Bioinformatics Advance Access published December 30, 2016

*Bioinformatics*, 2016, 1–8
doi: 10.1093/bioinformatics/btw768
Original Paper

Structural bioinformatics

# An ensemble approach to protein fold classification by integration of template-based assignment and support vector machine classifier

**Jiaqi Xia[1], Zhenling Peng[2], Dawei Qi[1], Hongbo Mu[1,]* and Jianyi Yang[3,]***

[1]Department of Physics, Northeast Forestry University, Harbin 150040, China, [2]Center for Applied Mathematics, Tianjin University, Tianjin 300072, China and [3]School of Mathematical Sciences, Nankai University, Tianjin 300071, China

*To whom correspondence should be addressed.

Associate Editor: Anna Tramontano

Received on August 26, 2016; revised on November 8, 2016; editorial decision on November 25, 2016; accepted on December 2, 2016

## Abstract

**Motivation:** Protein fold classification is a critical step in protein structure prediction. There are two possible ways to classify protein folds. One is through template-based fold assignment and the other is *ab-initio* prediction using machine learning algorithms. Combination of both solutions to improve the prediction accuracy was never explored before.

**Results:** We developed two algorithms, HH-fold and SVM-fold for protein fold classification. HH-fold is a template-based fold assignment algorithm using the HHsearch program. SVM-fold is a support vector machine-based *ab-initio* classification algorithm, in which a comprehensive set of features are extracted from three complementary sequence profiles. These two algorithms are then combined, resulting to the ensemble approach TA-fold. We performed a comprehensive assessment for the proposed methods by comparing with *ab-initio* methods and template-based threading methods on six benchmark datasets. An accuracy of 0.799 was achieved by TA-fold on the DD dataset that consists of proteins from 27 folds. This represents improvement of 5.4–11.7% over *ab-initio* methods. After updating this dataset to include more proteins in the same folds, the accuracy increased to 0.971. In addition, TA-fold achieved >0.9 accuracy on a large dataset consisting of 6451 proteins from 184 folds. Experiments on the LE dataset show that TA-fold consistently outperforms other threading methods at the family, superfamily and fold levels. The success of TA-fold is attributed to the combination of template-based fold assignment and *ab-initio* classification using features from complementary sequence profiles that contain rich evolution information.

**Availability and Implementation:** http://yanglab.nankai.edu.cn/TA-fold/

**Contact:** yangjy@nankai.edu.cn or mhb-506@163.com

**Supplementary information:** Supplementary data are available at *Bioinformatics* online.

## 1 Introduction

The term ‘fold’ describes the overall topology of a protein’s three-dimensional (3D) structure. For example, the well-known TIM barrel fold, in the shape of a doughnut, is one of the most common folds. It contains eight parallel $\beta$-strands forming the inner wall of the doughnut, and eight $\alpha$-helices forming the outer wall. It was estimated that there are 1000–2000 folds in nature (Zhang and DeLisi, 1998), though this number may be changed with a different estimation method (Liu *et al.*, 2004). The Structural Classification of Proteins (SCOP) database is a hierarchical classification of the protein domain structures at the following levels: class, fold, superfamily, family, protein and species (Fox *et al.*, 2014). There are 1431-fold types in the latest version of SCOP (release 2.06, 2016-07-21). Another popular resource for protein structural classification is CATH (Class, Architecture, Topology and Homology) (Sillitoe *et al.*, 2015), in which the Topology level corresponds to the fold level of SCOP.

In this work, the gold standard for protein fold classification is taken from SCOP rather than CATH based on the following observations. First, SCOP was created largely based on manual investigation while CATH was by a combination of manual and automated computation. Second, some CATH folds are further refined in SCOP. For example, the 3-layer ($\alpha\beta\alpha$) sandwich fold in CATH is divided into at least 16 different folds in SCOP (Hadley and Jones, 1999). Third, the majority of previous studies for protein fold classification are based on SCOP (Chen and Kurgan, 2007; Damoulas and Girolami, 2008; Ding and Dubchak, 2001; Dong *et al.*, 2009; Guo and Gao, 2008; Lyons *et al.*, 2015; Rangwala and Karypis, 2005; Shamim *et al.*, 2007; Sharma *et al.*, 2013; Shen and Chou, 2006; Shen and Chou, 2009; Yang and Chen, 2011).

The problem of protein fold classification is a typical classification problem, which aims to classify proteins into one of the known folds using the amino acid sequence information. We would like to mention that the problem of protein fold classification was also called 'fold recognition' by some previous work (Dong *et al.*, 2009; Yang and Chen, 2011; Zakeri *et al.*, 2014). However, the term fold recognition (known as threading) is more commonly used in template-based protein structure prediction, which means to match amino acid sequence with 3D structures (Jones *et al.*, 1992; Yang *et al.*, 2015). To avoid any confusion, we do not use the term fold recognition here.

Intensive efforts have been spent on the problem of protein fold classification. The pioneer work was done by Ding and Dubchak (Ding and Dubchak, 2001), to classify proteins into 27 SCOP folds, which received many follow-up studies and the accuracy was gradually improved from 0.56 to >0.7 (Chen and Kurgan, 2007; Damoulas and Girolami, 2008; Dong *et al.*, 2009; Guo and Gao, 2008; Lyons *et al.*, 2015; Rangwala and Karypis, 2005; Shamim *et al.*, 2007; Sharma *et al.*, 2013; Shen and Chou, 2006; Shen and Chou, 2009; Yang and Chen, 2011). These methods belong to the machine learning-based *ab-initio* approach, in which there are two key aspects, i.e. method for feature extraction and selection of appropriate classification algorithms. It turns out that the most informative features are those extracted from the evolutionary information and predicted secondary structure (Chen and Kurgan, 2007; Cheung *et al.*, 2016; Wei *et al.*, 2015; Yang and Chen, 2011; Zakeri *et al.*, 2014). Regarding to classification algorithms, support vector machines (SVMs) have been aggressively used, such as in (Ding and Dubchak, 2001), PFRES (Chen and Kurgan, 2007), iFC$^2$ (Chen *et al.*, 2011), ACCFold (Dong *et al.*, 2009) and TAXFOLD (Yang and Chen, 2011). Other algorithms employed include neural networks (Cheung *et al.*, 2016; Ding and Dubchak, 2001; Huang *et al.*, 2003), hidden Markov models (Deschavanne and Tuffery, 2009) and ensemble classifiers (Shen and Chou, 2006; Wei *et al.*, 2015).

In this work, we aim to improve the accuracy of protein fold classification through integrating template-based prediction and machine learning-based *ab-initio* classification. First, we develop HH-fold for template-based fold assignment using the tool HHsearch (Soding, 2005). Second, a machine learning-based *ab-initio* classification algorithm SVM-fold is designed, in which a comprehensive set of features are extracted from three complementary sequence profiles that contain rich evolution information. These features are fed into SVM to classify protein fold types. Finally, an ensemble approach TA-fold is proposed to combine the results of HH-fold and SVM-fold. These methods are evaluated and compared with both *ab-initio* and template-based methods on widely used benchmark datasets.

**Table 1.** The information about the benchmark datasets

| Dataset | #Fold | #Seq. | ID.[a] | Reference |
|---|---|---|---|---|
| DD | 27 | 311/384[b] | 0.35 | (Ding and Dubchak, 2001) |
| RDD | 27 | 311/380[b] | 0.35 | (Yang and Chen, 2011) |
| EDD | 27 | 3397 | 0.4 | (Yang and Chen, 2011) |
| TG | 30 | 1612 | 0.25 | (Taguchi and Gromiha, 2007) |
| F184 | 184 | 6451 | 0.25 | This Article |
| LE | 330 | 976 | 0.4 | (Lindahl and Elofsson, 2000) |

[a] Maximum pairwise sequence identity;
[b] number of sequences in the training/test set.

## 2 Materials and methods

### 2.1 Benchmark datasets

Six benchmark datasets are used to assess and compare our method with others: DD, RDD, EDD, TG, F184 and LE, with information summarized in Table 1. The proteins in the DD dataset are from 27 SCOP folds and were divided into training and independent test sets consisting of 311 and 384 proteins, respectively (Ding and Dubchak, 2001). This dataset was found to be inconsistent with the updated SCOP database (Chen and Kurgan, 2007; Shen and Chou, 2006; Yang and Chen, 2011), and a revised version was named RDD (Yang and Chen, 2011). For the DD and RDD datasets, the sequence identity between the test and training set is <35%. Furthermore, inclusion of more proteins in the same 27 folds of the DD dataset results to an extended DD dataset (EDD, 3397 domains) (Yang and Chen, 2011). The pairwise sequence identity between proteins in the EDD dataset is <40%. The fourth one is the TG dataset consisting of 1612 domains from 30 SCOP folds (Taguchi and Gromiha, 2007), which has <25% pairwise sequence identity. In order to make our method work for proteins from more folds (i.e. not just 27-folds), the fifth dataset was constructed from the latest version (release 2.06) of the SCOPe database (Fox *et al.*, 2014) as follows. First, domain sequences (using the option 'PDB SEQRES records') with less than 25% sequence identity were fetched. A total of 8679 sequences from 1222 folds were obtained. Then to have enough samples for training purpose, we filtered out those folds with <10 sequences, resulting to 6451 sequences from 184 folds. We name this dataset by F184 for convenience. The last dataset LE is from the work of Lindahl and Elofsson to recognize proteins at the SCOP family, superfamily and fold levels (Lindahl and Elofsson, 2000). This dataset is mainly used to compare our methods with other template-based threading methods. The amino acid sequences and profiles of proteins in these datasets are available for download at: http://yanglab.nankai.edu.cn/TA-fold/benchmark.

### 2.2 HH-fold for template-based fold assignment

As shown in Figure 1, the sequence of the query protein is aligned to the training proteins by the HHsearch program (Soding, 2005), one of the most popular profile-profile alignment algorithms. The profile used by HHsearch is represented in the form of a hidden Markov model (HMM). The parameters for running the HHsearch program are set to the default by './hhsearch –i query.hhm –d train.hhm',

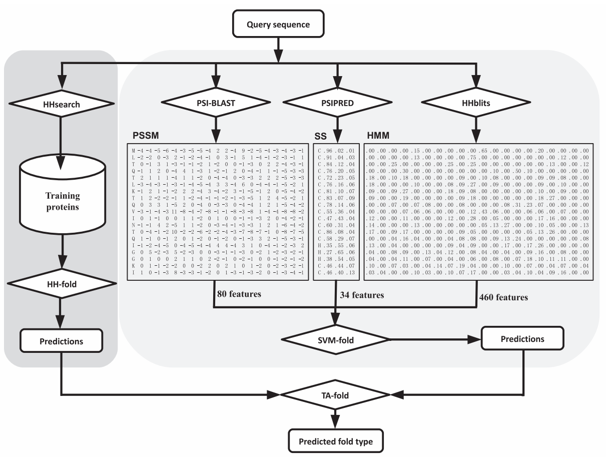

**Fig. 1.** The architecture of HH-fold, SVM-fold and TA-fold. The left/right hand side panel with darker/lighter background is the flowchart for HH-fold/SVM-fold. TA-fold is a combination of HH-fold and SVM-fold. Note that the elements shown in the HMM profile are between 0 and 1, which were converted from the original HMM profile

where query.hhm and train.hhm are the HMMs of the query and training proteins generated by the program HHblits (please refer to the next section for more information about HHblits). The folds of the top hits in the training set are then transferred to the query protein. We name this approach by HH-fold.

The 'Prob' column of the output returned by HHsearch measures the probability that the query and the corresponding proteins (called templates) in the training set share the same fold. In general, a higher probability value indicates the templates are more homologous to the query. Suppose there are $\mu$ possible folds in the training dataset. The likelihood $l_i$ that the query protein belongs to the $i$th fold is estimated as follows.

$$l_i = \frac{\sum_{j=1}^{i_n} p_{ij}}{\sum_{k=1}^{N} \sum_{j=1}^{k_n} p_{kj}}, i = 1, 2, \ldots, \mu \qquad (1)$$

where $p_{ij}$ is the probability for the $j$th template in the $i$th fold; $i_n$ ($k_n$) is the number of templates (from the top $N$ hits) belonging to the $i$th ($k$th) fold. The query protein is then assigned to the fold that obtains the maximum likelihood. The number $N$ remains to be determined later by experiment.

## 2.3 SVM-fold for machine learning-based *ab-initio* fold classification

It is apparent that HH-fold relies on the availability of homologous templates. When such condition is not satisfied, we need to refer to *ab-initio* prediction algorithms that do not use templates as HH-fold does not work anymore. As shown in Figure 1, the query sequence is submitted to three programs, PSI-BLAST (Altschul *et al.*, 1997), PSIPRED (Jones, 1999) and HHblits (Remmert *et al.*, 2012), to generate three complementary sequence profiles. These profiles are supposed to contain rich evolution information of the query protein. Then a comprehensive set of features is extracted from these profiles to encode the query. The resulting feature vector is finally fed into SVM to predict the query protein's fold. We name this method by SVM-fold. The sequence profiles and feature extraction are described below in details. Let $L$ denote the number of amino acids in a protein.

**Features from PSI-BLAST profile.** The query sequence is searched by the sequence-profile alignment tool PSI-BLAST (Altschul *et al.*, 1997) (with parameters '-j 3 -h 0.001') through the NCBI's non-redundant sequence database, where the profile is represented in the form of a position-specific scoring matrix (PSSM) of dimension $L$x20. Similar to (Yang and Chen, 2011), each integer $s_{ij}$ at the $i$th row and the $j$th column of the PSSM is converted to frequency using the inverse transform $f_{ij} = 2^{0.1 \times sij}$, which is normalized using the following formula:

$$M_{ij} = 100 \times \frac{f_{ij}}{\sum_{j=1}^{20} f_{ij}}, i = 1, 2, \ldots, L; j = 1, 2, \ldots, 20 \qquad (2)$$

The autocovariance (AC) transform, a statistical tool in time series analysis, is used to extract features from the PSI-BLAST profile. It was first applied to the analysis of biopolymer sequences in (Wold *et al.*, 1993). This transform was shown to be able to improve the accuracy of protein fold classification in Dong *et al.* (2009) and Yang and Chen (2011). Thus in this work, it is also applied to the 20 time series, each from one column of the normalized PSSM matrix. AC is the covariance of a sequence against a time-shifted version of itself. That is, for a time series $t = (t_1, t_2, \ldots, t_L)$, its AC transform will return

$$AC_l = \frac{1}{L-l} \sum_{i=1}^{L-l} (t_i - \bar{t})(t_{i+l} - \bar{t}), l = 1, 2, \ldots, l_{\max} \qquad (3)$$

where $\bar{t}$ is the average over all $t_i$, $l$ is the lag between two positions along the sequence, and $l_{max}$ is the maximum of $l$. The values for $AC_1, AC_2, \ldots, AC_{lmax}$ are then used as features. The value of $l_{max}$ is set to 4 based on 10-fold cross-validation on the RDD training set (Supplementary Fig. S1). Thus, a total of 80(=20 × 4) features are extracted from the PSI-BLAST profile.

**Features from PSIPRED profile.** We use the tool PSIPRED (Jones, 1999) to predict the three-state secondary structure (SS) profile. The three states are $\alpha$-helix (H), $\beta$-strand (E) and random coil (C). This profile provides the predicted state for each residue and the corresponding probability of folding into each state. Thus the dimension of the SS profile is $L$x4. The first three features from PSIPRED profile are the contents of three SS states, calculated by

$$P_H = \frac{N_H}{L}, P_E = \frac{N_E}{L}, P_C = \frac{N_C}{L} \qquad (4)$$

where $N_H$, $N_E$ and $N_C$ are the numbers of residues in the $\alpha$-helix, $\beta$-strand and random coil states, respectively. The next three features are the means of the probability series. In addition, the AC transform is applied to the probability series with $l_{\max}$ being 9, determined based on the RDD training set (Supplementary Fig. S1). In addition, the total number of amino acids in a protein is used as a feature as well. As a result, a total of 34 ($= 3 + 3 + 3 \times 9 + 1$) features are extracted from the PSIPRED profile.

Note that the values of $l_{\max}$ (for PSI-BLAST and PSIPRED profiles) were optimized on the RDD training set. For the sake of generality, they are also used for other datasets, though not necessarily optimal. For example on the F184 dataset, the maximum accuracy was obtained when $l_{\max}$ equals to 4 and 10 for PSI-BLAST and PSIPRED profiles, respectively (Supplementary Fig. S2). However, the difference between the accuracy with the default values (4 and 9) is very small (0.845 versus 0.84).

**Features from HHblits profile.** It was demonstrated that the alignment generated by the HMM-HMM alignment algorithm HHblits (Remmert *et al.*, 2012) is more accurate than the sequence-profile alignment algorithm PSI-BLAST. In this study, the HMM profile is generated by searching the query sequence against the database uniprot20_2015_06 using HHblits with parameters '-n 3 -

maxfilt 500000 -diff inf -id 99 -cov 60'. The dimension of HMM profile is $L$x30, in which the first 20 columns represent the match state amino acid emission frequencies, and the remaining 10 columns are seven transition frequencies and three local diversities. Similar to Ref. (Lyons *et al.*, 2015), only the first 20 columns are used in this study as the best performance was achieved with them. According to the HHsuite manual, the integers in HMM are equal to 1000 times the negative logarithm of the amino acid frequencies. Thus each element $h_{ij}$ in the HMM profile is converted to frequency by taking the following conversion:

$$h'_{ij} = 2^{-0.001\times h_{ij}}, i = 1, 2, \cdots, L; j = 1, 2, \cdots, 20 \tag{5}$$

The frequency $h'_{ij}$ is set to 0 if $h_{ij}$ is an asterisk *. A total of 400 (=20x20) features are calculated to describe the relationship between neighboring residues:

$$N(m, n) = \sum_{k=1}^{L-1} h'_{km} h'_{(k+1)n} \tag{6}$$

where $1 \le m,n \le 20$ represent the 20 kinds of amino acids. In addition, similar to the segment-based features in (Yang and Chen, 2011), the frequency profile after normalization (so that the summation of each row is one) is divided into three non-overlapping segment of equal size. For each segment, the mean of each column is computed, resulting to a total of 60 segment-based features. In total, 460 features are extracted from the HHblits profile. We did not apply the AC transform to the HHblits profile as incorporation of these features did not have significant impact on the accuracy (Supplementary Table S1), probably because they are not complementary to the already used features.

**Support Vector Machine.** SVM is one of the most popular machine learning algorithms. For the implementation of SVM, we use the LIBSVM package (https://www.csie.ntu.edu.tw/~cjlin/libsvm/). There are four basic kernel functions; 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 (Supplementary Table S2). It is defined as

$$K(\mathbf{x}, \mathbf{y}) = \exp\left(-\gamma\|\mathbf{x} - \mathbf{y}\|^2\right) \tag{7}$$

where **x** and **y** are the feature vectors of two proteins, and $\gamma$ is a kernel parameter. Another parameter for SVM training is the regularization factor $C$, which controls the control the trade-off between allowing training errors and forcing rigid margins.

The two parameters $C$ and $\gamma$ in SVM are optimized numerically based on the strategy of grid search. That is, a 2D grid with size 10 × 10 was used, where each grid point represents a combination of values for $C$ and $\gamma$. The possible values tested for $C$ and $\gamma$ were [$2^0, 2^1, \ldots, 2^9$] and [$2^0, 2^{-1}, \ldots, 2^{-9}$], respectively. In addition, the features were scaled to the range of [-1, 1] before training, using the 'svm-scale' program in the LIBSVM package. To avoid overfitting, a 10-fold cross-validation was applied on each dataset (for the DD and RDD datasets, only the training set was used). Optimal parameters were selected such that the maximum accuracy (defined later in the Section 3) was obtained. All programs were installed and run in a computer cluster with 120 CPU cores, 128GB memory and 10TB disk space.

### 2.4 An ensemble approach TA-fold for protein fold classification

In order to make full use of the advantages of both Template-based fold assignment and machine learning-based *Ab-initio* fold classification, an ensemble approach TA-fold is proposed here. Figure 1 illustrates the hierarchical architecture of TA-fold, which predicts the fold of query proteins by combining HH-fold and SVM-fold. When there are homologous templates to the query protein, the predictions by HH-fold are used. Otherwise, SVM-fold is adopted. In TA-fold, the E-value in the HHsearch output is used to decide if the templates from HH-fold are homologous enough to the query protein or not, the cutoff of which is determined by experiment subsequently.

## 3 Results and discussions

The performance of our proposed methods is measured by the overall accuracy, which is defined as the number of correctly predicted proteins divided by the total number of proteins under investigation. In addition, the accuracy for each fold and the optimal SVM parameters are presented in the Supplementary Tables S3–S7. For the DD and RDD datasets, the accuracy reported is on the independent test set. For the EDD, TG and F184 datasets, a 10-fold cross-validation is applied. For the LE dataset, a 2-fold cross-validation is used.

### 3.1 Accuracy of HH-fold

In HH-fold, the number of the top-ranked templates, i.e. the variable $N$ remains to be determined. We select the optimal value of $N$ based on 10-fold cross-validation on the RDD training set. The relationship between the variable $N$ and the overall accuracy is shown in Figure 2, where the value of $N$ varies between 1 and 10. The accuracy is the highest (0.839) when $N$ equals to 1 and decreases when $N$ gets bigger. As a result, the value of $N$ is set to 1. At this setting, the overall accuracy of HH-fold on the DD, RDD, EDD, TG and F184 datasets are 0.763, 0.887, 0.966, 0.915 and 0.906, respectively.

We want to point out that the value of $N$ was selected to be 1 according to experiment on the RDD training set. If we use the EDD, TG and F184 datasets to conduct similar experiments, we end up with the same conclusion. This is expected because the template ranked at the top of the list has the highest confidence score (evaluated by the 'Prob' or the 'E-value' in the HHsearch output). However, if we use the DD training set for optimization (this is what we did at the very beginning of this study), we got completely different result, i.e. the 'optimal' value of $N$ was 5 (Supplementary Fig. S3). This inconsistency may be caused by the errors in the DD dataset, i.e. wrong extractions of domain sequences, as detailed in

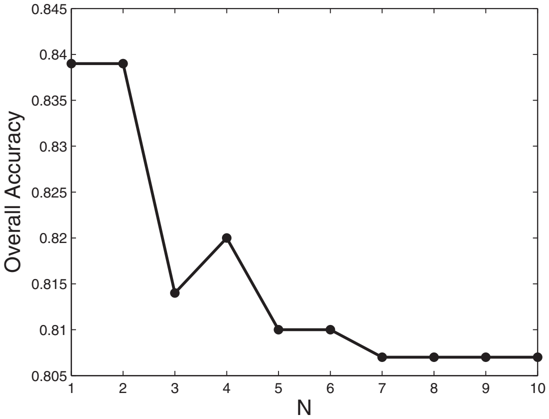

**Fig. 2.** The overall accuracy of HH-fold on the RDD training set with different numbers of top-ranked templates

the Supporting Information A of (Yang and Chen, 2011). Therefore, we suggest not using the DD dataset for training purpose in future studies as the results from this dataset may be misleading.

## 3.2 Feature contribution and accuracy of SVM-fold

A total number of 574 features have been extracted. To investigate their contribution to the overall prediction accuracy of SVM-fold, they are divided into three groups: (S1) 80 PSI-BLAST-based features; (S2) 34 PSIPRED-based features; and (S3) 460 HHblits-based features. Figure 3 shows the performance obtained with all possible combinations of feature groups on the five datasets.

When single-profile based features are used, the accuracy from the PSIPRED-based features is the lowest (0.447–0.718) while that from the HHblits-based features is the highest (0.729–0.92). The accuracy from the PSI-BLAST-based features is slightly higher ($\sim$0.05) than the PSIPRED-based features but significantly lower ($>$0.1) than the HHblits-based features. This result is striking because both PSSM and HMM represent the position-specific frequency profile and they are of the same dimension ($L$x20). A possible reason for this difference is different methods are used to extract features. Similar to the HHblits-based features, we extended the PSI-BASLST-based features to the dimension of 460, and the corresponding accuracy was improved marginally. Moreover, when adding them to our final feature set, no significant improvement was observed (data not shown). Thus for the PSI-BLAST profile, only 80 features were kept in this study.

The accuracy is improved by the combination of different feature groups. When combing the PSIPRED-based features with PSI-BLAST-based features, the accuracy increases significantly by $\sim$0.1, suggesting these two groups of features are complementary to each other. Combination of the HHblits-based features with either the PSI-BLAST- or PSIPRED-based features results to slight improvement (0.01–0.04). The highest accuracy is achieved when all features are used, i.e. 0.773, 0.9, 0.945, 0.865 and 0.84 for the DD, RDD, EDD, TG and F184 datasets, respectively.

## 3.3 Accuracy of TA-fold

In TA-fold, we need to decide the E-value threshold of using either HH-fold or SVM-fold. The RDD training set is used for this purpose. A total of 9 thresholds (0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1, 5 and 10) were tested and the highest accuracy (0.865) was achieved at the threshold of 0.05 (Supplementary Fig. S4). Similar to the case of the parameter $l_{\max}$, the E-value threshold 0.05 may not be optimal for other datasets. However, our experiments suggest that the accuracy does not change much when using dataset-specific E-value thresholds. Thus for generality, this threshold is used for all datasets. At this setting, the accuracy of TA-fold on the DD, RDD, EDD, TG and F184 datasets are 0.799, 0.932, 0.971, 0.927 and 0.913, respectively. This corresponds to the improvement of 2.8–8.7% and 0.5–5.1% over SVM-fold and HH-fold, respectively, suggesting that SVM-fold and HH-fold are complementary to each other.

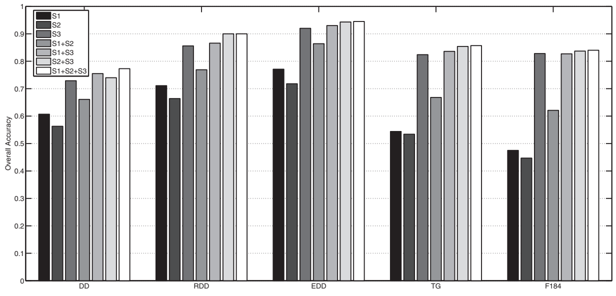

**Fig. 3.** The contribution of features to the overall accuracy of SVM-fold

The statistical significance of the accuracy difference among HH-fold, SVM-fold and TA-fold was investigated using the paired Student's $t$-tests. For each method, proteins with correctly/incorrectly predicted fold are labeled as 1/0. The $P$-values for the pairwise comparisons are listed in Table 2. We can see that for the DD and RDD datasets, the predictions by HH-fold and SVM-fold are essentially the same as judged by the $P$-values (1 and 0.73). The ensemble approach TA-fold does make statistically significant improvement over SVM-fold and HH-fold as the $P$-values are smaller than 0.05. On the large-size datasets EDD, TG and F184, HH-fold and TA-fold predictions are much better than SVM-fold. This suggests the advantage of template-based fold assignment over *ab-initio* fold classification. The difference between TA-fold and HH-fold are not significant on the EDD and TG datasets as witnessed by the respective $P$-values of 0.06 and 0.16, probably because the TA-fold predictions are dominated by HH-fold for these two datasets. However, on the largest dataset F184, TA-fold significantly outperforms HH-fold at $P$-value 0.0114 ($<$0.05), which once again suggests that HH-fold and SVM-fold is complementary to each other.

## 3.4 Comparison with machine learning-based *ab-initio* methods

To demonstrate the effectiveness of the proposed methods, we compare SVM-fold and TA-fold with machine learning-based *ab-initio* methods on four benchmark datasets, DD, RDD, EDD and TG. The results of F184 dataset are not available for other methods and this dataset is not used for comparison. Many *ab-initio* methods have been developed for protein fold classification. Five representative ones were selected based on three criteria: (i) developed recently; (ii) tested on most of the above four datasets and (iii) shown to have competitive performance. As we do not have the per-protein predictions for other methods, we are unable to perform statistical test to estimate the significance level of the accuracy difference.

As the DD dataset was divided into training and independent test sets, the accuracies reported were for the test set. However, for the compared method ACCFold (Dong *et al.*, 2009) (resp., NiRecor (Cheung *et al.*, 2016)), its accuracy was from a 2-fold (resp., 10-fold) cross-validation. It was shown that the accuracy of ACCFold would be 0.666 for the independent test set (Yang and Chen, 2011). The accuracy of NiRecor decreased from 0.812 to 0.793 when 5-fold cross-validation was applied (Cheung *et al.*, 2016). So it is unfair to compare with these two methods on the DD dataset and thus omitted.

From Table 3, we can see that SVM-fold outperforms all compared *ab-initio* methods by 0.6–8.2% on the four benchmark datasets. When we compare the ensemble approach TA-fold with the

**Table 2.** The $P$-values of the paired Student's $t$-tests on the accuracy difference between the proposed methods

| Dataset | HH-fold versus SVM-fold | TA-fold versus SVM-fold | TA-fold versus HH-fold |
|---|---|---|---|
| DD | 1 | 0.01 | 0.01 |
| RDD | 0.73 | 0.002 | 0.004 |
| EDD | $1.2391 \times 10^{-4}$ | $2.6413 \times 10^{-13}$ | 0.06 |
| TG | $1.0286 \times 10^{-9}$ | $5.8514 \times 10^{-20}$ | 0.16 |
| F184 | $3.8401 \times 10^{-52}$ | $1.7956 \times 10^{-98}$ | 0.0114 |

methods TAXFOLD, PFPA and HMMFold on the DD dataset, TA-fold makes improvement of 5.4%, 8.6% and 11.7%, respectively. For the RDD dataset, TA-fold achieved an accuracy of 0.932, which is 26.3% and 12% higher than ACCFold and TAXFOLD, respectively. For the extended DD dataset, EDD, the accuracy of all methods except ACCFold is higher than 0.9, suggesting that it is very accurate to predict folds for proteins in the 27 SCOP folds. On the EDD dataset, TA-fold achieves an accuracy of 0.971, 3.5% higher than the second best method HMMFold. When the number of SCOP folds is increased to 30 in the TG dataset, the accuracy of all methods decreases. TA-fold is the first method to achieve >0.9 accuracy on this dataset, which may be attributed mainly to the template-based assignment algorithm HH-fold (see Table 2).

### 3.5 Comparison with template-based methods

As TA-fold combines template-based method with *ab-initio* method, it is indispensable to compare it with template-based threading algorithms. Three state-of-the-art threading methods are selected for comparison: HHpred (Soding *et al.*, 2005), SPARKS-X (Yang *et al.*, 2011) and FFAS-3D (Xu *et al.*, 2014). A 2-fold cross-validation is adopted to evaluate the accuracy on the LE dataset because it has been applied to the same dataset by previous studies (Dong *et al.*, 2009; Lyons *et al.*, 2015; Yang and Chen, 2011). Nevertheless, a 5-fold cross-validation was also tested for TA-fold and similar accuracy was obtained (i.e. 0.822, 0.76 and 0.576 at the family, superfamily and fold levels, respectively), which suggests that there is no overfitting with the 2-fold cross-validation. For each level (family, superfamily or fold), the whole dataset is divided into two subsets with the same procedure used in previous studies (Dong *et al.*, 2009; Lyons *et al.*, 2015; Yang and Chen, 2011). The division at the superfamily/fold level was made in such a way that the training and test proteins come from different families/superfamilies. In addition, for each subset there should be at least one protein in each category so that SVM-fold can be trained. For more details about the division, one may refer to (Dong *et al.*, 2009).

The test results are listed in Table 4. We can see that the accuracy for HH-fold is comparable to the state-of-the-art threading methods at the family and fold levels. At the superfamily level, HH-fold outperforms other threading methods by 11–26%. Note that both HHpred and HH-fold use HHsearch for template identification but with different results at the superfamily and fold levels. This is mainly due to the fact that preprocessing was performed on the LE dataset here.

As an ensemble approach TA-fold, it has the advantage of both template-based method and *ab-initio* method. It is thus anticipated that TA-fold outperforms other threading methods at each level of the LE dataset. For example, TA-fold achieves an accuracy of 0.539 at the fold level, which is 19.2% higher than SPARKS-X, one of the most popular threading methods. Though TA-fold performs well in this test, it is necessary to point out that TA-fold does not provide an alignment, nor cover as many folds as the other threading methods.

**Table 3.** The accuracy of SVM-fold and TA-fold and other methods for protein fold classification

| Method (Reference) | Dataset | | | |
|---|---|---|---|---|
| | DD | RDD | EDD | TG |
| ACCFold (Dong et al., 2009) | 0.701[a] | 0.738[c] | 0.859 | 0.664 |
| TAXFOLD (Yang and Chen, 2011) | 0.715 | 0.832 | 0.9 | NA |
| PFPA (Wei et al., 2015) | 0.736 | NA | 0.926 | NA |
| HMMFold (Lyons et al., 2015) | 0.758 | NA | 0.938 | 0.86 |
| NiRecor (Cheung et al., 2016) | **0.812**[b] | NA | 0.917 | 0.846 |
| SVM-fold (This Article) | 0.773 | 0.9 | 0.945 | 0.865 |
| TA-fold (This Article) | 0.799 | **0.932** | **0.971** | **0.927** |

For the DD and RDD datasets, the accuracy reported is for the independent dataset. For the EDD and TG datasets, the accuracy was obtained using 10-fold cross-validation. The best results are highlighted in bold type.

[a] from 2-fold cross-validation;

[b] from 10-fold cross-validation;

[c] from Ref. (Yang and Chen, 2011).

**Table 4.** Comparison with threading methods on the LE dataset

| Method | Family<br>(555/176) | Superfamily<br>(434/86) | Fold<br>(321/38) |
|---|---|---|---|
| HHpred | 0.829 | 0.588 | 0.252 |
| FFAS-3D | 0.849 | 0.666 | 0.358 |
| SPARKS-X | 0.841 | 0.59 | 0.452 |
| HH-fold | 0.845 | 0.74 | 0.421 |
| TA-fold | **0.852** | **0.742** | **0.539** |

The accuracies for other methods are taken from (Xu et al., 2014). The numbers in parentheses are the number of sequences/categories. The best results are highlighted in bold type.

### 3.6 Application of TA-fold to structural class prediction

As mentioned earlier, the first level of the SCOP hierarchy is structural class, in which four main classes exist: $\alpha$, $\beta$, $\alpha/\beta$ and $\alpha+\beta$. Many methods have been proposed to predict structural class from amino acid sequence over the past two decades (Chou and Zhang, 1995; Mizianty and Kurgan, 2009; Yang *et al.*, 2010; Yang *et al.*, 2009). We applied TA-fold to the structural class prediction using five benchmark datasets, where the native class information was taken from the SCOP database. For each dataset, the proteins that do not belong to any of the four classes were removed before running TA-fold.

The prediction accuracy of TA-fold is listed in Table 5. We can see the overall accuracies for the RDD, EDD, TG and F184 datasets are >0.95. The accuracy on DD is relatively lower probably due to the errors in this dataset mentioned in the Section 3.1. The accuracy

**Table 5.** The comparison of TA-fold with RKS_PPSC for structural class prediction

| Dataset | Method/# | Accuracy | | | | |
|---|---|---|---|---|---|---|
| | | $\alpha$ | $\beta$ | $\alpha/\beta$ | $\alpha+\beta$ | Overall |
| | #Samples | 61 | 117 | 143 | 35 | 356 |
| DD | RKS_PPSC | 0.836 | 0.803 | 0.657 | 0.714 | 0.75 |
| | TA-fold | 0.934 | 0.966 | 0.832 | 0.543 | 0.862 |
| | #Samples | 60 | 117 | 142 | 34 | 353 |
| RDD | RKS_PPSC | 0.867 | 0.88 | 0.859 | 0.853 | 0.867 |
| | TA-fold | 1 | 0.974 | 0.986 | 0.686 | 0.955 |
| | #Samples | 556 | 967 | 1311 | 460 | 3294 |
| EDD | RKS_PPSC | 0.883 | 0.895 | 0.874 | 0.822 | 0.874 |
| | TA-fold | 0.996 | 0.993 | 0.995 | 0.97 | 0.991 |
| | #Samples | 252 | 478 | 589 | 185 | 1504 |
| TG | RKS_PPSC | 0.909 | 0.845 | 0.883 | 0.8 | 0.865 |
| | TA-fold | 0.976 | 0.983 | 0.992 | 0.876 | 0.972 |
| | #Samples | 1185 | 1471 | 1828 | 1423 | 5907 |
| F184 | RKS_PPSC | 0.915 | 0.829 | 0.851 | 0.73 | 0.829 |
| | TA-fold | 0.986 | 0.987 | 0.976 | 0.954 | 0.975 |

for proteins in the $\alpha+\beta$ class is lower than other classes, especially for the DD and RDD datasets probably because the number of proteins in this class is relatively smaller than others. We looked into the predictions and found that for these two datasets, some proteins in the $\alpha+\beta$ class were wrongly predicted into the $\alpha/\beta$ class. This may be explained by the fact that there are some overlap between the $\alpha+\beta$ and $\alpha/\beta$ proteins because both classes contain structural motifs of $\alpha$-helices and $\beta$-strands. We note that even human experts may differ in categorizing proteins in these two classes. For example, the CATH database merges them into a single $\alpha\beta$ class (Sillitoe *et al.*, 2015).

TA-fold is compared with RKS_PPSC (Yang *et al.*, 2010), one of the best programs for structural class prediction using predicted secondary structure. Note that there are some other good structural class prediction programs, such as the iFC$^2$ (Chen *et al.*, 2011) and MODAS (Mizianty and Kurgan, 2009). However, RKS_PPSC was selected for the sake of easy comparisons as we have the source codes and executables to run it locally for all datasets. The results of RKS_PPSC are listed in Table 5 as well. We can see that the overall accuracy of TA-fold is significantly higher than RKS_PPSC on all datasets, though the accuracy for the $\alpha+\beta$ class in the DD and RDD datasets is higher for RKS_PPSC. This demonstrates that TA-fold is also effective for structural class prediction.

### 3.7 Online TA-fold server

To facilitate the use of TA-fold, we have setup a web server to implement the TA-fold algorithm, which is freely available at http://yanglab.nankai.edu.cn/TA-fold. The only input information is the amino acid sequence of the query protein to be predicted. A job ID will be assigned to each submission. After the job is finished, a notification email will be sent to the users for accessing the prediction results. In general, the prediction can be completed within 15 min. The output of the server includes the predicted fold together with a confidence score (C-score), a summary of the submitted sequence, predicted secondary structure and sequence profiles. The C-score is in the range of [0, 1], obtained from the probability outputs of SVM and HHsearch. In general, a higher C-score indicates a more reliable prediction. Based on the analysis of the predictions on RDD dataset, a recommended C-score cutoff for trusting a prediction is 0.22, at which 96% proteins are predicted with an accuracy of 0.96 (Supplementary Fig. S5).

Note that the server takes each submission as a single-domain protein. Thus when the query protein contains multiple domains, it is advisable to split the protein into domains using other domain prediction software and submit each domain sequence to the server. This may be made automated by developing in-house domain parsing algorithm in future.

We estimate the possibility that the fold of a protein can be classified with the server as follows. The maximum number of folds that TA-fold could deal with is 184 (in the F184 dataset). As mentioned before, there are 8679 sequences from 1222 folds at 25% sequence identity cutoff in the SCOPe database. After filtering, 6451 proteins from 184 folds were kept in the F184 dataset. Therefore, though the proportion of folds considered is small (15%=184/1222), the possibility of a query protein being a TA-fold target is high (74%=6451/8679). Anyway, SVM-fold is not applicable for the remaining 26% proteins that do not belong to any of the 184 folds. The SVM-fold models remain to be re-built in future when there are enough samples for training.

To partially solve the above limitation, we incorporated proteins from other folds in the SCOP database into the HH-fold database. Currently, the maximum number of folds considered is 1193 (list available at http://yanglab.nankai.edu.cn/TA-fold/1193_name.txt). When the query proteins do not belong to any of the 184-folds and the confidence scores of the predictions are anticipated to be lower (<0.22), users are encouraged to check the prediction results in the 1193 folds.

## 4 Conclusions

Accurate classification of protein fold is essential for protein structure prediction. We have developed two complementary algorithms, HH-fold for template-based fold assignment, and SVM-fold for support vector machine-based *ab-initio* fold classification using features extracted from three complementary sequence profiles. These two algorithms are then combined to make accurate and robust fold type prediction, resulting to the ensemble approach TA-fold.

The proposed methods were assessed and compared with both machine learning-based *ab-initio* methods and template-based threading methods on six benchmark datasets. Experiments show that TA-fold consistently outperforms both *ab-initio* and threading methods. TA-fold was successfully applied to the problem of protein structural class prediction with accuracy of >0.95 for datasets of updated class information. We attribute the success of TA-fold to three factors: (1) template-based fold assignment; (2) *ab-initio* classification using features from three complementary sequence profiles that contain rich evolution information of query protein; and (3) integration of (1) and (2).

## Funding

The project is supported in part by National Natural Science Foundation of China (11501306, 11501407), China National 863 High-Tech Program (2015AA020101) and the Thousand Youth Talents Plan of China.

*Conflict of Interest*: none declared.

## References

Altschul,S.F. *et al.* (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res.*, **25**, 3389–3402.

Chen,K. and Kurgan,L. (2007) PFRES: protein fold classification by using evolutionary information and predicted secondary structure. *Bioinformatics*, **23**, 2843–2850.

Chen,K. *et al.* (2011) iFC(2): an integrated web-server for improved prediction of protein structural class, fold type, and secondary structure content. *Amino Acids*, **40**, 963–973.

Cheung,N.J. *et al.* (2016) Protein folds recognized by an intelligent predictor based-on evolutionary and structural information. *J. Comput. Chem.*, **37**, 426–436.

Chou,K.C. and Zhang,C.T. (1995) Prediction of protein structural classes. *Crit. Rev. Biochem. Mol. Biol.*, **30**, 275–349.

Damoulas,T. and Girolami,M.A. (2008) Probabilistic multi-class multi-kernel learning: on protein fold recognition and remote homology detection. *Bioinformatics*, **24**, 1264–1270.

Deschavanne,P. and Tuffery,P. (2009) Enhanced protein fold recognition using a structural alphabet. *Proteins*, **76**, 129–137.

Ding,C.H. and Dubchak,I. (2001) Multi-class protein fold recognition using support vector machines and neural networks. *Bioinformatics*, **17**, 349–358.

Dong,Q. *et al.* (2009) A new taxonomy-based protein fold recognition approach based on autocross-covariance transformation. *Bioinformatics*, **25**, 2655–2662.

Fox,N.K. *et al.* (2014) SCOPe: Structural Classification of Proteins–extended, integrating SCOP and ASTRAL data and classification of new structures. *Nucleic Acids Res.*, **42**, D304–D309.

Guo,X. and Gao,X. (2008) A novel hierarchical ensemble classifier for protein fold recognition. *Protein Eng. Des. Select. PEDS*, **21**, 659–664.

Hadley,C. and Jones,D.T. (1999) A systematic comparison of protein structure classifications: SCOP, CATH and FSSP. *Structure*, **7**, 1099–1112.

Huang,C.D. *et al.* (2003) Hierarchical learning architecture with automatic feature selection for multiclass protein fold classification. *IEEE Trans. Nanobiosci.*, **2**, 221–232.

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

Jones,D.T. *et al.* (1992) A new approach to protein fold recognition. *Nature*, **358**, 86–89.

Lindahl,E. and Elofsson,A. (2000) Identification of related proteins on family, superfamily and fold level. *J. Mol. Biol.*, **295**, 613–625.

Liu,X. *et al.* (2004) The number of protein folds and their distribution over families in nature. *Proteins*, **54**, 491–499.

Lyons,J. *et al.* (2015) Advancing the Accuracy of Protein Fold Recognition by Utilizing Profiles From Hidden Markov Models. *IEEE Trans. Nanobiosci.*, **14**, 761–772.

Mizianty,M.J. and Kurgan,L. (2009) Modular prediction of protein structural classes from sequences of twilight-zone identity with predicting sequences. *BMC Bioinformatics*, **10**, 414.

Rangwala,H. and Karypis,G. (2005) Profile-based direct kernels for remote homology detection and fold recognition. *Bioinformatics*, **21**, 4239–4247.

Remmert,M. *et al.* (2012) HHblits: lightning-fast iterative protein sequence searching by HMM-HMM alignment. *Nat. Methods*, **9**, 173–175.

Shamim,M.T. *et al.* (2007) Support Vector Machine-based classification of protein folds using the structural properties of amino acid residues and amino acid residue pairs. *Bioinformatics*, **23**, 3320–3327.

Sharma,A. *et al.* (2013) A feature extraction technique using bi-gram probabilities of position specific scoring matrix for protein fold recognition. *J. Theor. Biol.*, **320**, 41–46.

Shen,H.B. and Chou,K.C. (2006) Ensemble classifier for protein fold pattern recognition. *Bioinformatics*, **22**, 1717–1722.

Shen,H.B. and Chou,K.C. (2009) Predicting protein fold pattern with functional domain and sequential evolution information. *J. Theor. Biol.*, **256**, 441–446.

Sillitoe,I. *et al.* (2015) CATH: comprehensive structural and functional annotations for genome sequences. *Nucleic Acids Res.*, **43**, D376–3D381.

Soding,J. (2005) Protein homology detection by HMM-HMM comparison. *Bioinformatics*, **21**, 951–960.

Soding,J. *et al.* (2005) The HHpred interactive server for protein homology detection and structure prediction. *Nucleic Acids Res.*, **33**, W244–W248.

Taguchi,Y.H. and Gromiha,M.M. (2007) Application of amino acid occurrence for discriminating different folding types of globular proteins. *BMC Bioinformatics*, **8**, 404.

Wei,L. *et al.* (2015) Enhanced protein fold prediction method through a novel feature extraction technique. *IEEE Trans. Nanobiosci.*, **14**, 649–659.

Wold,S. *et al.* (1993) DNA and peptide sequences and chemical processes multivariately modelled by principal component analysis and partial least-squares projections to latent structures. *Anal. Chim. Acta*, **277**, 239–253.

Xu,D. *et al.* (2014) FFAS-3D: improving fold recognition by including optimized structural features and template re-ranking. *Bioinformatics*, **30**, 660–667.

Yang,J. *et al.* (2015) The I-TASSER Suite: protein structure and function prediction. *Nat. Methods*, **12**, 7–8.

Yang,J.Y. and Chen,X. (2011) Improving taxonomy-based protein fold recognition by using global and local features. *Proteins*, **79**, 2053–2064.

Yang,J.Y. *et al.* (2010) Prediction of protein structural classes for low-homology sequences based on predicted secondary structure. *BMC Bioinformatics*, **11**, S9.

Yang,J.Y. *et al.* (2009) Prediction of protein structural classes by recurrence quantification analysis based on chaos game representation. *J. Theor. Biol.*, **257**, 618–626.

Yang,Y. *et al.* (2011) Improving protein fold recognition and template-based modeling by employing probabilistic-based matching between predicted one-dimensional structural properties of query and corresponding native properties of templates. *Bioinformatics*, **27**, 2076–2082.

Zakeri,P. *et al.* (2014) Protein fold recognition using geometric kernel data fusion. *Bioinformatics*, **30**, 1850–1857.

Zhang,C., and DeLisi,C. (1998) Estimating the number of protein folds. *J. Mol. Biol.*, **284**, 1301–1305.