). CPHmodels-3.0–remote homology modeling using structure-guided sequence profiles. *Nucleic Acids Res*, **38**(Web Server issue), W576–581.

[20] Rampasek, L. (2011). RNA structural motif search is NP-complete. In *Studentska vedecka konferencia FMFI UK, Bratislava*, pages 341–348.

[21] Rampasek, L., Jimenez, R. M., Luptak, A., Vinar, T., and Brejova, B. (2016). RNA motif search with data-driven element ordering. *BMC Bioinformatics*, **17**(1), 216.

[22] Reeder, J., Reeder, J., and Giegerich, R. (2007). Locomotif: from graphical motif description to RNA motif search. *Bioinformatics*, **23**(13), i392–400.

[23] Rose, P. W. *et al.* (2011). The RCSB Protein Data Bank: redesigned web site and web services. *Nucleic Acids Res*, **39**(Database issue), D392–401.

[24] Sadreyev, R. I. and Grishin, N. V. (2008). Accurate statistical model of comparison between multiple sequence alignments. *Nucleic Acids Res*, **36**(7), 2240–2248.

[25] Teixeira, M. T. and Gilson, E. (2005). Telomere maintenance, function and evolution: the yeast paradigm. *Chromosome Res*, **13**(5), 535–538.

[26] Vapnik, V. N. (1995). *The Nature of Statistical Learning Theory*. Springer.

[27] Waldispühl, J., Berger, B., Clote, P., and Steyaert, J. (2006). Predicting transmembrane β-barrels and interstrand residue interactions from sequence. *PROTEINS: Structure, Function, and Bioinformatics*, **65**(1), 61–74.

[28] Webb, C.-H. T., Riccitelli, N. J., Ruminski, D. J., and Luptak, A. (2009). Widespread occurrence of self-cleaving ribozymes. *Science*, **326**(5955), 953.

[29] Xu, J., Li, M., Kim, D., and Xu, Y. (2003). RAPTOR: optimal protein threading by linear programming. *J Bioinform Comput Biol*, **1**(1), 95–117.

[30] Yu, E. Y., Sun, J., Lei, M., and Lue, N. F. (2012). Analyses of Candida Cdc13 orthologues revealed a novel OB fold dimer arrangement, dimerization-assisted DNA binding, and substantial structural differences between Cdc13 and RPA70. *Mol Cell Biol*, **32**(1), 186–188.