SWIFT: A New Threading Method Based on Short-Cut Phenomena in Protein Structure Comparison

Mingfu Shao, Chao Wang, Xiongying Yuan, Shuai Cheng Li, and Dongbo Bu*

Bioinformatics Laboratory, Institute of Computing Technology,
Chinese Academic of Sciences, Beijing, China
**Email: dbu@ict.ac.cn*

Motivation: Recent Critical Assessment of Techniques for Protein Structure Prediction (CASP) suggest that template-based threading methods are the most successful tools in predicting proteins' 3D structure. The core of threading methods is to generate accurate alignment of the query sequence and a template with known structure. We observed an interesting "short-cut" phenomena when analyzing CASP8 results: in the optimal alignments between some target sequences and templates, two spatially close cores of template are aligned to continuous positions in sequence; however, the intermediate regions of the two cores are totally skipped. Most of state-to-art threading methods perform badly in the "short-cut" cases since they usually require the matching of all cores.

Methods: Based on the observation, we first propose a formal definition to describe the "short-cut" phenomena; then design a score function to characterize it and employ dynamic programming technique to calculate the optimal alignment even if "short-cut" regions are allowed.

Results: We compare our method with state-of-art threading methods on commonly used benchmarks. On four representative CASP8 targets with "short-cut" phenomena, our method can generate high-quality alignment while RAPTOR and HHpred fail. On Prosup dataset, our method performs 6.3% better than RAPTOR in the measure of the alignment accuracy. In addition, our method show comparable fold recognition performance with other methods in the family level. We also evaluate the quality of the final predicted structures of our method. On 20 representative CASP8 targets, our method shows a comparable performance with Zhang-Server, and 0.02 better than RAPTOR, and 0.06 better than HHpred in the measure of average GDT_TS score.

Availability: We implemented our method into a C++ package SWIFT. SWIFT web-server can be accessed at http://www.bioinfo.org.cn/SWIFT/.

1. INTRODUCTION

Protein structure determination is critical for understanding protein functions, and also highly relevant with therapeutics and drugs design. Computational prediction methods for protein structure plays important roles due to the speed of experimental determination methods cannot catch up with that of generation of protein primary sequences by genome projects.

Protein structure prediction can be formulated as an optimization problem. That is, given a target sequence s and a candidate structure d, we use a score function $S(s, d)$ to measure the probability that the protein adopt the structural conformation. The prediction algorithms differ in their objectives (global optimum, local optimum, approximate optimum), search spaces, and search strategies they used to yield the optimal prediction.

The existing prediction methods can be divided into two categories: template-based modeling, and Ab initio methods. A template-based modeling method tries to find the optimal alignment between the query sequence and templates with known structures; in contrast, an Ab initio method predict protein structure without dependence of known structures. Recent advance suggest that threading is one of the most effective and accurate template-based modeling techniques.

Various threading methods have been proposed to calculate the optimal alignments under different scoring functions. For example, FASTA[1], BLAST[2], and PSI-BLAST[3] assume the independence of amino acids while HMMer[4] and HHpred[5] use Hidden Markov Model to introduce the transition information between adjacent residues into score function. Dynamic programming technique is naturally employed to obtain a global optimal solution as only local information are taken into consideration in the

The first two authors contribute equally to this work.
*Corresponding author.

score functions. However, local information alone is insufficient to detect protein homology in cases of low sequence identity. Recent CASPs results reveal the importance of global contact information [8]. However, if all pairwise interactions are added into score function, the optimization problem becomes NP-hard[9]. Thus, a trade-off between the efficiency of algorithms and adding more global information to score function should be considered. PROSPECT[6] simplifies the general case by placing a cutoff distance on the pairwise interactions and proposes a divide-and-conquer algorithm with time complexity of $O(Mn^{1.5C} + mn^{C+1})$ where M,n,m is the size of cores, template and sequence, respectively, and C is a small constant. RAPTOR[8] employs integer linear programming(ILP) model to characterize pairwise interaction and applies graph reduction technique to improve efficiency.

We observed an interesting "short-cut" phenomenon in CASP8 results: a long region of template (see A, part 3 of Fig. 1) is totally skipped in the optimal alignment to the sequence; however, the two end of this region are usually spatially-close, and are aligned closely in sequence (see B, part 3 of Fig. 1). Intuitively, if we cut off the short-cut regions, then the template and sequence can be aligned perfectly. Short-cut reflects a kind of global contact information, and the structure relationship between two proteins. The existing threading methods usually fail to generate high-quality alignment in the short-cut case. The potential reason is that global contacts are not perfectly described, and short-cut are usually treated as long gaps and received a heavy penalty.

In the study, we first give a mathematically strict definition of short-cut, then design a score function to characterize short-cut by introducing a special short-cut score item, and finally propose a dynamic programming method to find the optimal alignment when short-cut are allowed. We evaluate the performance of our method on widely-used benchmark datasets, including Prosup, SALIGN, Lindahl, and CASP8 targets.

2. Methods

2.1. Definition

We first describe preliminary notations and formal definitions as follows.

Definition 2.1. (Alignment) An alignment between a template $t = \{t_1, t_2, \cdots, t_n\}$ and a query sequence $s = \{s_1, s_2, \cdots, s_m\}$ is a mapping f satisfying:
(a) $f(t_i) \in s \cup \{-1\}$;
(b) if $f(t_i) = s_j$, $f(t_k) = s_l$, $i < k$, then $j < l$;
(c) if $f(t_i) = s_j$, $f(t_k) = s_l$, $i+1 < k$, and $f(t_z) = -1$ for all $z : i < z < k$, then $j + 1 = l$.

In restriction (a), $f(t_i) = -1$ means that t_i is a gap on template while $f(t_i) = s_j$ means t_i aligned to s_j; (b) ensures no intersection in the alignment; (c) ensures that *insert* and *delete* cannot appear simultaneously.

A template t is described as a series of *cores*, $t = \{c_1, c_2, \cdots, c_N\}$, where a core $c_i = \{t_{n_i}, t_{n_i+1}, \cdots, t_{n_i+|c_i|-1}\}$ refers to a continuous segment of template started at n_i. In practice, core is defined according to secondary structure information. A core might be aligned to the sequence ($f(c_i) = f(t_{n_i})$), or be totally neglected (denoted as $f(c_i) = -1$). The widely-accepted restriction on alignment is that if a core is aligned, it should appear in the alignment as a whole.

Now we give a mathematically strict definition of *short-cut*.

Definition 2.2. (Short-Cut) An alignment f has a short-cut at core c_i and c_j if
a) $i + 1 < j$;
b) there exists a contact between c_i and c_j;
c) $f(c_i) \neq -1$, $f(c_j) \neq -1$, $f(c_k) = -1$ for all $k : i < k < j$.

Intuitively, if two non-adjacent cores are aligned to sequence and all cores between them are skipped, then we say this alignment has a short-cut between the two cores. Note that this definition ensures the two cores are aligned to the adjacent positions in the sequence.

We define an index function to describe whether there are short-cut between two cores c_i and c_j in the alignment f.

$$sc_f(i,j) = \begin{cases} 1 \text{ if } f \text{ has short-cut at } c_i \text{ and } c_j \\ 0 \text{ otherwise} \end{cases}$$

2.2. Score Function

We introduce a specific score item to describe short-cut cores into the score function. Specifically, our score function $S(f)$ is the linear weighted sum of 3 items:

$$S(f) = \omega_m S_m(f) + \omega_g S_g(f) + \omega_{sc} S_{sc}(f),$$

where ω denote weights of score items. The three score items are described in details as follows:

matching item $S_m(f)$

The match item $S_m(f)$ evaluates matching status, which is the sum of matching status score of all residues:

$$S_m(f) = \sum_{1 \le i \le n} m(t_i, f(t_i)).$$

$$m(t_i, -1) = 0$$

and

$$m(t_i, s_j) = mu(t_i, s_j) + ss(t_i, s_j) + sa(t_i, s_j).$$

The mutation item $mu(t_i, s_j)$ represents the compatibility of substituting t_i to s_j.

$$mu(t_i, s_j) = \mathbf{t_i}^{\mathbf{T}} \mathbf{M} \mathbf{s_j},$$

where $\mathbf{t_i}$ and $\mathbf{s_j}$ is the profile vector at t_i and s_j obtained by running PSI-BLAST[3], $\mathbf{M}$ is substitute matrix, here we use BLOSUM-62[17].

The secondary structure item $ss(t_i, s_j)$ represents the secondary structure similarity of t_i and s_j.

$$ss(t_i, s_j) = \mathbf{ss}[s_j, SS(t_i)]$$

where $SS(t_i)$ is the secondary structure type(alpha, beta, or coil) of t_i, $\mathbf{ss}[s_j, SS(t_i)]$ is the probability that the secondary structure of s_j is $SS(t_i)$, which is predicted by PSIPRED[18].

The solvent accessibility item $sa(t_i, s_j)$ represents the compatibility of residue s_j to the environment t_i of the template.

$$sa(t_i, s_j) = \sum_{a=1}^{20} \mathbf{s_j}[a] \cdot \mathbf{sa}[a, acc(t_i)].$$

$acc(t_i)$ is the solvent accessibility value obtained from DSSP[21]. $\mathbf{sa}[a, acc(t_i)]$ is the probability that amino acid a appears in a spatial position whose solvent accessibility value is $acc(t_i)$.

gap penalty item $S_g(f)$

The gap penalty item $S_g(f)$ represents the penalty for the gap on the alignment.

$$S_g(f) = \omega_{g1} N_{g1} + \omega_{g2}(N_{g2} - N_{g1})$$

where N_{g1} is the number of gaps in f and N_{g2} is the number of gap residues.

short-cut item $S_{sc}(f)$

The short-cut item $S_{sc}(f)$ represents the short-cut award.

$$S_{sc}(f) = \sum_{1 \le i \le N} \sum_{1 \le j \le N} sc_f(i,j) \cdot sc(c_i, c_j),$$

and

$$sc(c_i, c_j) = -[\omega_{g1} + \omega_{g2}(n_j - n_i - |c_i|)] + \sum_{i<k<j} \sum_{a \in c_k} \mathbf{p}[a].$$

Here, $\omega_{g1} + \omega_{g2}(n_j - n_i + |c_i|)$ is the gap penalty between c_i and c_j calculated in $S_g(f)$. $\mathbf{p}[a]$ is the average match score that if residue a is aligned to certain position.

2.3. Algorithm

We design a *dynamic programming* algorithm to find the optimal alignment, i.e., solving the optimization problem $\min_f S(f)$. It should be noticed that dynamic programming technique still works even if the score function contains contact information between cores. Briefly, the reason is that short-cut never intersect: for any alignment f, if $i < k < j < l$, it is impossible that $sc_f(i,j) = 1$ and $sc_f(k,l) = 1$ simultaneously since $sc_f(i,j) = 1$ implies that $f(c_k) = -1$, i.e., there is no short-cut at c_k. This property of S_{sc} guarantees the independence of the sub-problem, which is necessary and sufficient condition to apply dynamic programming technique.

Three tables are calculated in the dynamic programming algorithm, namely $n \times m$ table M, D and I. Here, $M_{i,j}$ denotes the minimum of the following sub-problem

$$\begin{aligned} & \min_{f \in \mathcal{F}} \quad S(f) \\ s.t. \quad & f(t_i) = s_j \\ & t = \{t_1, t_2, \cdots, t_i\} \\ & s = \{s_1, s_2, \cdots, s_j\} \end{aligned} \tag{M}$$

$D_{i,j}$ and $I_{i,j}$ denote the optimal solution values where the restriction (M) is replaced with (D) and (I), respectively.

$$f(t_k) = s_j \quad \exists\, k < i \tag{D}$$

$$f(t_i) = s_l \quad \exists\, l < j \tag{I}$$

The recurrence can be described as follows. For a residue in the core c_i, let $h_i = n_i + |c_i| - 1$,

$$\begin{aligned} I_{h_i,j} &= \min\{I_{h_i,j-1} + \omega_{g2}, M_{h_i,j-1} + \omega_{g1} + \omega_{g2}\} \\ D_{h_i,j} &= \min\{D_{n_i-1,j} + \omega_{g2}|c_i|, M_{n_i-1,j} + \omega_{g1} + \omega_{g2}|c_i|\} \\ M_{h_i,j} &= \min\{M_{n_i-1,j-|c_i|}, D_{n_i-1,j-|c_i|}, I_{n_i-1,j-|c_i|}, \\ &\qquad SC(c_i,j)\} + \sum_{k=0}^{|c_i|-1} m(n_i+k, j+k) \end{aligned}$$

where

$$SC(c_i,j) = \min_{k<i:c(k,i)=1} M_{h_k,j-|c_i|} + \omega_{g1} + \omega_{g2}(n_i - h_k - 1)$$

For a residue i in non-core region,

$$\begin{aligned} I_{i,j} &= \min\{I_{i,j-1} + \omega_{g2}, M_{i,j-1} + \omega_{g1} + \omega_{g2}\} \\ D_{i,j} &= \min\{D_{i-1,j} + \omega_{g2}, M_{i-1,j} + \omega_{g1} + \omega_{g2}\} \\ M_{i,j} &= \min\{M_{i-1,j-1}, D_{i-1,j-1}, I_{i-1,j-1}\} + m(i,j) \end{aligned}$$

Our objective is to calculate

$$\min\{M_{n,m}, D_{n,m}, I_{n,m}\}.$$

The time complexity of this algorithm is $O(n^2 mN)$.

3. Results

3.1. Parameter Training

We select 1600 protein pairs from PDB90 for training while 967 pairs for validation. Each pair shares high structural similarity (TMscore > 0.75) and low sequence identity($< 30\%$) with length between 80 and 500. The proteins sharing sequence identity over 30% with the proteins in the benchmarks (Prosup, SALIGN, Lindahl and CASP8) were also removed.

The goal of the training process is to maximize the average alignment accuracy on the training set. For an alignment, accuracy is defined as the ratio of the number of correct matching residues in an alignment over that in a reference alignment. In our experiments, reference alignment is generated by TMalign[12]. Linear search technique is employed to find the optimal parameters. Using the acquired parameters, our method achieves an average alignment accuracy of 0.781 in the validation set.

3.2. Performance of SWIFT on Short-Cut Examples in CASP8

We perform case studies on 4 representative CASP8 targets with obvious short-cut phenomena. For each target, the alignment with a structure-similar template are shown in Fig. 1, where template's structure is drawn in red, and the native structure of the target in green. The cores that are short-cut are labelled by A, and the two end-point cores of the short-cut region are marked by B.

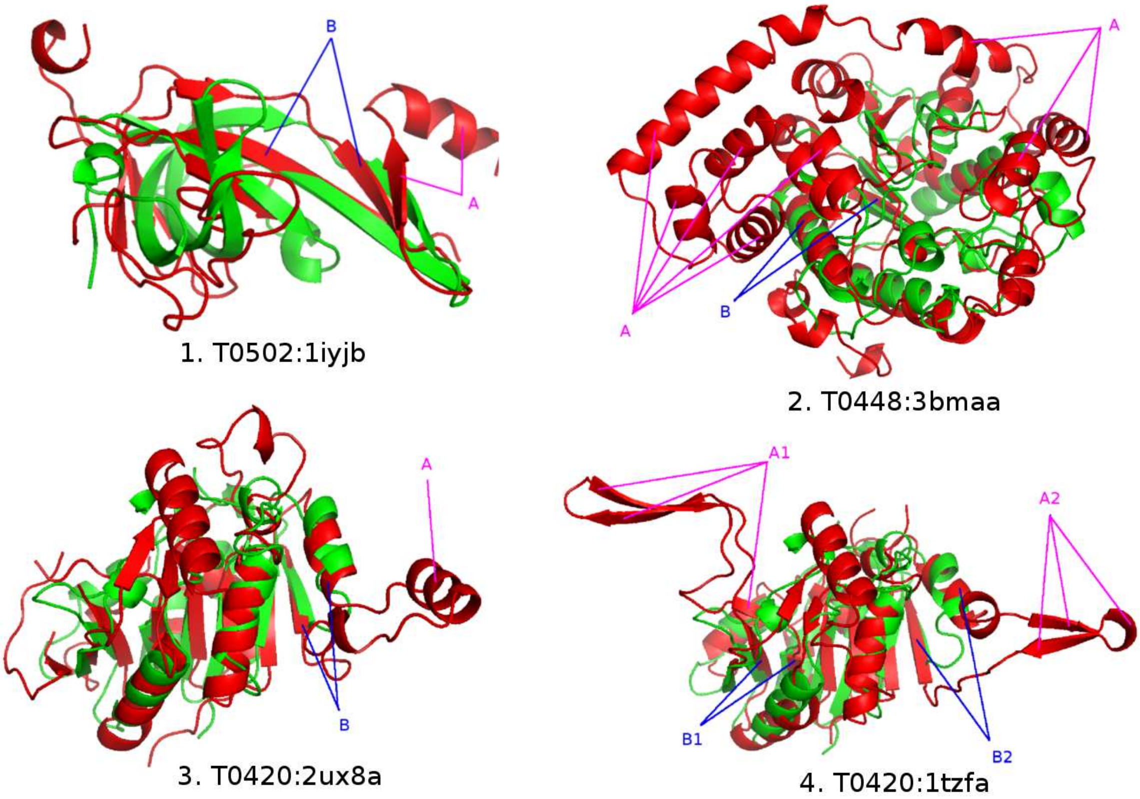
Target T0502 has a beta barrel structure (See Panel 1 of Fig. 1). In this target, short-cut forms by replacing the hairpin of two long beta strands with a alpha helix and a beta strand. This is the most popular forms of short-cut: two beta strands are tightly linked by hydrogen bond, and short-cut occurs in the connection regions. Another example of this form is the alignment between T0420 and 1tzfa (see A2, B2, Panel 4 of Fig. 1), where short-cut cores are three beta strands. From the point view of topology, alpha/beta proteins consist of a beta protein surrounded by several alpha helices. The short-cut cores change the class of the proteins from beta to alpha/beta.

Short-cut can also take place between alpha helix and beta strand. An example is the alignment between T0448 and 3bmaa (See Panel 2 of Fig. 1): eight outer alpha helices are short-cut cores. Similar examples are the alignment between T0420 and 2ux8a(A, B, Panel 3 of Fig. 1), and the alignment between T0420 and 1tzfa (A2, B2, Panel 4 of Fig. 1), where short-cut cores are one alpha helix and 3 beta strands, respectively. In this case, the short-cut cores bring no change to the basic topology; however, active site might be significantly changed.

It should be noticed that even for a same target, different short-cut cores might be observed if aligned with different templates. Take T0420 as a concrete example, the alignment against 1tzfa has two short-cut regions while the alignment against 2ux8a has only one short-cut regions (See Panels 3 and 4 of Fig. 1).

On these 4 pairs, SWIFT can detect the struc-

Fig. 1. Four short-cut examples



tural similarity by generating high-quality alignments. Table 1 lists the comparison of HHpred, RAPTOR and SWIFT in the sense of alignment accuracy. HHpred performs badly on T0502 and T0448. This is probably because HHpred utilizes the relationship of adjacent residues only, and thus cannot characterize remote contact. RAPTOR performs badly on all the four cases. One possible reason is that RAPTOR requires all cores to be aligned and no core can be skipped. Thus, the short-cut regions are usually aligned to wrong positions, or are heavily penalized if treated as a long gap.

Table 1. Comparison of SWIFT against RAPTOR, HHpred on 3 Representative CASP8 Targets with Short-cut.

				Alignment Accuracy %		
Target	Template	TMscore	Identity	HHpred	RAPTOR	SWIFT
T0502	1iyjb	0.77	0.10	0	0.29	0.62
T0448	3bmaa	0.63	0.12	0.35	0	0.61
T0420	2ux8a	0.77	0.12	0.68	0.20	0.76
T0420	1tzfa	0.79	0.20	0.61	0.31	0.74

3.3. Alignment Accuracy Comparison on Prosup and SALIGN Benchmark

We evaluate SWIFT by comparing with HHpred and RAPTOR in the sense of alignment accuracy on two commonly used benchmarks, namely, Prosup[11] and SALIGN[10]. The Prosup benchmark dataset contains 127 protein pairs, and SALIGN benchmark contains 200 protein pairs. Two proteins in a pair shares low sequence identity (13.7% on average) and high structural similarity (TMscore = 0.595 on average). Reference alignments are generated through TMalign[12].

Table 2 demonstrates that the performance of SWIFT is about 6.3% better than RAPTOR and 3.3% worse than HHpred on Prosup benchmark, and about 4.8% better than RAPTOR and 5.8% worse than HHpred on SALIGN benchmark. This result further indicates that some key remote contacts are sufficient to characterize the topology. On the other hand, HHpred shows its advantage of employing adjacent residues' interaction.

Table 2. Alignment Accuracy Comparison on Prosup and SALIGN benchmark.

Prosup		SALIGN	
Methods	Align.Acc(%)	Methods	Align.Acc(%)
HHpred[1]	52.1	HHpred[1]	58.3
RAPTOR[1]	42.5	RAPTOR[1]	47.7
SWIFT[2]	48.8	SWIFT[2]	52.5

3.4. Fold Recognition Comparison on Lindahl Benchmark

We evaluate the fold recognition rate of our method on Lindahl benchmark dataset containing 976 proteins, with 555, 434 and 321 protein pairs in the same fold, superfamily, and family, respectively. The top1 (top5) columns show the percentage of proteins that are ranked correctly as top1 (top5).

Table 3. Fold Recognition Rate(%) on Lindahl Benchmark

	family		superfamily		fold	
Methods	Top1	Top5	Top1	Top5	Top1	Top5
PSIBLAST[1]	71.2	72.3	27.4	27.9	4.0	4.7
HMMer[2]	67.7	73.5	20.7	31.3	4.4	14.6
SAM-98[2]	70.1	75.4	28.3	38.9	3.4	18.7
HHpred[1]	82.9	87.1	58.8	70.0	25.2	39.4
RAPTOR[3]	86.6	89.3	56.3	69.0	38.2	58.7
SWIFT	79.7	86.9	50.6	59.4	10.1	29.8

[1] from [14] [2] from [15] [3] from [13]

Tab. 3 suggests that SWIFT outperforms PSI-BLAST, HMMer and SAM98 on all three levels, while performs worse compared with HHpred and RAPTOR on superfamily and fold levels. One possible reason is that for these proteins, say {d1prcl1, d1occl1, d1occk1, d1occj1, d1occi1, d1occg1, d1occd1, d1occc1, d1occb2}, secondary structure are inaccurately predicted, and SWIFT relies heavily on secondary structure prediction. How to tune parameters to overcome this shortcoming remains as one of the future works.

3.5. Performance of Structure Prediction on CASP8 targets

To predict the 3D structure for a given target sequence, SWIFT first aligns the sequence to all templates, and then selects the best alignments according to the final score. Taking alignment and structure of template as input, Modeller[20] is executed to yield the 3D structure as prediction. We select 20 targets from CASP8, of which 7 targets are classified as TBM-HA(Template Based Modeling, High Accuracy), 13 are TBM(Template Based Modeling). Table 4 illustrates that SWIFT is comparable to Zhang-Server, and better than both RAPTOR and HHpred.

Table 4. Structure Prediction Performance(GDT_TS) on 20 targets of CASP8

Target	Class	Zhang[1]	RAPTOR[1]	HHpred[1]	SWIFT
T0432	TBM	0.62	0.58	0.58	0.6
T0471	TBM	0.29	0.29	0.29	0.3
T0390	TBM	0.65	0.64	0.55	0.63
T0435	TBM	0.65	0.68	0.62	0.65
T0485	TBM	0.75	0.72	0.73	0.74
T0464	TBM	0.67	0.7	0.62	0.66
T0409	TBM	0.65	0.61	0.57	0.64
T0480	TBM	0.64	0.54	0.65	0.64
T0398	TBM	0.77	0.71	0.36	0.74
T0493	TBM	0.53	0.38	0.44	0.59
T0419	TBM	0.37	0.27	0.29	0.31
T0400	TBM	0.42	0.39	0.33	0.39
T0468	TBM	0.69	0.68	0.66	0.68
T0495	TBM-HA	0.6	0.6	0.48	0.58
T0489	TBM-HA	0.72	0.72	0.64	0.71
T0423	TBM-HA	0.96	0.96	0.94	0.96
T0462	TBM-HA	0.86	0.87	0.81	0.87
T0474	TBM-HA	0.96	0.96	0.96	0.97
T0392	TBM-HA	0.83	0.84	0.84	0.83
T0494	TBM-HA	0.81	0.81	0.81	0.83
Average		0.67	0.65	0.61	0.67

[1] models are download from CASP8 website, GDT_TS score is obtained by running TMscore.

4. Conclusion and Discussion

This paper reports a new threading method based on the observation of short-cut. Using a linear score function plus a short-cut score item, our method's performance can compare to the leading threading algorithms. Besides, on short-cut examples, our method performs better than both HHpred and RAPTOR.

Short-cut can be used to classify proteins. Our method can detect proteins that have different secondary structure type, which might be classified as different classes by SCOP[19]. This criterion would be more reasonable in the point view of topology.

An extensive test of our method on more datasets is one of the future works.

References

1. Pearson WR, Lipman DJ. Improved tools for biological sequence comparison. Proc Natl Acad Sci USA. 1988 Apr;85(8):2444-8.
2. Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ. Basic local alignment search tool. J Mol Biol. 1990 Oct 5;215(3):403-10.
3. Altschul SF, Madden TL, Schaffer AA, Zhang J, Zhang Z, Miller W, Lipman DJ. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Res. 1997 Sep 1;25(17):3389-402.
4. Durbin, Richard; Sean R. Eddy, Anders Krogh, Graeme Mitchison (1998). Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids. Cambridge University Press. ISBN 0521629713.
5. Soding J. Protein homology detection by HMM-HMM comparison. Bioinformatics. 2005 Apr 1;21(7):951-60. Epub 2004 Nov 5.
6. Xu Y, Xu D, Uberbacher EC. An efficient computational method for globally optimal threading. J Comput Biol. 1998 Fall;5(3):597-614.
7. Xu Y, Xu D. Protein threading using PROSPECT: design and evaluation. Proteins. 2000 Aug 15;40(3):343-54.
8. Xu J, Li M, Kim D, Xu Y. RAPTOR: optimal protein threading by linear programming. J Bioinform Comput Biol. 2003 Apr;1(1):95-117.
9. Richard H. Lathrop. The protein threading problem with sequence amino acid interaction preferences is NP-complete. Protein Engineering vol. 7 no. 9 pp. 1059-1068, 1994.
10. Marti-Renom MA, Madhusudhan MS, Sali A. Alignment of protein sequences by their profiles. Protein Sci. 2004 Apr;13(4):1071-87.
11. Lackner P, Koppensteiner WA, Sippl MJ, Domingues FS. ProSup: a refined tool for protein structure alignment. Protein Eng. 2000 Nov;13(11):745-52.
12. Zhang Y, Skolnick J. TM-align: a protein structure alignment algorithm based on the TM-score. Nucleic Acids Res. 2005 Apr 22;33(7):2302-9. Print 2005.
13. Xu J. Fold recognition by predicted alignment accuracy. IEEE/ACM Trans Comput Biol Bioinform. 2005 Apr-Jun;2(2):157-65.
14. Zhang W, Liu S, Zhou Y. SP5: improving protein fold recognition by using torsion angle profiles and profile-based gap penalty model. PLoS One. 2008 Jun 4;3(6):e2325.
15. Cheng J, Baldi P. A machine learning information retrieval approach to protein fold recognition. Bioinformatics. 2006 Jun 15;22(12):1456-63. Epub 2006 Mar 17.
16. Xu J, Li M, Kim D, Xu Y. RAPTOR: optimal protein threading by linear programming. J Bioinform Comput Biol. 2003 Apr;1(1):95-117.
17. S Henikoff and J G Henikoff. Amino acid substitution matrices from protein blocks. Proc Natl Acad Sci U S A. 1992 November 15; 89(22): 1091510919.
18. Jones DT. Protein secondary structure prediction based on position-specific scoring matrices. J Mol Biol. 1999 Sep 17;292(2):195-202.
19. Hubbard TJ, Ailey B, Brenner SE, Murzin AG, Chothia C. SCOP, Structural Classification of Proteins database: applications to evaluation of the effectiveness of sequence alignment methods and statistics of protein structural data. Acta Crystallogr D Biol Crystallogr. 1998 Nov 1;54(Pt 6 Pt 1):1147-54.
20. N. Eswar, M. A. Marti-Renom, B. Webb, M. S. Madhusudhan, D. Eramian, M. Shen, U. Pieper, A. Sali. Comparative Protein Structure Modeling With MODELLER. Current Protocols in Bioinformatics, John Wiley & Sons, Inc., Supplement 15, 5.6.1-5.6.30, 2006.
21. Kabsch W, Sander C. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. Biopolymers. 1983 Dec;22(12):2577-637.