PROTEINS
STRUCTURE ■ FUNCTION ■ BIOINFORMATICS

Improving taxonomy-based protein fold recognition by using global and local features

Jian-Yi Yang* and Xin Chen

Division of Mathematical Sciences, School of Physical and Mathematical Sciences, Nanyang Technological University, 21 Nanyang Link, Singapore 637371

ABSTRACT

Fold recognition from amino acid sequences plays an important role in identifying protein structures and functions. The taxonomy-based method, which classifies a query protein into one of the known folds, has been shown very promising for protein fold recognition. However, extracting a set of highly discriminative features from amino acid sequences remains a challenging problem. To address this problem, we developed a new taxonomy-based protein fold recognition method called TAXFOLD. It extensively exploits the sequence evolution information from PSI-BLAST profiles and the secondary structure information from PSIPRED profiles. A comprehensive set of 137 features is constructed, which allows for the depiction of both global and local characteristics of PSI-BLAST and PSIPRED profiles. We tested TAXFOLD on four datasets and compared it with several major existing taxonomic methods for fold recognition. Its recognition accuracies range from 79.6 to 90% for 27, 95, and 194 folds, achieving an average 6.9% improvement over the best available taxonomic method. Further test on the Lindahl benchmark dataset shows that TAXFOLD is comparable with the best conventional template-based threading method at the SCOP fold level. These experimental results demonstrate that the proposed set of features is highly beneficial to protein fold recognition.

Proteins 2011; 00:000–000.
© 2011 Wiley-Liss, Inc.

Key words: fold recognition; evolutionary information; secondary structure; feature extraction; support vector machine.

INTRODUCTION

Protein fold recognition from amino acid sequences is one of the fundamental problems in structural bioinformatics, as fold information could facilitate the identification of a protein's tertiary structure and function. In the last two decades, a substantial amount of research effort has been devoted to developing efficient and effective computational methods for protein fold recognition. These computational methods can be broadly classified into two categories, that is, template-based[1–6] and taxonomy-based.[7–13] In recent years, the taxonomy-based method has attracted great attention due to its encouraging performance.

The taxonomy-based method for protein fold recognition was first proposed in 1995 by Dubchak *et al.*[7,8] It follows a popular belief that there are only a limited number of different protein folds in nature. Consequently, the problem of protein fold recognition can be viewed as a classification problem so that it can be tackled by using the methods in machine learning. Most implementations of the taxonomy-based method, if not all, have adopted the SCOP protein structural classification architecture,[14] with which a query protein is classified into one of the known folds. To implement a classification task, two major procedures are generally required—feature extraction and a machine learning classifier. Below we briefly review the existing taxonomy-based methods from these two aspects.

Feature extraction refers to a procedure by which we extract features from a query amino acid sequence so as to represent the underlying protein as a fixed-length numerical vector. Dubchak et al.[7,8] first proposed a way to extract features using global description of amino acid sequence. Since then, many new features have been developed to improve the recognition accuracy, such as those based on pseudo-amino acid composition,[10,15] structural properties of amino acid residues and amino acid residue pairs,[16] autocross-covariance transformation,[13] and hidden Markov model structural alphabet.[17] Besides the features extracted directly from amino acid sequences, some features are constructed through exploiting information such as predicted secondary structure,[12] sequence evolution,[12,13] functional domain,[18] and predicted solvent accessibility.[16]

Additional Supporting Information may be found in the online version of the article.
Grant sponsor: Singapore NRF; Grant number: NRF2007IDM-IDM002-010; Grant sponsor: MOE AcRF Tier 1; Grant number: RG78/08.
*Correspondence to: Jian-Yi Yang, Division of Mathematical Sciences, School of Physical and Mathematical Sciences, Nanyang Technological University, 21 Nanyang Link, Singapore 637371.
E-mail: yang0241@ntu.edu.sg
Received 12 December 2010; Revised 5 February 2011; Accepted 3 March 2011
Published online 21 March 2011 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/prot.23025

These features were reported capable of achieving satisfactory fold recognition accuracies, especially when they are utilized in combination.[19,20]

A machine learning classifier is basically an algorithmic procedure that assigns each fixed-numerical vector a predefined class label. For protein fold recognition, a number of classifiers have been applied, such as neural networks (NNs),[7–9] Support Vector Machines (SVMs),[9,16,13,20] probabilistic multiclass multi-kernel classifier,[19] and various ensemble classifiers.,[10–12,18,21]

Note that feature extraction is a key step toward the success of classification. For protein fold recognition, however, it is not yet clear what features are the most discriminative. In this study, we look into this challenging problem and explore various ways to extract features from PSI-BLAST profiles[22] and also from PSIPRED profiles.[23] These two profiles are believed to contain rich information about protein sequence evolution and secondary structure, respectively. To depict their global characteristics and local characteristics, we extract 82 features from PSI-BLAST profiles and 55 features from PSIPRED profiles, resulting in a comprehensive set of 137 features. With these features, we have developed a new taxonomy-based fold recognition method called TAXFOLD, which additionally uses an SVM-based classifier. As our experimental tests on four datasets demonstrate, TAXFOLD can achieve an average 6.9% accuracy improvement over the best available method, indicating that the proposed set of features, has the enhanced power to discriminate between different folds.

MATERIALS AND METHODS

Datasets

We used five datasets in this study to evaluate our proposed method: DD, RDD, EDD, F95, F194, and Lindahl. The first dataset was originally created by Ding and Dubchak,[9] and later revised by Shen and Chou.[10] It comprises 311 protein domain sequences for the training purpose and 383 sequences for testing, each of which is classified into one of 27 folds. It was reported that none of the testing sequences shared more than 35% sequence identity to any of the training sequences.[9] However, Chen and Kurgan[12] later found seven duplicate pairs between the training and testing sequences. Our close inspection further revealed that 11 training sequences and 76 testing sequences were already updated in the latest release of SCOP database (release 1.75, June 2009, http://astral.berkeley.edu/); see details in Supporting Information A. Moreover, there exists a domain (1BUCA1), which is no longer classified into any of the above 27 folds. On the basis of these observations, we updated the dataset accordingly, so that the latest domain sequences are experimented, and the domain 1BUCA1 is excluded from further consideration. The resulting dataset is thus called the revised DD dataset (RDD), to distinguish from its original dataset (called DD).

The other three datasets are constructed with the same procedure as done in Ref. 13, except that the latest release of SCOP (release 1.75, June 2009) is used. The domain sequences that have less than 40% pairwise identity are first extracted from the Astral SCOP 1.75 release (http://astral.berkeley.edu/) and then those shorter than 31 residues are further removed, resulting in a total of 10,493 domain sequences. Of these sequences, 3397 are classified into one of the above-mentioned 27 folds, so we use them to construct an extended DD dataset (EDD). To cover more folds, we construct another two datasets by selecting the folds that contain at least 26 sequences and at least 11 sequences, respectively. Consequently, one dataset comprises 6364 sequences from 95 folds and the other comprises 8026 sequences from 194 folds. We call them F95 and F194, respectively.

The above four datasets, RDD, EDD, F95, and F194, are available at http://www1.spms.ntu.edu.sg/~chenxin/TAXFOLD/. The last dataset is Lindahl benchmark dataset,[24] which can be downloaded from http://www.bioinfo.se/protein-id/. This dataset contains 976 proteins in three SCOP levels: family, superfamily, and fold. The pair-wise sequence identity in this dataset is smaller than 40%. This dataset is used to compare our method with template-based threading methods.

Note that, in this study, we make fold recognition for domain sequences rather than the whole protein sequences. For multiple-domain proteins, we will make fold recognition for each individual domain sequence separately.

Feature extraction methods

In this study, we extract features from both profiles of PSI-BLAST[22] and PSIPRED,[23] where the rich sequence evolution information and secondary structure information are present. The features are carefully developed so that they can depict both global and local characteristics of profiles. Global characteristics refer to the patterns that the whole profile is held by, whereas local characteristics refer to the patterns particular to some profile fragment. Accordingly, the features thus extracted are called global and local features, respectively. We further divide the global features into two categories. If a global feature does not depend on sequence order, we call it a globalA feature (e.g., the first-order entropy of sequences); otherwise, a globalB feature. The whole procedure of feature extraction is depicted in Figures 1–3.

PSI-BLAST profile-based features

The PSI-BLAST profile is represented as a so-called position-specific score matrix (PSSM), which is obtained through aligning a query amino acid sequence to the NCBI's nonredundant (NR) database by using PSI-

BLAST[22] with three iterations and a cutoff *E*-value of 0.001. The PSSM is a log-odds matrix of size $L \times 20$, where L is the length of the query sequence and 20 is due to the 20 amino acids. Its elements are the log-odds ratios between the observed base frequencies and the background base frequencies, followed by scaling by 10 and rounding down to the nearest integer. Therefore, the positive (respectively, negative) element values mean that the corresponding amino acids appear more (resp., less) often than expected from the background.

It should be noted that PSSMs have been considered in many taxonomy-based fold recognition methods. For example, Chen and Kurgan[12] extracted from PSSM a 20-D profile-based composition vector (PCV) in a way by which the negative elements of PSSM are first replaced by zero, and then each column is averaged. Although replacing negative elements by zero can ensure that the elements of PCV are all non-negative, it would definitely lose valuable evolutionary information that might be beneficial to fold recognition. To avoid this disadvantage, we propose an alternative way to extract features from PSSMs, as detailed below.

Our feature extraction method starts by transforming each element s_{ij} of the PSSM into s'_{ij} using

$$s'_{ij} = 2^{0.1 \times s_{ij}}. \tag{1}$$

Note that this transformation is the inverse of the algorithmic operation that PSI-BLAST used to compute the PSSM log-odds ratios.[22] The resulting value s'_{ij} thus represents a ratio between the observed base frequency and the corresponding background-based frequency and is guaranteed to be non-negative even when s_{ij} is negative. We further apply the normalization to the values s'_{ij} such that each row would sum to one. Let f_{ij} denote the normalized value of s'_{ij}. All the values f_{ij} form a matrix, which we called the frequency matrix (FM).

To extract globalA features (i.e., sequence order-free features), a so-called consensus sequence (CS)[25] is first constructed from the FM as follows:

$$\mu(i) = \arg\max\{f_{ij} : 1 \le j \le 20\}, \quad 1 \le i \le L \tag{2}$$

where "arg" stands for the argument of the maximum. The ith base CS (i) of the consensus sequence is then set to be the $\mu(i)$-th amino acid in the amino acid alphabet. It can be seen that a consensus sequence retains the most valuable evolutionary information from the PSSM. Then, we compute

$$\mathrm{AACCS}(j) = \frac{n(j)}{L}, \quad 1 \le j \le 20 \tag{3}$$

where $n(j)$ is the number of the amino acid j occurring in the CS. It will give 20 features corresponding to the amino acid composition of the CS. Moreover, we also include the entropy into our feature set, that is,

$$\mathrm{ECS} = -\sum_{j=1}^{20} \mathrm{AACCS}(j) \ln \mathrm{AACCS}(j) \tag{4}$$

where the base of the logarithm is Euler's number e.

Note that the above features can be computed with the original protein sequences as well. Our experimental results in Analysis of Future Contribution Section show that the features extracted from CSs are more fold-specific discriminative than those from the original protein sequences. Another entropy-based feature is directly computed from FM to reflect the global characteristic of the PSSM.

$$\mathrm{EFM} = -\frac{1}{L}\sum_{i=1}^{L}\sum_{j=1}^{20} f_{ij} \ln f_{ij} \tag{5}$$

To extract local features, we first divide FM into λ nonoverlapping fragments of equal length (see Fig. 1).[†] Then, for each fragment s , by applying a similar procedure in Ref. 18, the following 20 features are computed:

$$\mathrm{AOF}_s(j) = \frac{1}{\mathrm{len}_s}\sum_{i} f_{ij}, \quad 1 \le s \le \lambda,\ 1 \le j \le 20 \tag{6}$$

where the summation is done over the fragment s and len_s is the length of the fragment s. $\mathrm{AOF}_s(j)$ represents the average occurrence frequency of the amino acid j in the fragment s during the evolution process. The features $\mathrm{AOF}_1(j)$ and $\mathrm{AOF}_\lambda(j)$ in the first and last fragments may reflect the sequence characteristics at the N-terminus and C-terminus, respectively. How to determine the optimal value of λ will be discussed in Optimal values of λ and l_{max} Section.

In summary, for each query domain sequence, a total of $(22 + 20\times\lambda)$ features are extracted from its PSI-BLAST profile, among which 22 are globalA features and $20\times\lambda$ are local features.

PSIPRED profile-based features

The PSIPRED profile of a query protein contains the secondary structure information predicted with PSIPRED.[23] It comprises a state sequence and three probability sequences. The state sequence is a sequence of three possible symbols H, E, and C, representing states of helix, strand, and coil, respectively. The three probability sequences, each for one state, are the sequences of probability values with which the states occur along the query amino acid sequence. For example, see Supporting

[†]The last fragment may be longer because L is not always divisible by λ.

Information B. We make use of all these sequences to extract features.

First, we extract several global features from a state sequence by using a method that we have presented in a previous work on protein structural classes.[26] This method is indeed rooted in the work of Refs. 27 and 28. For readers' convenience, we briefly introduce these features below.

The first two features describe the helix and strand contents of the state sequence:

$$p(\mathrm{H}) = \frac{n(\mathrm{H})}{L}, \quad p(\mathrm{E}) = \frac{n(\mathrm{E})}{L} \tag{7}$$

where $n(H)$ and $n(E)$ are the numbers of the helix and strand states, respectively. In addition, the entropy of the state sequence and the length L of the query domain sequence are also included into our feature set because our experiments show that they can improve prediction accuracies. Note that these are four globalA features.

To extract globalB features, we first reduce a state sequence into a segment sequence that is composed of helix segments and strand segments (denoted by α and β, respectively). Here, a helix segment refers to a continuous segment of all H symbols in the state sequence, and a similar definition is also applied to a strand or coil segment. As at least three and two residues are generally required to form an α helix structure and an β strand structure, respectively, we will ignore those helix and strand segments that do not meet this size requirement. Moreover, to focus on spatial arrangement of α helix and β strand segments, all the coil segments are further ignored.

Let p_t denote the probability of transitions between α and β segments in a segment sequence, which is essentially the relative frequency of the substring αβ or βα occurring in the segment sequence. Let $p_{c\alpha}$ (respectively, $p_{c\beta}$) denote the probability of segments of two consecutive α (respectively, β). It hence follows that $p_t + p_{c\alpha} + p_{c\beta} = 1$; therefore, any probability can be deduced from the other two. So, we choose to include p_t and $p_{c\alpha}$ into our feature set, together with the probability $p(\beta)$ of strand segments occurring in the segment sequence. Note that these are three globalA features.

In addition, we define 18 globalB features as follows. For each state sequence, two time series are generated based on chaos game representation (CGR). These two time series are then analyzed by a nonlinear technique called recurrence quantification analysis (RQA), from which a total of 18 features are obtained. The procedure of extracting features from a state sequence based on CGR and RQA is illustrated in Figure 2. For the details about CGR and RQA and the features, please refer to Supporting Information B and/or the Refs. 26 and 29. Note that CGR and RQA are used to analyze the secondary structure here and it can be applied to amino acid sequences as well.[29] However, it was shown that better results could be frto b7bal sequf7wdy7.

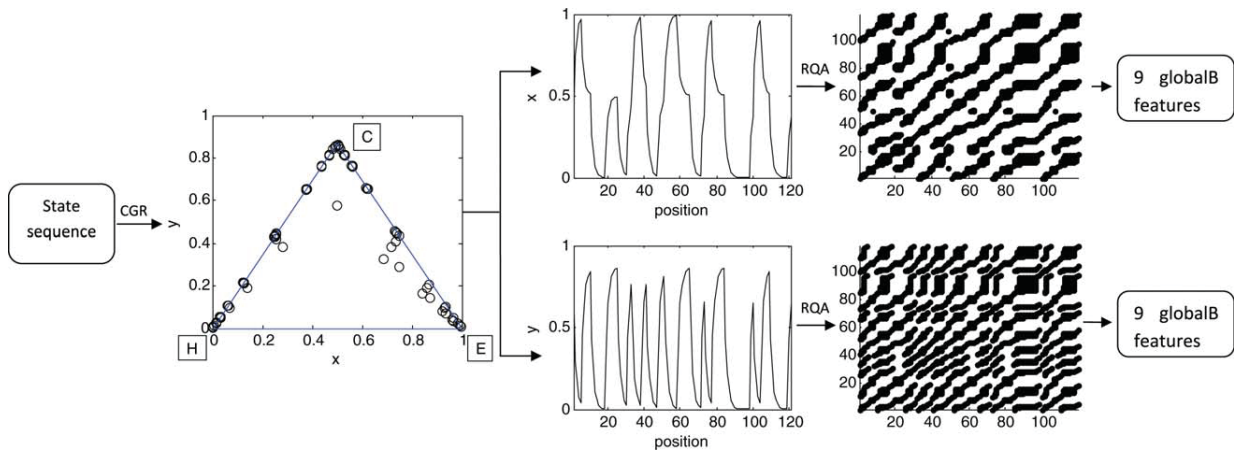

Figure 2

Illustration of the procedure to extract features from the state sequences of PSIPRED profiles by applying CGR and RQA.

transformation on the probability sequences. AC is the covariance of the sequence against a time-shifted version of itself. That is, for the sequence $t = (t_1,t_2,\cdots,t_L)$ of length L, the AC transformation will return

$$\mathrm{AC}_{l,t} = \frac{1}{L-l}\sum_{i=1}^{L-l}(t_i - \bar{t})(t_{i+l} - \bar{t}), \quad l = 1, 2, \cdots, l_{\max} \tag{9}$$

where $\bar{t}$ is the average over all t_i, l is the lag (distance) between two positions along the sequence, and l_{max} is the maximum of l. All these $\mathrm{AC}_{l,t}$ will be used as features. For the three-probability sequences, we thus obtain $3 \times l_{max}$ globalB features. The selection of the optimal value of l_{max} will be discussed in Optimal values of λ and l_{max} Section.

Note that AC transformation was previously applied to PSSM for fold recognition,[13] resulting a huge number of features, which makes it difficult to train SVM-based classifiers. We have tried to incorporate these features into our feature set, but the results were not improved, and thus, we do not use these features in this study.

In summary, a total of $(28 + 3 \times l_{max})$ features have been extracted from a PSIPRED profile. Among these, 4 globalA features and 21 globalB features are extracted from the state sequence, while 3 globalA features and $3 \times l_{max}$ globalB features are from the probability sequences. Combining these features with those extracted from PSI-BLAST, we obtain $(50 + 20 \times\lambda + 3 \times l_{max})$ features in total, which will be fed into a SVM-based classifier to perform protein fold recognition.

Support vector machines

SVMs are one of the state-of-the-art machine learning algorithms for binary classification introduced by Vapnik.[30] SVMs have been extensively applied in various fileds and theoretical descriptions about SVMs abound in the literature. Protein fold recognition is a multiclass classification problem, which can be converted into binary classification problems by using either one-against-one or one-against-all strategy. For the implementation of SVMs, we use the LIBSVM package[31] with the one-against-one classification strategy.

There are four basic kernel functions commonly used by SVMs; that is, linear, polynomial, radial basis function (RBF), and sigmoid. Here, we choose the RBF kernel, because it produces higher prediction accuracy than other kernel functions (see e.g., Refs. 12, 13, 27). It is formally defined as

$$K(\mathbf{x}_i, \mathbf{x}_j) = e^{-\gamma||\mathbf{x}_i - \mathbf{x}_j||^2} \tag{10}$$

where γ is a kernel parameter, $\mathbf{x}_i$ and $\mathbf{x}_j$ are the feature vectors of proteins i and j, respectively. The values of γ and the cost parameter C (used to control the trade-off between allowing training errors and forcing rigid margins) are optimized based on grid search (see Supporting Information C for details).

The proposed method

Figure 3 illustrates the overall architecture of our proposed method called TAXFOLD. The query amino acid sequence submitted to TAXFOLD is first input into PSI-BLAST and PSIPRED to obtain the sequence evolution information and predicted secondary structure. Then, a comprehensive set of features are extracted from the output profiles of PSI-BLAST and PSIPRED, as described above. These features are finally fed into an SVM-based classifier for fold recognition. Compared with the existing taxonomy-based methods such as PFRES,[12] TAXFOLD

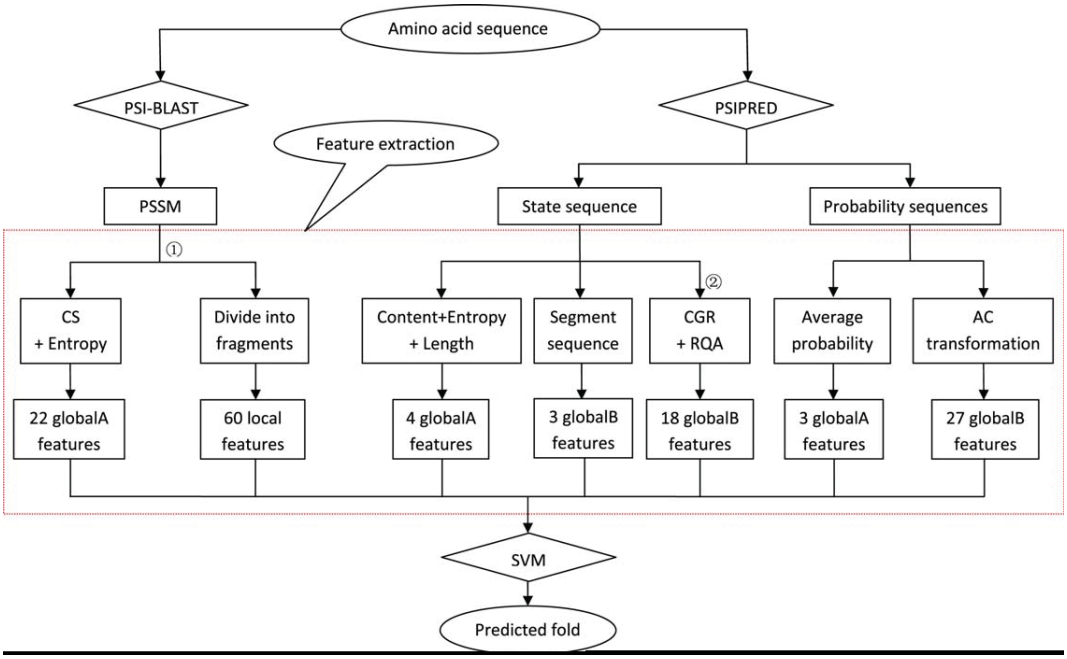

Figure 3

The overall architecture of TAXFOLD. The proposed set of features are enclosed inside a red box. The detailed procedures of 1 and 2 in the figure are already illustrated in Figures 1 and 2, respectively.

differs mainly by a novel set of features, which are so carefully designed that they are capable of capturing both global and local characteristics of PSI-BLAST and PSIPRED profiles. For the experiments in this study, we train SVM-based classifiers in TAXFOLD for three different numbers (27, 95, and 194) of folds.

For public use, we have developed a web server for TAXFOLD at http://www1.spms.ntu.edu.sg/~chenxin/TAXFOLD/. The top five predictions and their corresponding probability estimates are provided for users to determine the reliability of predictions.

Performance evaluation

Two metrics, precision and recall, are used to evaluate the performance of TAXFOLD on each individual fold. They are defined, respectively, as

$$p(i) = \frac{\mathrm{TP}(i)}{\mathrm{TP}(i) + \mathrm{FP}(i)}, \quad i = 1, 2, \cdots, \mu \qquad (11)$$

and

$$r(i) = \frac{\mathrm{TP}(i)}{\mathrm{TP}(i) + \mathrm{FN}(i)}, \quad i = 1, 2, \cdots, \mu \qquad (12)$$

where $\mathrm{TP}(i)$, $\mathrm{FP}(i)$, and $\mathrm{FN}(i)$ represent the numbers of true positives, false positives, and false negatives, respectively, and μ the total number of folds under consideration.

Note that the above metrics might overestimate the performance of a classifier in some cases. Therefore, a more robust metric called *F*-measure is often used, which is basically a harmonic mean of precision and recall as define below.

$$Fm(i) = \frac{2 \times p(i) \times r(i)}{p(i) + r(i)}, \quad i = 1, 2, \cdots, \mu \qquad (13)$$

The overall accuracy Q of a classifier is defined as the ratio of correctly predicted instances against all the tested instances.[9,13,16] It can be calculated as

$$Q = \frac{\sum_i^{\mu} \mathrm{TP}(i)}{\sum_i^{\mu} [\mathrm{TP}(i) + \mathrm{FN}(i)]}. \qquad (14)$$

It will be used to evaluate the overall performance of TAXFOLD.

RESULTS AND DISCUSSIONS

Optimal values of λ and l_{max}

As mentioned earlier, the values of λ and l_{max} remain to be determined. In this study, we choose their values by aiming to achieve the overall prediction accuracy as

Table I
Overall Accuracies (%) on Four Datasets Obtained with Different Combinations of Feature Subsets

Datasets	S1(29)	S2(48)	S3(60)	PSIPRED(54–55)	PSI-BLAST(82–83)	S1 + S2(77)	S1 + S3(89)	S2 + S3(108)	S1 + S2 + S3(137)
RDD	72.8	66.8	72.3	70.4/71.2	70.9/74.9	79.1	78.3	78.8	**83.2**
EDD	76	77.5	80	78.3/78.8	81.7/82.5	86.5	86.3	88.9	**90**
F95	62.2	61.2	72.3	62.2/63.1	74.1/75.5	75.3	78.3	81.5	**82.4**
F194	58.6	57	70.5	57.7/58.8	72.2/73	71.3	76.2	78.8	**79.6**

For the dataset RDD, the overall accuracies are obtained from the independent testing sequences. For the other datasets, the overall accuracies are obtained instead by applying 10-fold cross-validation. Enclosed in the parentheses are the number of features of the corresponding feature subset. The accuracies from feature subsets PSIPRED and PSI-BLAST are obtained with/without the feature of protein length.

high as possible. To this end, we run TAXFOLD on the training sequences of the dataset RDD and compute the overall accuracies Q s with varying values of λ and l_{max} based on five-fold cross-validation. Figure SC1 in Supporting Information C depicts all Q s obtained when λ ranges from 1 to 5 and l_{max} from 1 to 15. The highest accuracy of 77.2% is thus achieved when $\lambda = 3$ and $l_{max} = 9$. For the sake of simplicity and generality, we set the default values of λ and l_{max} in TAXFOLD to be 3 and 9, respectively, and all the rest experiments are carried out with these default values. This setting hence gives rise to a total of 137 ($= 50 + 20 \times 3 + 3 \times 9$) features among which 29, 48, and 60 are globalA, globalB, and local features, respectively (see Fig. 3).

Analysis of feature contribution

To investigate the contributions of features to the overall prediction accuracy, we divide the 137 features into three subsets: S1 (contains 29 globalA features), S2 (contains 48 globalB features), and S3 (contains the remaining 60 local features). In addition, to assess the contributions of predicted secondary structure and evolutionary information, we separate features generated from the PSIPRED profile and PSI-BLAST profile and denote them as PSIPRED and PSI-BLAST, respectively. The feature of protein length can be incorporated into both subsets.

Table I lists the overall prediction accuracies obtained with all the possible combinations of feature subsets S1, S2, and S3. It can be seen that when the feature subsets are used individually, the resulting overall prediction accuracies range from 57.7 to 82.5% for the four tested datasets. The feature subsets S1 and S2 perform comparatively well for all datasets, whereas the feature subset S3 performs the best for three datasets (EDD, F95, and F194). As more features are used, the overall accuracy values increase steadily. For instance, the combination of the feature subsets S2 and S3 achieves the accuracies of 78.8, 88.9, 81.5, and 78.8% for the four tested datasets, making at least 6.5% improvement over those obtained with any individual feature subset S2 or S3. If all these feature subsets are used together, the accuracy values increase to the maximum possible for all datasets (83.2, 90, 82.4, and 79.6%, respectively). Therefore, these three feature subsets could make complementary contributions to each other for improving protein fold recognition.

From Table I, we can see that the PSI-BLAST features seem to make higher contributions than the PSIPRED features for all tested datasets. This is is especially obvious for the F95 and F194 datasets, where the PSI-BLAST features achieve more than 10% improvement compared with the PSIPRED features. However, just using the PSI-BLAST features can not obtain the best prediction accuracy. As can be seen from Table I, when the PSIPRED features are combined with the PSI-BLAST features, the accuracies are improved by 8.3, 6.5, 6.9, and 6.6% for the four tested datasets, respectively. This suggests that the PSI-BLAST features and PSIPRED make complementary contributions to each other, and it is important to use both kinds of features for enhanced protein fold recognition. In addition, it is interesting to see that the feature of protein length improves the accuracy by about 1%, suggesting the necessariness of using this feature.

To demonstrate the advantage of features extracted from the consensus sequences rather than directly from the corresponding amino acid sequences (please refer to PSI–BLAST profile-based features Subsection), we compared their respective prediction accuracies. When the 21 globalA features are extracted from the original amino acid sequences (see Eqs. 3 and 4), the overall accuracies are only 51.3, 42.9, 27.5, and 22% for the four tested datasets. When these features are instead extracted from the consensus sequences, the prediction accuracies increase to 63.1, 62.7, 47.8, and 43.7%, respectively, making an average 18.4% improvement. These indicate that the consensus sequences contain much more evolution information than the amino acid sequences, thereby leading to more accurate protein fold recognition.

Accuracies for four tested datasets

TAXFOLD is evaluated on four datasets, RDD, EDD, F95, and F194, and the resulting overall accuracies are listed in the last column of Table I. The precision, recall, and F-measures of individual folds and the correspond-

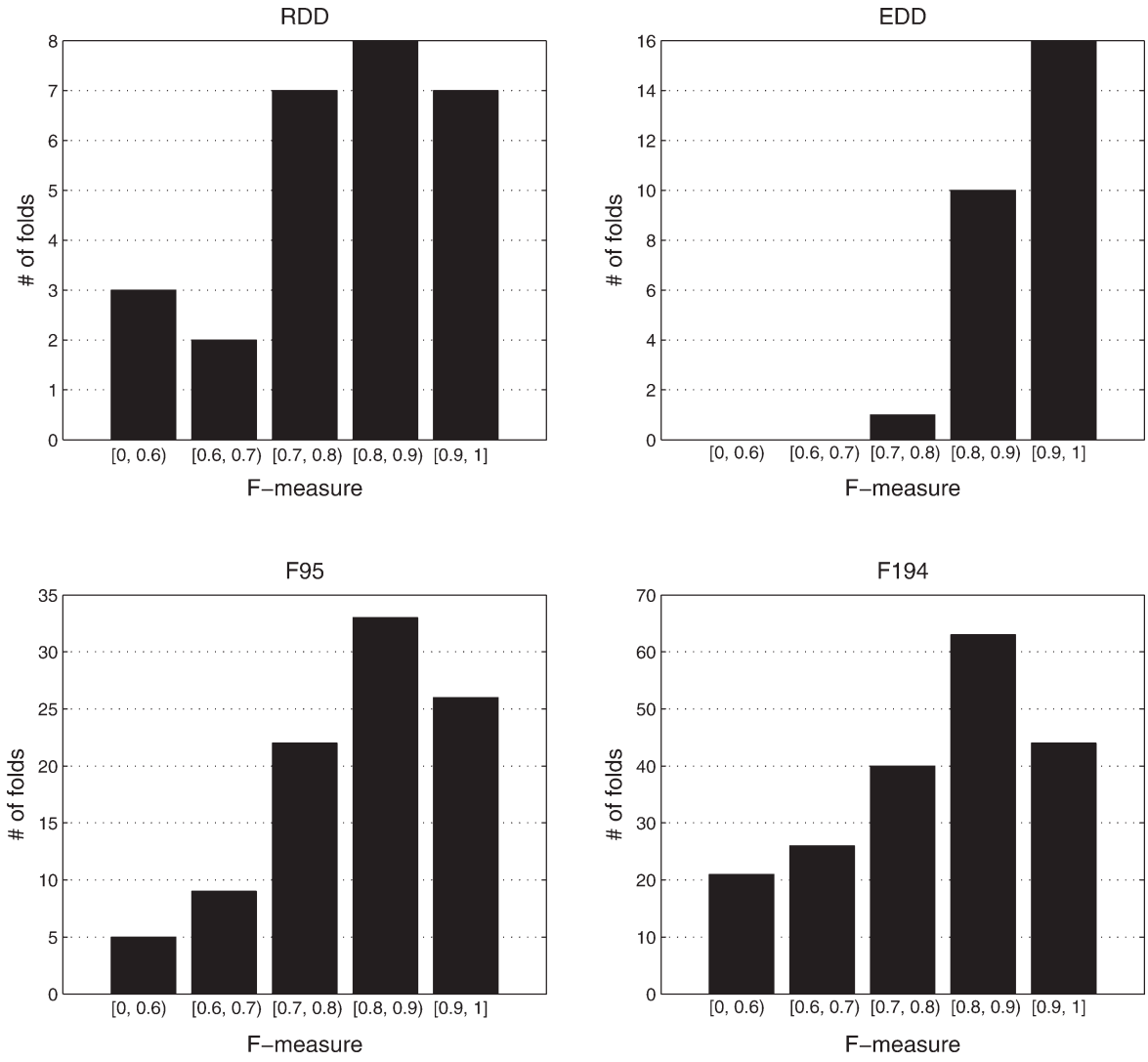

Figure 4

Histogram of the *F*-measure values for four tested datasets.

ing values of the parameters C and γ in SVMs are presented in Supporting Information C.

We can see that the overall accuracies for the first three datasets are all above 80% (i.e., 83.2, 90 and 82.4%, respectively). It is commonly believed that, the larger number of training sequences, the more reliable a classifier to be trained. Although both RDD and EDD are made of sequences from the same 27 folds, the latter contains more sequences than the former. Therefore, it is not surprising to see that the overall accuracy for EDD is higher than that for the dataset RDD. Moreover, as revealed in Ref. 13, the prediction becomes very challenging as more folds are considered. This can also be observed from Table 1, where the accuracy value decreases from 90 to 82.4% and further to 79.6% when the number of folds increase from 27 to 95 and 194, respectively.

We also performed histogram analysis on the *F*-measure values over all the folds. We count the number of folds that have *F*-measure values in one of the following five intervals: [0, 0.6), [0.6, 0.7), [0.7, 0.8), [0.8, 0.9), [0.9, 1]. The resulting histograms are shown in Figure 4. Note that among the 27 folds used for the dataset RDD, 22 obtained the *F*-measures over 70%. When more sequences of these 27 folds are experimented in the dataset EDD, the *F*-measures become over 70% for all the folds. Indeed, all the folds achieve a *F*-measure value over 80%, except for the fold *OB-fold*. These *F*-measure results further indicate that TAXFOLD can achieve superior performance for this 27-class fold recognition problem.

Table II
Comparisons of prediction accuracies (%) obtained from the original DD dataset and its revised dataset RDD

Methods	References	Q^a	Q^b
D-D	9	56	NA
PFP-Pred	10	62.1	NA
GAOEC	11	64.7	NA
Multi-kernel	19	68.1	NA
PFP-FunDSeqE	18	70.5	NA
SWPSSM[c]	32	67.8	NA
Shamim	16	60.5	66.2[d]
ACCFold_AC	13	65.3	73.6[e]
ACCFold_ACC	13	66.6	73.8[f]
PFRES	12	68.4	80.1
TAXFOLD	This article	**71.5**	**83.2**

Note that the prediction accuracies for the original DD dataset are taken directly from their corresponding references, except for the methods ACCFold_AC and ACCFold_ACC.
[a]Overall accuracies for the original DD dataset.
[b]Overall accuracies for the revised dataset RDD.
[c]Cited from Ref. 13.
[d]The accuracy obtained with the feature set Feature4.
[e]The accuracy obtained with the parameter LG being 8.
[f]The accuracy obtained with the parameter LG being 10.

TAXFOLD can still achieve satisfactory prediction for 95 and 194 folds, although the *F*-measures get relatively lower values. For the five folds in the dataset F95, their *F*-measures are all below 60%. For the dataset F194, there are 21 folds with F-measures less than 60%. Among these 21 folds, 11 have no more than 20 sequences. This might partially explain their low *F*-measures because it is generally very hard to train a reliable classifier on a small number of instances. It is thus anticipated that the performance of TAXFOLD on a large number of folds would be improved as more amino acid sequences are accumulated.

Comparisons with existing taxonomy-based methods

To demonstrate the effectiveness of the proposed method, we compare TAXFOLD with several major existing taxonomy-based methods on four datasets.

The original DD dataset has been widely used to evaluate various methods. For a fair comparison, we tested TAXFOLD on both the original DD dataset and its revised version RDD. Both the accuracies that were reported in the literature and those obtained from our experiments are listed in Table II. It should be noticed that the reported accuracies of ACCFold_AC,[13] ACCFold_ACC,[13] and Shamim[16] were measured by applying two-fold cross-validation on the training sequences of the original DD dataset, while the reported accuracies of the other methods were obtained with the independent testing sequences. So, we ran ACCFold_ACC on the testing sequences of the original DD dataset,[‡] and obtained its prediction accuracy (66.6%), which turns out to be slightly lower than the accuracy (70.1%) measured via two-fold cross-validation. Besides, we ran four major existing methods, PFRES,[12] ACCFold_AC,[13] ACCFold_ACC,[13] and Shamim[16] on the revised DD dataset RDD and collected their overall prediction accuracies in Table II. Their corresponding precision, recall, and *F*-measures can be found in Supporting Information C. It is evident from Table 2 that, when tested on the original DD dataset, TAXFOLD achieves an accuracy of 71.5%, which is at least 1% higher than that of any other method. When tested on the revised DD dataset RDD, all the tested methods had their accuracies increased. However, TAXFOLD still achieves the highest overall accuracy of 83.2%, which is 3.1–17% higher than those of the other tested methods.

We further tested the above methods on the datasets EDD, F95, and F194. However, we found that the running of ACCFold_ACC and PFRES on these large-sized datasets with 10-fold cross-validation seems no way to complete in a reasonable amount of time (executed on a workstation with eight CPUs of 2.8 GHz each and 24-GB RAM). Therefore, to make the comparisons possible, we adopt two-fold cross-validation rather than 10-fold cross-validation for them. Even with two-fold cross-validation, it is still too challenging to complete the test of PFRES on the largest dataset F194. Note that PFRES relies on an ensemble classifier, which consists of six individual classifiers, (SVMs, multiple logistic regression, instance learning-based Kstar, IB1 algorithms, Navïe Bayes, and random forest), and it was reported that this ensemble classifier made only a slight accuracy improvement over the individual classifier random forest (68.4 vs. 66.8%) when tested on the original DD dataset.[12] On the basis of these observations, to run PFRES on the dataset F194, we activate the classifier random forest only. Although the resulting accuracy might be biased against PFRES, we include it for the sake of a complete comparison. On the other hand, for the method Shamim, the feature set Feature4 as described in Ref. 16 can yield the best prediction on the original DD dataset. So, this feature set is applied to the other three datasets as well. Table III lists the overall accuracies obtained with these methods. The corresponding precision, recall, and *F*-measures are presented in Supporting Information C.

In general, 10-fold cross-validation shall yield a higher accuracy value than two-fold cross-validation because

Table III
Comparisons of Overall Accuracies (%) on Three Large-Sized Datasets

Datasets	Shamim	ACCFold_AC	ACCFold_ACC	PFRES	TAXFOLD
EDD	61	73.9	77.3	81.1	**86.9**
F95	41.6	62.5	71.8	68	**76.5**
F194	35.4	58.6	68.8	60[a]	**72.6**

These accuracies are obtained via two-fold cross validation.
[a]The accuracy obtained with the classifier random forest alone.

[‡]The PSSMs were acquired from the authors.

Table IV
Comparisons with Other Methods on the Four Benchmark Datasets

	RDD				EDD				F95				F194			
Methods	W	E	B	*P*-values	W	E	B	*P*-values	W	E	B	*P*-values	W	E	B	*P*-values
Shamim	0	1	26	3.75e-9	0	0	27	1.42e-10	0	0	95	5.41e-30	0	3	191	1.21e-48
ACCFold_AC	2	3	22	4.36e-5	0	0	27	1.01e-8	3	0	92	8.32e-15	25	13	156	2.38e-7
ACCFold_ACC	5	4	18	0.003	0	0	27	7.41e-4	11	1	83	7.36e-6	79	21	94	0.541
PFRES	6	6	15	0.013	3	1	23	2.12e-5	2	0	93	9.99e-13	15	7	172	9.98e-12

RDD represents revised DD dataset. The columns marked by W, E, and B mean respectively the number of folds that TAXFOLD has worse, equal, and better performance compared with the corresponding method. The *p* -values are used to measure the statistical significance.

more sequences are used to train a classifier in the former case. It is true for our method TAXFOLD as we can see from Tables I and III, and so is for the methods Shamim, ACCFold_AC, ACCFold_ACC, as the accuracy values obtained with two-fold cross-validation are lower than those reported in the literature, which were instead obtained with five-fold cross-validation.

It is evident from Table III that TAXFOLD outperforms any other method for all the tested datasets. Shamim performs the worst among these methods, which we believe is due to the fact that it does not exploit PSI-BLAST profiles. Therefore, we conclude that the sequence evolution information is a key factor for successful protein fold recognition. PFRES performs better than ACCFold_ACC for the dataset EDD but worse for two datasets F95 and F194. It might be explained in part by the fact that PFRES was developed (trained) particularly for the recognition of 27 folds. Compared with other methods, TAXFOLD achieves 5.8–25.9%, 4.7–34.9%, and 3.8–37.2% improvements for the datasets EDD, F95, and F194, respectively. We believe that the superior performance of TAXFOLD shall be attributed to a comprehensive set of features developed, which is capable of capturing both global and local characteristics of PSI-BLAST and PSIPRED profiles of a query protein domain.

Further comparisons are made on *F*-measures of the individual folds. We count the number of folds for which TAXFOLD has a lower (respectively, equal or higher) *F*-measure value than any other methods. It implies that for the folds to be counted, TAXFOLD performs worse (respectively, comparatively or better) than the method under comparison. The detailed tally is shown in Table IV, where we can see that TAXFOLD performs better for a majority of folds in all cases.

Moreover, statistical test is applied to the *F*-measure values of TAXFOLD and other methods to assess their statistical significance as follows. Shapiro-Wilk test is first used to determine if the samples are normally distributed (at 0.05 significance level). If they follow normal distri-

Table V
Comparisons with Template-Based Methods on the Lindahl Benchmark Dataset

	Family			Superfamily			Fold		
	Subset0	Subset1	Subset2	Subset0	Subset1	Subset2	Subset0	Subset1	Subset2
	$N_{min} = 1$	$N_{min} = 3$	$N_{min} = 4$	$N_{min} = 1$	$N_{min} = 3$	$N_{min} = 5$	$N_{min} = 1$	$N_{min} = 3$	$N_{min} = 5$
No. of sequences	555	97	47	434	225[a]	91[a]	321	239[a]	177
No. of categories	176	13	5	86	23[a]	6[a]	38	16[a]	8
TAXFOLD	68.6	90.7	100	39.3	61.7	84.5	40.6	56.9	67.7
ACCFold_ACC[b]	53.9	79.6	95.7	23.1	55.4	78.3	29.9	41.4	51.9
ACCFold_AC[b]	53.1	79.5	93.6	20	47.7	64	28	41.3	50.9
RAPTOR[b]	86.6			56.3			38.2		
Fugue[c]	82.2			41.9			12.5		
HHPred[c]	82.9			58.8			25.2		
SPARKS[c]	81.6			52.5			24.3		
SP5[c]	82.4			59.8			37.9		
FOLDPro[c]	85			55.5			26.5		
DescFold_I[c]	80.7			57.8			24.9		
DescFold_II[c]	81.1			60.6			32.4		
BoostThreader[d]	86.5			66.1			42.6		

Note that the only the top 1 accuracies (%) are reported for all methods.
[a]Slightly different from those in Ref. 13 because of random partition of the dataset into two subsets for cross-validation. In our partition, we tried to make the number of samples in each subset of the same category as even as possible.
[b]The results were cited from Ref. 13.
[c]The results were cited from Ref. 6.
[d]The results were cited from Ref. 33.

bution, the paired t -test is then used for statistical test on the differences between TAXFOLD and other methods; otherwise, the nonparametric Wilcoxon rank sum test is used. The resulting P-values are also reported in Table IV. We can see that the P-values are smaller than 0.05 for all the tested datasets and methods except the case of ACCFold_Acc on the F194 dataset, indicating that TAXFOLD has made statistically significant improvements over any other method for fold recognition.

Comparisons with template-based methods

TAXFOLD is further compared with the conventional template-based threading methods: RAPTOR,[2] HHPred,[3] FOLDPro,[4] SP5,[5] DescFold,[6] and BoostThreader.[33] To this end, the Lindahl benchmark dataset[24] is used, and two-fold cross-validation is adopted to assess the accuracy. As done in Ref. 13, before performing predictions, the dataset is preprocessed so that the number of proteins in each category is larger than or equal to a threshold $N_{\min}$. For details about this preprocessing, please refer to Ref. 13.

The accuracies of TAXFOLD, ACCFold_AC, ACCFold_ACC, and eight template-based methods are listed in Table V. First, we can see that TAXFOLD performs consistently better than ACCFold_AC and ACCFold_ACC at each SCOP level. Second, for the family and superfamily levels, when the number of samples in each category is very small, the taxonomic methods including TAXFOLD perform worse than the template-based threading methods. This is not surprising because a few samples in general do not allow us to train an accurate classifier. When the number of samples increases, the performance of taxonomic methods is improved, which can be seen from the accuracy increase from Subset0 to Subset1 and Subset2. Nevertheless, at the fold level, TAXFOLD achieved an accuracy comparable with the best template-based threading method (BoostThreader; for Subset0), still showing that TAXFOLD is very promising for protein fold recognition.

CONCLUSIONS

In this study, we have developed a new taxonomy-based method called TAXFOLD for protein fold recognition. It extensively exploits the sequence evolution information from PSI-BLAST profiles[22] and the secondary structure information from PSIPRED profiles.[23] A comprehensive set of 137 features is thus constructed, which has been demonstrated capable of depicting both global and local characteristics of profiles. We notice that some computational methods already proposed to exploit PSI-BLAST and PSIPRED profiles for feature extraction (e.g., Ref. 12). However, we carried out this task in quite a few different ways, as briefly summarized below.

1. We deal with the negative elements of PSSMs by applying an inverse algorithmic operation rather than simply replacing all of them by zero. The later would lose sequence evolution information to some extent.
2. We extract features from the consensus sequences constructed from PSSMs rather than from their respective amino acid sequences. The former retains richer sequence evolution information.
3. We divide PSI-BLAST profiles into several nonoverlapping fragments, which allows for the depiction of local characteristics (e.g., at N-terminus and C-terminus).
4. We reduce the state sequences of PSIPRED profiles into the so-called segment sequences, which allows us to characterize the spatial arrangements of α helices and β strands.
5. We apply AC transformation to the probability sequences of PSIPRED profiles for feature extraction. To our best knowledge, this is the first attempt to extract features from the probability sequences.

Four datasets (RDD, EDD, F95, and F194) are used to test and compare the proposed method TAXFOLD with the major existing methods. The first two datasets contains protein domain sequences from 27 folds, and the third and fourth contains sequences from 95 and 194 folds, respectively, representing different levels of challenges that we might face for the task of fold recognition. Our experiments show that, among all the tested methods, TAXFOLD achieves the highest overall accuracies of 83.2, 90, 82.4 and 79.6% respectively for the four tested datasets, making an average accuracy improvement of 6.9% over the best available method. We further tested TAXFOLD with a dataset containing a large number of folds (710 folds, see Supporting Information C). The resulting overall accuracy is relatively low (68.1%), because for as many as 185 fold types there are less than 10 sequences. Comparisons on the Lindahl benchmark dataset shows that TAXFOLD performs comparably with the best conventional template-based threading methods at the SCOP fold level. These together clearly indicate that the proposed set of features is highly beneficial to protein fold recognition. Therefore, we believe that TAXFOLD is a promising and practical tool for protein fold recognition. For public use, a web server for TAXFOLD is freely accessible at http://www1.spms.ntu.edu.sg/~chenxin/TAXFOLD/.

ACKNOWLEDGMENTS

The author thank Dr. Qiwen Dong for sending the PSSMs of the original DD dataset and useful discussions on the experimental tests of ACCFold via email.

REFERENCES

1. Jones DT, Taylort WR, Thornton JM. A new approach to protein fold recognition. Nature 1992;358:86–89.
2. Xu J, Li M, Kim D, Xu Y. RAPTOR: optimal protein threading by linear programming. J Bioinf Comput Biology 2003;1:95–118.

3. Söding J, Biegert A, Lupas AN. The HHpred interactive server for protein homology detection and structure prediction. Nucl Acids Res 2005;33:W244–W248.
4. Cheng J, Baldi P. A machine learning information retrieval approach to protein fold recognition. Bioinformatics 2006;22: 1456–1463.
5. Zhang W, Liu S, Zhou Y. SP^5: improving protein fold recognition by using torsion angle profiles and profile-based gap penalty model. PLoS ONE 2008;3:e2325.
6. Yan RX, Si JN, Wang C, Zhang Z. DescFold: a web server for protein fold recognition. BMC Bioinformatics 2009;10:416.
7. Dubchak I, Muchnik I, Holbrook SR, Kim SH. Prediction of protein folding class using global description of amino acid sequence. Proc Nat Acad Sci 1995;92:8700–8704.
8. Dubchak I, Muchnik I, Mayor C, Dralyuk I, Kim SH. Recognition of a protein fold in the context of the SCOP classification. Proteins: Struc Func Bioinf 1999;35:401–407.
9. Ding CHQ, Dubchak I. Multi-class protein fold recognition using support vector machines and neural networks. Bioinformatics 2001; 17:349–358.
10. Shen HB, Chou KC. Ensemble classifier for protein fold pattern recognition. Bioinformatics 2006;22:1717–1722.
11. Guo X, Gao X. A novel hierarchical ensemble classifier for protein fold recognition. Protein Eng Des Select 2008;21:659–664.
12. Chen K, Kurgan L. PFRES: protein fold classification by using evolutionary information and predicted secondary structure. Bioinformatics 2007;23:2843–2850.
13. Dong Q, Zhou S, Guan J. A new taxonomy-based protein fold recognition approach based on autocross-covariance transformation. Bioinformatics 2009;25:2655–2662.
14. Murzin AG, Brenner SE, Hubbard T, Chothia C. SCOP: a structural classification of proteins database for the investigation of sequences and structures. J Mol Biol 1995;247:536–540.
15. Chou KC. Prediction of protein cellular attributes using pseudo-amino acid composition. Proteins: Struct Funct Gene 2001;43: 246–255.
16. Shamim MTA, Anwaruddin M, Nagarajaram HA. Support vector machine-based classification of protein folds using the structural properties of amino acid residues and amino acid residue pairs. Bioinformatics 2007;23:3320–3327.
17. Deschavanne P, Tufféry P. Enhanced protein fold recognition using a structural alphabet. Proteins: Struct Funct Bioinf 2009;76: 129–137.
18. Shen HB, Chou KC. Predicting protein fold pattern with functional domain and sequential evolution information. J Theor Biol 2009; 256:441–446.
19. Damoulas T, Girolami MA. Probabilistic multi-class multi-kernel learning: on protein fold recognition and remote homology detection. Bioinformatics 2008;24:1264–1270.
20. Ying Y, Huang K, Campbell C. Enhanced protein fold recognition through a novel data integration approach. BMC Bioinformatics 2009;10:267.
21. Bologna G, Appel RD.A comparison study on protein fold recognition. In Proceedings of the 9th International Conference on Neural Information Processing, Vol. 5, Singapore, 2002. pp 2492–2496.
22. 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. Nucl Acids Res 1997;25: 3389–3402.
23. Jones DT. Protein secondary structure prediction based on position-specific scoring matrices. J Mol Biol 1999;292:195–202.
24. Lindahl E, Elofsson A. Identification of related proteins on family, superfamily and fold level. J Mol Biol 2000;295:613–625.
25. Patthy L. Detecting homology of distantly related proteins with consensus sequences. J Mol Biol 1987;198:567–577.
26. Yang JY, Peng ZL, Chen X. Prediction of protein structural classes for low-homology sequences based on predicted secondary structure. BMC Bioinformatics 2010;11:S9.
27. Kurgan L, Cios K, Chen K. SCPRED: Accurate prediction of protein structural class for sequences of twilight-zone similarity with predicting sequences. BMC Bioinformatics 2008;9:266.
28. Mizianty MJ, Kurgan L. Modular prediction of protein structural classes from sequences of twilight-zone identity with predicting sequences. BMC Bioinformatics 2009;10:414.
29. Yang JY, Peng ZL, Yu ZG, Zhang RJ, Anh V, Wang D. Prediction of protein structural classes by recurrence quantification analysis based on chaos game representation. J Theor Biol 2009;257:618–626.
30. Vapnik VN. The nature of statistical learning theory. New York: Springer Verlag; 1995.
31. Chang CC, Lin CJ. LIBSVM: a library for support vector machines, Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm, 2001.
32. Rangwala H, Karypis G. Profile-based direct kernels for remote homology detection and fold recognition. Bioinformatics 2005; 21:4239–4247.
33. Peng J, Xu J. Boosting protein threading accuracy. Lec Notes Comput Sci 2009;5541:31–45.