PROTEIN FOLD RECOGNITION USING RESIDUE-BASED ALIGNMENTS OF SEQUENCE AND SECONDARY STRUCTURE

*Zafer Aydin**, *Hakan Erdogan***, *and Yucel Altunbasak**

*Center for Signal and Image Processing,
Centergy 5^{th} floor, Georgia Institute of Technology
Atlanta, GA 30332-0250
E-mail:{aydinz,yucel}@ece.gatech.edu

**Faculty of Engineering and Natural Sciences
Room 1097, Sabanci University
Orhanli, Tuzla, 34956 Istanbul, Turkey
E-mail: haerdogan@sabanciuniv.edu

ABSTRACT

Protein structure prediction aims to determine the three-dimensional structure of proteins form their amino acid sequences. When a protein does not have similarity (homology) to any known fold, threading or fold recognition methods are used to predict structure. Fold recognition methods frequently employ secondary structure, solvent accessibility, and evolutionary information to enhance the accuracy and the quality of the predictions.

In this paper, we present a residue based alignment method as an alternative to the state-of-the-art SSEA method, originally introduced by Przytycka *et al.* [1], and further modified by McGuffin *et al.* [2]. We introduce a residue-based score function, which can incorporate amino acid similarity matrices such as BLOSUM into secondary structure similarity scoring and compute joint alignments. We show that the power of the SSEA method comes from the length normalization instead of the element alignment technique and similar performance can be achieved using residue-based alignments of secondary structures by optimizing gap costs. In simulations with the two benchmark datasets, our method performs slightly better than the SSEA in terms of the fold recognition accuracy. When the secondary structure similarity matrix is combined with the amino acid based BLOSUM30 matrix, the accuracy of our method improves further (4% for the McGuffin set and 10% for the Ding and Dubchak set). The availability of aligning the amino acid and secondary structure sequences in a joint manner offers a better starting point for more elaborate techniques that employ profile-profile alignments and machine learning methods [3, 4].

Index Terms— protein fold recognition, secondary structure alignment, amino acid alignment, score normalization, gap cost.

1. INTRODUCTION

Protein structure prediction aims to determine the three-dimensional structure of proteins form their amino acid sequences. When a protein shows significant sequence similarity to other proteins with known structures, comparative modeling techniques can be used to determine the three-dimensional structure with reasonable accuracy. On the other hand, if there is little similarity (homology) to any known fold, threading or fold recognition methods are used to predict structure.

Protein threading or fold recognition refers to a class of computational methods for predicting the structure of a protein from the amino acid sequence. The basic idea is that the target sequence (the protein sequence for which the structure is being predicted) is threaded through the backbone structures of a collection of template proteins (known as the fold library) and a goodness of fit score calculated for each sequence-structure alignment. Protein fold recognition problem can be stated as the problem of assigning a protein of unknown structure (target) to one of the known fold classes (templates) as defined in the SCOP or CATH classification standards. Fold recognition methods can be broadly divided into two types: (1) methods that derive a 1-D profile for each structure in the fold library and align the target sequence to these profiles; (2) methods that consider the full 3-D structure of the protein template. A simple example of a profile representation would be to take each amino acid in the structure and simply label it according to whether it is buried in the core of the protein or exposed on the surface. More elaborate profiles might take into account the local secondary structure (e.g. whether the amino acid is part of an alpha helix) or even evolutionary information (how conserved the amino acid is). In the 3-D representation, the structure is modelled as a set of inter-atomic distances *i.e.*, the distances are calculated between some or all of the atom pairs in the structure. This is a much richer and far more flexible description of the structure, but is much harder to use in calculating an alignment. Also note that, methods in the second category greatly benefit from the profile-based approaches.

Recent approaches in fold recognition follow two major directions, namely machine learning methods (neural networks and support vector machines) and alignment methods. In this paper, we present a residue based alignment method as an alternative to the state-of-the-art SSEA method, originally introduced by Przytycka *et al.* [1], and further modified by McGuffin *et al.* [2]. We introduce a score function, which allows us to incorporate amino acid similarity scores such as BLOSUM into secondary structure alignments, which is not possible with the SSEA method. By combining the amino acid and the secondary structure similarity matrices, it is possible to compute joint alignments of sequence and secondary structure.

2. SECONDARY STRUCTURE ELEMENT ALIGNMENT (SSEA)

Secondary structure element alignment (SSEA) was first introduced by Przytycka *et al.* [1], which is based on the alignment of secondary structure segments instead of the residue pairs. McGuffin *et al.* [2] then adopted the idea and compared a slightly modified SSEA to other alignment methods. The result was that SSEA performed the best of all tested sequence and secondary structure alignment methods in predicting the fold class of a given protein. The pairwise alignment procedure described by McGuffin *et al.* [2] can be summarized as follows:

1. Represent each sequence as the sequence of secondary structure elements and annotate the length of the elements. Discard leading and trailing coils. For instance, the secondary

structure string LLLLLHHHHHHLLEEELLLHHHHHLLL is represented by: (HLELH, 62335), where HLELH is the secondary structure element representation and the numbers 6, 2, 3, 3, 5 code for the length information of each secondary structure element (or segment).

2. Align the two element sequences using dynamic programming with zero gap costs. The scoring function is defined as follows. Matching elements (H-H, E-E, and L-L) are scored by the minimum length of the aligned elements, H-L and E-L mismatches are scored by half the minimum length, and finally H-E mismatch scores 0. The total alignment score ($rawscore$) is the sum of all aligned element-pair scores.
3. Normalize the alignment score by the mean trimmed sequence length (sequence length minus initial and final coil regions) of the two proteins. If l_1, and l_2 denote the lengths of the trimmed secondary structure sequences, the normalized score is computed as $2\frac{rawscore}{l_1+l_2}$. Here, the lengths are obtained over the sequences in the original form, instead of the element representation. For instance, the lengths of the two element sequences (HLELH, 62335) and (HLHLH, 51796), become $l_1 = 19$, and $l_2 = 28$, respectively.

A more detailed description of the SSEA method, its possible variants and other alignment techniques used in the McGuffin evaluation can be found in http://www.cs.ucl.ac.uk/staff/L.McGuffin/methods.html.

3. RESIDUE-BASED ALIGNMENTS OF SEQUENCE AND SECONDARY STRUCTURE

In residue-based alignments of two secondary structure sequences, one is concerned with finding the optimum pairing of the secondary structure symbols instead of the secondary structure elements or segments. Therefore it is necessary to first define the similarity matrix to score the matches and mismatches for a pair of secondary structure symbols.

3.1. Secondary Structure Similarity Matrix

In our method, we used the similarity matrix shown in Table 1, which is introduced in Wallqvist *et al.* [5]. This matrix is obtained from a

M_{ss}	H	E	L
H	2	-15	-4
E	-15	4	-4
L	-4	-4	2

Table 1. Secondary structure similarity matrix M_{ss}, obtained from the 3D_ali database which is used to score the alignment of two secondary structure symbols.

set of representative 3-D structure alignments provided in the 3D_ali database and reflects the occurrence propensities of secondary structure symbols paired in the alignments, *i.e.* how often a helical residue is paired with another helical residue. Here, the matrix elements M_{ss}^{ij}, are defined by

$$M_{ss}^{ij} = 2log_2(\frac{P_{ij}}{P_{ij}^{ex}}), \quad (1)$$

where P_{ij} is the probability of finding the paired structural elements i and j in an alignment of two secondary structure sequences, and P_{ij}^{ex} is the probability of finding the same pair in an alignment of two random sequences. Therefore, the matrix elements provide a measure of how often a pairing occurs relative to the random case, where a positive value indicates a favorable score. For a more detailed description of how the probability terms P_{ij}, P_{ij}^{ex}, and the similarity matrix is computed see Wallqvist *et al.* [5].

3.2. Gap Scoring

Having defined the similarity matrix, we will proceed with the gap scoring function. When a secondary structure symbol (H, E, or L) in one sequence does not have any counterpart (or match) in the other sequence, then that symbol is aligned to a gap symbol '-'. Allowing gap regions in an alignment enables us to better represent the similarity between the aligned sequences in a biologically meaningful manner. In the state-of-the-art gap scoring, opening a gap is penalized more than extending it. For example in "affine gap scoring", which is one of the most widely used gap scoring techniques, starting a gap is scored by parameter d, and extending a gap region is scored by e, where d typically takes values around -10 or -12 and e is set to -1 or -2.

In this paper, we implemented an affine gap scoring, where the parameters d and e are optimized by searching for the values that maximize the fold recognition accuracy (see Section 4 for details).

3.3. Score Function with Amino Acid Similarity Matrix

The scoring function to align a pair of secondary structure sequences A and B can be defined as:

$$\phi(A,B) = \sum_{k=1}^{m_{ab}} M_{ss}^{a_k,b_k} + N_o g_o + N_e g_e, \quad (2)$$

where M_{ss} corresponds to the secondary structure similarity matrix, m_{ab} is the number of paired elements in the alignment between sequences A and B, and a_k, b_k denote the k^{th} secondary structure pair of the aligned sequences. In this equation, the number of gap openings N_o is multiplied by the gap opening penalty, g_o, and the number of gap extensions N_e is multiplied by the gap extension penalty g_e.

The scoring function defined in Eq. 2 can be extended to incorporate the amino acid similarity matrix as follows:

$$\phi_{\alpha\beta}(A,B) = \sum_{k=1}^{m_{ab}} (\alpha M_{aa}^{c_k,d_k} + \beta M_{ss}^{a_k,b_k}) + N_o g_o + N_e g_e, \quad (3)$$

where M_{aa} denote the amino acid similarity matrix (BLOSUM50 or BLOSUM30), c_k, d_k represent the k^{th} amino acid pair of the aligned sequences, and α, β determine the weighted importance of the amino acid and secondary structure similarity scores, respectively.

3.4. Content Dependent Score Function

Typically, α and β are set to 0.5 so that equal weights are assigned to the amino acid and secondary structure similarity scores. In a more realistic setting, one can consider the fact that secondary structures are obtained as predictions and most of the prediction algorithms assign a confidence value to each position. Therefore the weights α, and β could be adjusted in such a way that higher weight is given to the secondary structure similarity score, $M_{ss}^{a_k,b_k}$, when the prediction confidence is high for a_k and b_k and vice versa. To model this, Eq. 3 can be rephrased as follows:

$$\phi_{\alpha\beta}(A,B) = \sum_{k=1}^{m_{ab}} (\alpha^{A,B} M_{aa}^{c_k,c_k} + \beta^{A,B} M_{ss}^{a_k,b_k}) + G, \quad (4)$$

where $\alpha^{A,B}$ and $\beta^{A,B}$ are the content dependent weights and G is the gap penalty, which is equal to $N_o g_o + N_e g_e$. To investigate the effect of this approach, we implemented several functions for choosing $\alpha^{A,B}$ and $\beta^{A,B}$. However, we did not observe any improvement in the fold recognition accuracy when the weights are adjusted according to the prediction confidence values (data not shown). Therefore, we have concluded that the confidence values of a single secondary structure prediction method is not informative when applied to fold recognition problem, and the best performance is achieved by choosing $\alpha^{A,B} = \beta^{A,B} = c$, where c is a constant. An interesting avenue could be to utilize the prediction confidence values of multiple predictors in an attempt to improve the secondary structure prediction accuracy and thus the fold recognition performance.

3.5. Score Normalization

After computing the raw score of an alignment, it is useful to normalize it to a statistically meaningful range. This procedure helps to assign a statistical significance to an alignment score by estimating the likelihood of that score to arise by chance. In this paper, we considered using two normalization procedures: (1) Conversion to E-values; (2) Normalization by the average length of the aligned sequences as described in Section 2. The E-value of an alignment is computed as $E = Kmne^{-\lambda S}$, where K and λ are constant parameters, m, n are the sequence lengths and S is the alignment score. Empirical estimates of K and λ at different gap costs is given in Altschul and Gish [6]. Here, we simply chose values from Tables V and VI of Altschul and Gish [6] corresponding to the gap scores being used.

3.6. Computation of the Best Scoring Alignment

Given a scoring function, the computation of the optimum (best scoring) alignment can be found using the dynamic programming techniques. In this paper, we used the Smith-Waterman algorithm, to compute the local alignment between a pair of sequences. More details on the alignment algorithms and dynamic programming can be found in Durbin *et al.* [7].

4. RESULTS

In our simulations, we used two benchmark datasets. The first one is introduced by McGuffin and Jones [8] and is a "difficult" set. It contains 542 non-redundant domains based on CATH [9] version 1.7 and is divided into a subset of 252 known folds which have at least one other match in the set, and 290 unique folds, *i.e.*, domains which have folds unique with respect to this set. In order to evaluate the performance of our method, we selected the set of known folds as targets and the complete set as templates, excluding identical hits. Then we aligned each target to all templates, and compared the fold classes of the maximum scoring target-template pair. If both belonged to the same fold class, then this alignment is counted as a successful prediction. In this evaluation, the fold class assignments are taken from CATH V3.0. The second set is provided by Ding and Dubchak [10] and is relatively easy. It contains 386 SCOP domains in 27 SCOP folds. This set is known to contain distant homologues [11], a fact that leads to higher recognition rate for such target-template pairs. To evaluate the performance of our method on this benchmark set, we performed all-against-all alignments (leave-one-out test), in which a domain is chosen from the set as the target and is aligned to the remaining domains, which form the template library. Then the alignments are sorted and fold classes of the maximum scoring target-template pair are compared. This process is repeated until all domains are chosen as targets and aligned to the remaining set of templates.

In all simulations, we used sensitivity as the performance measure, which is defined as:

$$Q = \frac{N_c}{N}, \tag{5}$$

where N_c is the number of targets with correctly predicted fold classes, and N is the total number of targets evaluated.

In simulations with score normalization and gap score optimization, secondary structure assignments are taken from the PDB (Protein Data Bank, http://www.pdb.org). Note that, PDB uses a version of the DSSP algorithm to assign secondary structure from atomic coordinates of experimentally solved proteins. In simulations comparing the performance of our method to SSEA, and evaluating the performance of amino acid and secondary structure alignments, secondary structures are predicted using PSIPRED version 2.4.

4.1. Score Normalization Techniques

In this section, we considered three alternatives for the score normalization: (1) no normalization (raw scores); (2) normalization by the average length; (3) normalization by converting to e-values. We performed all-against-all alignments of secondary structure sequences on the Ding and Dubchak set [10], where the secondary structure assignments are taken from the PDB. Here, the score function in Eq. 1 is used, with gap opening and gap extension penalties are set to $d = -12$ and $e = -2$. In E-value conversion, the parameters K, and λ are set to 0.09, and 0.3, respectively (see Table V in Altschul and Gish [6]). Although these values are estimated for the BLOSUM62 matrix, we believe that the optimization of K and λ for the SSSM matrix will not bring significant improvements. From the results in Table 2, the average length normalization gives the best results in predicting the fold class. Another interesting observation is that when there is no normalization, the fold recognition accuracy drops significantly. This indicates that the true power of the SSEA method comes from the score normalization but not from the element alignment technique. To further validate this hypothesis, we performed the same simulations using the SSEA method (e-value normalization is not defined for SSEA). From the results shown in Table 3, we concluded that the element alignment approach does not produce satisfactory performance without score normalization. In the following sections, we will show that for optimized values of d and e, residue-based alignment approach performs comparably better than the SSEA method.

Normalization Method	$Q(\%)$
Raw Scores	38.45
E-value	52.87
Average Length	59.12

Table 2. Comparison of score normalization methods for the residue-based secondary structure alignments.

4.2. Gap Score Optimization

To find the optimum values of the gap opening, and the gap extension penalties, *i.e.*, the parameters d, and e, we sampled a representative set of values and chose the ones that maximize the fold recognition

Normalization Method	$Q(\%)$
Raw Scores	10.47
Average Length	59.68

Table 3. The effect of average length normalization on the fold recognition accuracy of the SSEA method.

accuracy. For the gap opening cost, we chose integer values from -8 to -3. For the gap extension cost, we considered values from -1.6 to -0.2 with the increment set to 0.2. In addition to these values, we also found it useful to evaluate the performance for $d = -12$, $d = -10$, which are commonly used in sequence alignment methods.

We performed all-against-all alignments on the Ding and Dubchak set [10], where the secondary structure assignments are taken from the PDB. We have found that, the maximum fold recognition accuracy ($Q(\%) = 68.04$) is achieved at multiple values of $d\,(-6, -5, -4)$, and at a single e value ($e = -1.2$).

4.3. SSEA and Residue-Based Alignments

In this section, we compared the fold recognition accuracies of our method and the SSEA approach. Then, we evaluated the effect of incorporating amino acid similarity matrix (BLOSUM30) into the residue-based alignments of secondary structure. Tables 4, and 5 show the simulation results on the McGuffin, and Ding and Dubchak sets, respectively. Here, RBSS refers to the residue-based alignments of secondary structure using the secondary structure similarity matrix in Table 1. Secondary structures were predicted using the PSIPRED_v2.4. To serve as a reference point, we also computed the alignments using true secondary structure assignments obtained from the PDB. In simulations with the residue-based alignments, the gap opening and gap extension parameters are set to $d = -6$, and $e = -1.2$, respectively. Similar to the SSEA method, we discarded the leading and trailing coils and aligned the trimmed sequences.

Method	$Q(\%)$
SSEA	26.98
RBSS	29.62
BLOSUM30+RBSS ($\alpha = 0.5, \beta = 0.5$)	33.73
BLOSUM30+RBSS (True secondary structures)	35.31

Table 4. Fold recognition accuracy evaluated on the McGuffin set.

Method	$Q(\%)$
SSEA	60.47
RBSS	60.73
BLOSUM30+RBSS ($\alpha = 0.5, \beta = 0.5$)	70.68
BLOSUM30+RBSS (True secondary structures)	75.39

Table 5. Fold recognition accuracy evaluated on the Ding and Dubchak set.

From these results, the residue-based alignments of secondary structures performs comparable or better than the SSEA method. In addition, the incorporation of amino acid similarity scores such as BLOSUM30 brings significant improvements over the secondary structure alignments.

5. CONCLUSIONS

In this paper, we revisited the utilization of secondary structure alignments in fold recognition. We showed that the power of the state-of-the-art SSEA method comes from score normalization instead of the element alignment approach, and similar performance could be achieved using residue-based alignments when proper score normalization and gap scoring are applied. The residue-based nature of the proposed method also allows us to incorporate amino acid similarity matrices such as BLOSUM. The availability of aligning the amino acid and secondary structure sequences in a joint manner offers a better starting point for more elaborate techniques that employ profile-profile alignments and machine learning methods [3, 4].

6. REFERENCES

[1] T. Przytycka, R. Aurora, and G. D. Rose, "A protein taxonomy based on secondary structure," *Nat. Struct. Biol.*, vol. 6, pp. 672–682, 1999.

[2] L. J. Mcguffin, K. Bryson, and D. T. Jones, "What are the baselines for protein fold recognition?," *Bioinformatics*, vol. 17, pp. 63–72, 2001.

[3] Z. Zhang, S. Kochhar, and M. G. Grigorov, "Descriptor-based protein remote homology identification," *Protein Sci.*, vol. 14, pp. 431–444, 2005.

[4] J. E. Gewehr, N. V. Ohsen, and R. Zimmer, "Combining secondary structure element alignment and profile-profile alignment for fold recognition," in *German Conference on Bioinformatics*, 2004, pp. 141–148.

[5] A. Wallqvist, Y. Fukunushi, L. R. Murphy, A. Fadel, and R. M. Levy, "Iterative sequence/secondary structure search for protein homologs: comparison with amino acid sequence alignments and application to fold recognition in genome databases," *Bioinformatics*, vol. 16, no. 11, pp. 988–1002, 2000.

[6] S. F. Altschul and W. Gish, "Local alignment statistics," *Meth. Enzymol.*, vol. 266, pp. 460–480, 1996.

[7] R. Durbin, S. Eddy, A. Krogh, and G. Mitchison, *Biological Sequence Analysis: Probabilistic models of proteins and nucleic acids*, Cambridge University Press, 1981.

[8] L. J. Mcguffin and D. T. Jones, "Targeting novel folds for structural genomics," *Proteins: Structure, Function and Genetics*, vol. 48, pp. 44–52, 2002.

[9] C. A. Orengo, A. D. Michie, S. Jones, D. T. Jones, M. B. Swindells, and J. M. Thornton, "CATH: A hierarchical classification of protein domain structures," *Structure*, vol. 5, pp. 1093–1108, 1997.

[10] C. H. Q. Ding and I. Dubchak, "Multi-class protein fold recognition using support vector machines and neural networks," *Bioinformatics*, vol. 17, pp. 349–358, 2001.

[11] E. Bindewald, A. Cestaro, J. Hesser, M. Heiler, and S. C. E. Tosatto, "MANIFOLD: protein fold recognition based on secondary structure, sequence similarity and enzyme classification," *Protein Eng.*, vol. 16, pp. 785–789, 2003.