Structural bioinformatics

A new taxonomy-based protein fold recognition approach based on autocross-covariance transformation

Qiwen Dong1,2, Shuigeng Zhou1,2,∗ and Jihong Guan3

1Shanghai Key Lab of Intelligent Information Processing, 2School of Computer Science, Fudan University and 3Department of Computer Science and Technology, Tongji University, Shanghai, China

Received on March 11, 2009; revised on August 11, 2009; accepted on August 13, 2009

ABSTRACT

Motivation: Fold recognition is an important step in protein structure and function prediction. Traditional sequence comparison methods fail to identify reliable homologies with low sequence identity, while the taxonomic methods are effective alternatives, but their prediction accuracies are around 70%, which are still relatively low for practical usage.

Results: In this study, a simple and powerful method is presented for taxonomic fold recognition, which combines support vector machine (SVM) with autocross-covariance (ACC) transformation. The evolutionary information represented in the form of position-specific score matrices is converted into a series of fixed-length vectors by ACC transformation and these vectors are then input to a SVM classifier for fold recognition. The sequence-order effect can be effectively captured by this scheme. Experiments are performed on the widely used D-B dataset and the corresponding extended dataset, respectively. The proposed method, called ACCFold, gets an overall accuracy of 70.1% on the D-B dataset, which is higher than major existing taxonomic methods by 2–14%. Furthermore, the method achieves an overall accuracy of 87.6% on the extended dataset, which surpasses major existing taxonomic methods by 9–17%. Additionally, our method obtains an overall accuracy of 80.9% for 86-folds and 77.2% for 199-folds. These results demonstrate that the ACCFold method provides the state-of-the-art performance for taxonomic fold recognition.

Availability: The source code for ACC transformation is freely available at http://www.iipl.fudan.edu.cn/demo/accpkg.html.

1 INTRODUCTION

The prediction of 3D structure of a protein from its amino acid sequence is one of the most important tasks in computational biology. A solution to this enigma is ever more essential considering that the number of completely sequenced genomes rapidly increases (Kouranov et al., 2006), while the proteins with known structures are very limited (Kouranov et al., 2006). Theoretical prediction methods provide possible ways to fill in this gap. During the last decades, many template-based and template-free methods have been presented (Moule et al., 2007). A protein structure can be accurately modeled when the homologous templates are identified. However, the detection of homologies with low sequence identity remains a challenging problem.

Fold recognition has been proven an effective method for protein structure prediction. The general sequence comparison methods such as sequence–sequence alignment (Altschul et al., 1997), sequence–profile alignment (Gough et al., 2001) and the profile–profile alignment (Wang and Dunbrack, 2004) fail to recognize homologous templates when the similarities between the query sequence and the templates are low. An alternative is to thread the query sequence onto the template structures (Wu and Zhang, 2007). Unfortunately, these methods, although widely used, are not able to achieve satisfactory results at the fold level (Cheng and Baldi, 2006). Recently, the taxonomic methods have been developed for protein fold recognition, which classify a sequence into one of the fold classes according to the SCOP architecture (Andreeva et al., 2004). The basic idea is that the fold types in proteins are finite. The total number of folds in globular, water-soluble proteins is estimated at about 1000 (Wolf et al., 2000). Some variant methods have also been proposed for taxonomic fold recognition. Ding and Dubchak (2001) used the support vector machine (SVM) and neural network for multi-class protein fold recognition and achieved an accuracy of 56% on a dataset containing 27 folds. Here, the accuracy refers to the percentage of proteins whose folds have been correctly identified. Other researchers tried to improve prediction performance by either incorporating new features or developing novel algorithms for multi-class classification (Bindewald et al., 2003; Le et al., 2005; Rangwala and Karypis, 2006). Shen and Chou (2006) presented an ensemble classifier called PFP-Pred, which improves the accuracy to 62% on the same dataset used by Ding and Dubchak (2001). Other ensemble classifiers have also been developed, such as the genetic algorithm optimized ensemble classifier (Guo and Gao, 2008), the probabilistic multi-class multi-kernel classifier (Damoulas and Girolami, 2008), the hybrid machine learning classifier (Melvin et al., 2008). Besides the basic sequences or physical–chemical propensities of amino acids as fold discriminatory features, the informations such as secondary structural state and solvent accessibility state can be used to further improve the accuracy to 79% (Chen and Kurgan, 2007; Shamim et al., 2007). The most promising result was achieved by Deschavanne and Tuffery (2009), who employed the hidden Markov
To evaluate the proposed method and compare it with existing methods, D-B dataset and the extended D-B dataset are listed in Table 1.

sequences. The fold names and the number of proteins in each fold of the D-B dataset are extracted. The resulting dataset contains 3202 sequences have <40% identity. The proteins that belong to any of the 27 folds of the D-B dataset belong to 27 different SCOP folds representing all major structural classes: all α, all β, α/β, α+β and small proteins.

Structure alphabet as additional feature and got an accuracy of 78% for 27 folds (extended D-B dataset) and 68% for 60 folds.

Machine learning-based techniques such as SVM and neural network require fixed-length vectors for training. However, protein sequences often have different lengths. Since protein sequences have univariate direction from beginning to end, which is analogous to time sequences of process data, thus time series models can be used. Here, the autocovariance (ACC) transformation is introduced to transform protein sequences into fixed-length vectors. As a statistical tool for analyzing sequences of vectors developed by Wold et al. (1993), ACC has been successfully used for protein family classification (Guo et al., 2006; Lapinsh et al., 2002) and protein interaction prediction (Guo et al., 2008). Since each residue has many physical-chemical properties, such as hydrophobicity and hydropilocicity, normalized van der Waals volume, polarity, polarizability, sequence profile, etc., a sequence can be represented as a numeric matrix. ACC can measure the correlation of two properties (or the same property) along the protein sequence and transforms the matrix into a fixed-length vector. Note that the PFP-Pred method implicitly used the autocovariance (AC) transformation for hydrophobicity and hydrophilicity properties (Shen and Chou, 2006).

In this study, a novel taxonomic method that combines SVM with ACC is presented for protein fold recognition. Previous successful applications of PSI-BLAST profile show that evolutionary information is more informative than query sequence itself, so the position-specific score matrix (PSSM) is transformed into a fixed-length vector by ACC. This vector is then input to a SVM classifier to predict folds. Experiments are performed on the widely used D-B dataset established by Ding and Dubchak (2001) and the extended D-B dataset, respectively. The proposed method, called ACCFold, gets an overall accuracy of 70.1% on the D-B dataset and 87.6% on the extended D-B dataset, which outperforms major existing taxonomic methods. Furthermore, the proposed method can be applied to large-scale folds without significant performance degradation. Performance comparison with the alignment-based threading methods is also presented.

2 MATERIALS AND METHODS

2.1 Datasets

To evaluate the proposed method and compare it with existing methods, five datasets are used here: the D-B dataset (Ding and Dubchak, 2001), the extended D-B dataset, the prD dataset and the extended prD dataset and the Lindahl dataset (Lindahl and Elofsson, 2000).

The D-B dataset (available at http://ranger.uta.edu/~chqding/protein/) has been widely used by many existing studies (Chen and Kurgan, 2007; Guo and Gao, 2008; Shamim et al., 2007; Shen and Chou, 2006). It contains 311 proteins for training and 383 proteins for test. According to Ding and Dubchak (2001), any two proteins in the training set share <40% identity and each fold has at least seven members. Each protein in the test set has not more than 35% identity to any protein in the training set. The proteins in this dataset belong to 27 different SCOP folders representing all major structural classes: all α, all β, α/β, α+β and small proteins.

The extended D-B dataset is generated by populating additional protein samples. It is based on the Astral SCOP 1.73 release, in which any two sequences have <40% identity. The proteins that belong to any of the 27 folds of the D-B dataset are extracted. The resulting dataset contains 3202 sequences. The fold names and the number of proteins in each fold of the D-B dataset and the extended D-B dataset are listed in Table 1.

Table 1. Summary of the 27 folds in the two datasets

<table>
<thead>
<tr>
<th>Index</th>
<th>Fold</th>
<th>D-B dataset</th>
<th>Extended D-B dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>N_{train}</td>
<td>N_{test}</td>
</tr>
<tr>
<td></td>
<td></td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>1</td>
<td>Globin-like</td>
<td>13</td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td>Cytochrome c</td>
<td>7</td>
<td>9</td>
</tr>
<tr>
<td>3</td>
<td>DNA/RNA-binding</td>
<td>12</td>
<td>20</td>
</tr>
<tr>
<td>4</td>
<td>3-helical bundle</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>5</td>
<td>4-helical up-and-down bundle</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>6</td>
<td>EF-hand-like</td>
<td>6</td>
<td>9</td>
</tr>
<tr>
<td>7</td>
<td>Immunoglobulin-like</td>
<td>30</td>
<td>44</td>
</tr>
<tr>
<td>8</td>
<td>Cupredoxin-like</td>
<td>9</td>
<td>12</td>
</tr>
<tr>
<td>9</td>
<td>Nucleoplasmin-like/VP (viral coat and capsid proteins)</td>
<td>16</td>
<td>13</td>
</tr>
<tr>
<td>10</td>
<td>CtnA-like lectins/glucanases</td>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>11</td>
<td>SH3-like barrel</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>12</td>
<td>OB-fold</td>
<td>13</td>
<td>19</td>
</tr>
<tr>
<td>13</td>
<td>Beta-Trefoil</td>
<td>8</td>
<td>4</td>
</tr>
<tr>
<td>14</td>
<td>Trypsin-like serine proteases</td>
<td>9</td>
<td>4</td>
</tr>
<tr>
<td>15</td>
<td>Lipocalins</td>
<td>9</td>
<td>7</td>
</tr>
<tr>
<td>16</td>
<td>TIM beta/alpha-barrel</td>
<td>29</td>
<td>48</td>
</tr>
<tr>
<td>17</td>
<td>FAD/NADP-binding domain</td>
<td>11</td>
<td>12</td>
</tr>
<tr>
<td>18</td>
<td>Flavodoxin-like</td>
<td>11</td>
<td>13</td>
</tr>
<tr>
<td>19</td>
<td>NADP-binding Rossmann</td>
<td>13</td>
<td>27</td>
</tr>
<tr>
<td>20</td>
<td>F-loop containing NTH</td>
<td>10</td>
<td>12</td>
</tr>
<tr>
<td>21</td>
<td>Thioredoxin-fold</td>
<td>9</td>
<td>8</td>
</tr>
<tr>
<td>22</td>
<td>Ribonuclease-H-like motif</td>
<td>10</td>
<td>12</td>
</tr>
<tr>
<td>23</td>
<td>alpha-beta-Hydrolases</td>
<td>11</td>
<td>7</td>
</tr>
<tr>
<td>24</td>
<td>Periplasmic binding protein-like</td>
<td>11</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>alpha+beta</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>25</td>
<td>beta-grap (ubiquitin-like)</td>
<td>13</td>
<td>27</td>
</tr>
<tr>
<td>26</td>
<td>Ferredoxin-like</td>
<td>13</td>
<td>27</td>
</tr>
<tr>
<td>27</td>
<td>Knottins (small inhibitors, toxins, lectins)</td>
<td>13</td>
<td>27</td>
</tr>
</tbody>
</table>

The D-B dataset and the extended D-B dataset contain only 27 folds, which cover a small part of the total folds (1086 folds in SCOP 1.73 database). To evaluate their applicabilities, the taxonomic fold recognition methods are often tested on new datasets containing larger numbers of folds. Again, the non-redundant Astral SCOP 1.73 subset is used. Since a small numbers of samples cannot effectively cover the sample space, we select the folds that contains at least N=1 (a predefined parameter) members. Two new datasets are derived: F86 and F199. The F86 dataset contains 86 folds and 5671 sequences, in which each fold has at least 25 members. The F199 dataset contains 199 folds and 4743 sequences, in which each fold has at least 10 members. The extended D-B dataset, the F86 and F199 dataset, are available at the Supplementary Material.
2.2 ACC transformation

Each residue has many physical-chemical properties as mentioned above, so a protein sequence can be viewed as a time sequence of the corresponding properties. In this study, only the evolutionary information represented in the form of PSSM is selected as the feature, since it alone can achieve promising results. The profile of each sequence is generated by running PSI-BLAST (Altschul et al., 1997) against the NCBI’s NR dataset (ftp://ftp.ncbi.nih.gov/blast/db/nr) with parameters (-i 3, -e 0.01). The element $S_{ij}$ in the matrix reflects the probability of amino acid $i$ occurring at the position $j$.

ACC, as one of the multivariate modeling tools, can transform the PSSMs of different lengths into fixed-length vectors by measuring the correlation between any two properties. ACC results in two kinds of variables: AC between the same property, and cross-covariance (CC) between two different properties. The AC variable measures the correlation of the same property between the same property, and cross-covariance (CC) between two different properties. The AC variable measures the correlation of the same property between two residues separated by a distance of $lg$ along the sequence, which can be calculated as:

$$AC(i, lg) = \sum_{j=1}^{L} (S_{ij} - \bar{S}_i)(S_{ij+lg} - \bar{S}_j)/(L - lg)$$

(1)

where $i$ is one of the residues, $L$ is the length of the protein sequence, $S_{ij}$ is the PSSM score of amino acid $i$ at position $j$, $\bar{S}_i$ is the average score for amino acid $i$ along the whole sequence:

$$\bar{S}_i = \sum_{j=1}^{L} S_{ij}/L$$

(2)

In such a way, the number of AC variables can be calculated as $20 \times LG$, where LG is the maximum of $lg$ ($lg = 1, 2, ..., LG$).

The CC variable measures the correlation of two different properties between two residues separated by $lg$ along the sequence, which can be calculated by:

$$CC(1, 2, lg) = \sum_{j=1}^{L} (S_{1j} - \bar{S}_1)(S_{2j+lg} - \bar{S}_2)/(L - lg)$$

(3)

where $1, 2$ are two different amino acids and $\bar{S}_i$ ($\bar{S}_j$) is the average score for amino acid $1$ ($2$) along the sequence. Since the CC variables are not symmetric, the total number of CC variables is $380 \times LG$.

In view of ACC transformation, the sequence effect can be indirectly and partially, but quite effectively reflected. Additionally, the ACC transformation is independent of sequence alignments.

In this study, each protein sequence is represented as a vector of either AC variable or ACC variable that is a combination of AC and CC.

2.3 Support vector machine

SVM is a class of supervised learning algorithms first introduced by Vapnik (1998). Given a set of labeled training vectors (positive and negative input examples), SVM can learn a linear decision boundary to discriminate the two classes. The result is a linear classification rule that can be used to classify new test examples. When the examples are linearly non-separable, the kernel function can be used to map the examples to a high-order feature space in which the optimal decision boundary can be found. SVM has a strong theoretical foundation of statistical learning and has exhibited excellent performance in practice.

In this study, the LIBSVM package (Chang and Lin, 2009) is used as an implementation of SVM. The radial basis function (RBF) is taken as the kernel function, which is defined as:

$$K(X_i, X_j) = \exp(-\gamma \|X_i - X_j\|^2)$$

(4)

The values of $\gamma$ and regularization parameter $C$ are optimized on the datasets by cross-validation. For the multi-state classification, the one-versus-one strategy is used.

2.4 Performance metrics

The performance of various methods is evaluated by the overall accuracy, which is the most commonly used metric for assessing the global performance of a multi-class problem. The overall accuracy ($Q$) is defined as the ratio of correctly predicted samples to all tested samples:

$$Q = \frac{CN}{N} \times 100$$

(5)

where $CN$ is the number of samples whose folds have been correctly predicted and $N$ is the total number of samples in the test dataset.

Other two measures are used to assess the performance of individual class: sensitivity ($Sn$) and specificity ($Sp$):

$$Sn = \frac{TP \times 100}{TP + FN}$$

(6)

$$Sp = \frac{TP \times 100}{TP + FP}$$

(7)

where $TP$, $FN$, and $FP$ are the number of true positives, false negatives and false positives, respectively.

The sensitivity measures the accuracy for each class, which is the same as the $i$, defined in other studies (Ding and Dubchak, 2001). The overall accuracy can be calculated as the weighted average of the sensitivities of all classes.

3 RESULTS AND DISCUSSIONS

To evaluate the proposed method (ACCfold), we first check the impact of parameter $LG$ on the performance of ACCfold, then conduct an extensive performance comparison between ACCfold and major existing taxonomy-based methods, including LA kernel (Saigo et al., 2004), SWPSSM kernel (Rangwala and Karypis, 2005) and Shamim method (Shamim et al., 2007) over D-B dataset and the extended D-B dataset, F86 and F199 datasets, and finally we compare ACCFold (including some other taxonomy-based methods) with typical threading methods.

For the D-B dataset, a 2-fold cross-validation is performed, where the original training and test sets are used as the sample partitions. For the extended D-B dataset, a 5-fold cross-validation is performed, where the samples in each fold are first randomly divided into five groups, and from each of the 27 folds, one group is randomly collected, and then the five selected groups are merged into a larger group. Thus finally, five large groups are generated for cross-validation. In the cross-validation experiments, the average performance across all trials is reported. The F86 and F199 datasets are used to test the prediction capability of the proposed method in the cases of large numbers of folds.

The LA kernel and the SWPSSM kernel have been successfully used for binary classification of remote homology detection. Here, we extend them to multi-class classification for fold recognition. The Shamim method is one of the best fold recognition methods so far. All these methods are in-house implemented and tested on the same datasets to give an unbiased comparison with the ACCFold method.
ACC are 8 and 10, respectively, whereas, on the extended dataset, ACCFold_ACC, respectively. The value of LG can impact the AC variable gets an overall accuracy of 68.6%. When the ACC the Shamim method gets an overall accuracy of 59.1%, which is slightly lower than the accuracy reported in the literature (59.5%).

The proposed method can use AC or ACC variables, and the average accuracy is 68.1% with SD of 1.2%. So the method may be largely dependent on the weight parameters of different kernels. The multi-kernel method is based on a complex Bayesian hierarchical framework that combines multiple kernels and its best result is 70%, obtained by 20 randomly initialized trials. The SWPSSM kernel achieves attractive performance with an overall accuracy of 67.8%, which uses the PSSM to compute the kernel. Such a result indicates that evolutionary information is more informative than the sequence itself.

3.2 Performance comparison with existing taxonomy-based methods

3.2.1 Results on the D-B dataset The performance comparison with three other taxonomy-based methods by 2-fold cross-validation is shown in Table 2. The detailed results are given in the Supplementary Material. As can be seen, the ACCFold method with AC variable gets an overall accuracy of 68.6%. When the ACC variable is used, the accuracy further increases to 70.1%, which stands for the state-of-the-art performance over this dataset. The folds in the all α and all β classes are easy to be predicted, while the folds in the other three classes are relatively difficult to be discriminated. Most of the folds in the last three classes achieve high specificities but low sensitivities, which indicates that these folds are under predicted, that is, many samples are predicted to be other folds. To the best of our knowledge, except the method (referred to as multi-kernel method here) developed by Damoulas and Girolami (2008), all the other existing methods reported their accuracies not more than 70% over the D-B dataset (Chen and Kurgan, 2007; Ding and Dubchak, 2001; Guo and Gao, 2008; Shen and Chou, 2006). The multi-kernel method is based on a complex Bayesian hierarchical framework that combines multiple kernels and its best result is 70%, obtained by 20 randomly initialized trials. The average accuracy is 68.1% with SD of 1.2%. So the method may be largely dependent on the weight parameters of different kernels. The Shamim method gets an overall accuracy of 59.1%, which is slightly lower than the accuracy reported in the literature (59.5%). Such minor difference may be caused by the different schemes used for selecting the SVM parameters. Our current study uses the same optimal parameters to test all the groups during cross-validation, whereas the Shamim method selected optimal parameters for each test group (Supplementary Table II of Shamim et al., 2007). In addition, the Shamim method seems to be not robust, since the best results of different datasets are obtained with different combinations of features (Shamim et al., 2007). The LA method has extremely low performance. This may be due to the low sequence identity in this dataset, noting that the LA kernel measures the sequence similarity using only the sequence information. The SWPSSM kernel achieves attractive performance with an overall accuracy of 67.8%, which uses the PSSM to compute the kernel. Such a result indicates that evolutionary information is more informative than the sequence itself.

To give a more comprehensive comparison, we consider several other methods in the literature. All these methods have been tested on the D-B dataset, and their best accuracies are presented in Table 3. They either use different sequence features or employ different algorithms for multi-class classification. As can be seen from the table, the proposed ACCFold method outperforms these methods by 2–14%.

3.2.2 Results on the extended D-B dataset In the D-B dataset, each fold contains a relatively small number of samples, which is unfavorable to fold prediction. So the extended D-B dataset is used. The performance comparison with three other taxonomy-based methods by 5-fold cross-validation is shown in Table 4. The detailed results are given in the Supplementary Material. As shown in the table, all the methods get improved performance in comparison with that obtained on the D-B dataset. This may be mainly attributed to the larger number of samples in the extended D-B dataset, from
which powerful classifiers can be trained. Other related studies have also demonstrated that the usage of extended datasets can improve the performance of protein fold recognition (Deschavanne and Tuffery, 2009; Shamim et al., 2007). The ACCFold method with AC variable achieves an overall accuracy of 81.7%. The usage of ACC variable boosts the accuracy to 87.6%, which is the best result for multi-class protein fold recognition so far. The sequence length has minor influence on the proposed methods (see Supplementary Table 3). Both the short sequences and the long sequences can be quite successfully predicted. All the folds in different classes are better predicted than those of the D-B dataset. Especially, the performance of the folds in the $\alpha/\beta$, $\alpha+\beta$ and small proteins classes are significantly improved. This may be attributed to the extra samples in the extended dataset, such that the sample space can be effectively covered. Another interesting observation is that the folds with large numbers of samples must not achieve absolutely high performance. This can be explained as follows. In the training of SVM, only the samples near the decision boundary contribute to the classifier, while most of the samples, which are far away from the decision boundary, are not used. The Shamim method gets an overall accuracy of 68.9%, which is slightly lower than that reported in the literature (70%) (Shamim et al., 2007). Besides the factor mentioned above, another reason is the difference of datasets. Even the LA method, which uses the sequence alone, has significantly improved performance on the extended dataset with an overall accuracy of 61.3%. The SWPSSM method gets an overall accuracy of 86.8%, which is comparable with the ACCFold method. Both methods use evolution information but in different manners. The SWPSSM method measures the similarity between any two residues along the sequence, which takes the sequence-order effect into consideration. In addition, the ACCFold method is superior to SWPSSM method in terms of time complexity. Both methods first compute the kernels between any two samples and then input the kernels to SVM for fold recognition. The SWPSSM method is based on the dynamic programming algorithm, which need a time complexity of $O(n^2L^2)$ to compute the kernel, where $n$ is the total number of samples and $L$ is the length of the longest sequence in the dataset. The ACCFold method first converts the sequence into a fixed-length vector and then computes the kernel by the kernel function, which need a time complexity of $O(n^2L^2)$ as well. The outstanding performance of our approach can be attributed to the effective usage of evolutionary information by ACC transformation as well as well-trained fold-specific SVM.

### Table 5. The best overall accuracy comparison with some typical existing taxonomic methods on the extended D-B dataset

<table>
<thead>
<tr>
<th>Methods</th>
<th>$Q$</th>
<th>SCOP version</th>
<th>No.</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shamim</td>
<td>70.5</td>
<td>171</td>
<td>2554</td>
<td>Shamim et al. (2007)</td>
</tr>
<tr>
<td>ACCFold</td>
<td>87.6</td>
<td>173</td>
<td>3202</td>
<td>Deschavanne and Tuffery (2009)</td>
</tr>
</tbody>
</table>

*The total number of proteins in the dataset.

### Table 6. Performance comparison on the F86 and F199 datasets

<table>
<thead>
<tr>
<th>Methods</th>
<th>F86</th>
<th>F199</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACCFold</td>
<td>ave Sp</td>
<td>ave Sn</td>
</tr>
<tr>
<td>ACCFold_ACC</td>
<td>80.9</td>
<td>89.4</td>
</tr>
<tr>
<td>ACCFold_AC</td>
<td>75.6</td>
<td>84.2</td>
</tr>
<tr>
<td>SWPSSM</td>
<td>82.5</td>
<td>95.8</td>
</tr>
<tr>
<td>Shamim</td>
<td>59.9</td>
<td>41.8</td>
</tr>
<tr>
<td>LA</td>
<td>44.0</td>
<td>57.1</td>
</tr>
</tbody>
</table>

### 3.2.3 Results on the F86 and F199 datasets

The results on these two datasets are listed in Table 6. The ACCFold method with ACC variable gets an overall accuracy of 80.9% on the F86 dataset and 77.2% on the F199 dataset. The results indicate that the proposed method can be applied to the cases of large number of folds without significantly affecting its performance, as long as the number of samples in each fold is not too small. More folds will be covered if the number of sequences in the non-redundant SCOP subset increases, for example, by adopting a larger threshold of the sequence identity. It is possible to employ other...
At the family level, we select the families that contain at least two
ACCFold_ACC 53.9  a 79.6  b 95.7  c 23.1  d 55.4  e 78.3  f 29.9  g 41.4  h 51.9
ACCFold_AC 53.1  i 79.5  j 93.6  k 20.0  l 47.7  m 64.0  n 28.0  o 41.3  p 50.9
Performance comparison with threading methods on the Lindahl dataset
Table 7. (Cheng and Baldi, 2006) and SP3/SP5 (Zhang et al., 2005). The results are listed in Table 8. The ACCFold_ACC method outperforms the other taxonomic methods. Experiments are also performed at the family, the superfamilies and the fold level. A 2-fold cross-validation is used to get the final results. The scheme of sample segmentation is the same as that used on the Lindahl dataset. At each level, we select the categories that contain at least five samples in each group (N_{min}=5). All the taxonomic fold recognition methods are locally implemented and tested on this dataset. However, most of the threading methods do not have available packages or are difficult to be configured. So we only compare the taxonomic fold recognition methods with the HHpred method (Soding et al., 2005). The results are listed in Table 8. The ACCFold_ACC method outperforms the other taxonomic methods at all levels. In comparison with the HHpred method, the performance of ACCFold_ACC method is slightly lower than that of HHPred method at the family and superfamilal level, but higher than that of HHPred method at the fold level. Similar result is gotten on the Lindahl dataset.

Since HHpred is not the top-performance threading method, these results may indicate that the taxonomic methods are not as good as threading methods and they are currently difficult to be applied to practical fold recognition due to the small number of samples in many folds. However, it is generally regarded that the total number of folds are limited. And as the number of proteins with known structure increases, there will be more and more space and chance to exploit the taxonomic methods to develop effective fold cognition system. The proposed ACCFold method, as a new taxonomic protein sophisticated techniques to construct more sensitive classifiers for practical applications, such as training samples selection, active learning, incremental learning, etc., which are left to our future work.

3.3 Performance comparison with threading methods
3.3.1 Results on the Lindahl dataset The taxonomic fold recognition methods construct classifiers to discriminate the correct folds of proteins while the threading methods, such as RAPTOR (Xu, 2005; Xu et al., 2003), HHpred (Soding et al., 2005), FOLDPro (Cheng and Baldi, 2006) and SP3/SP5 (Zhang et al., 2008), use the sequence–template alignments to detect the remote homologies of proteins. To compare these two kinds of methods, the Lindahl dataset (Lindahl and Elofsson, 2000) is used as a benchmark. Remote homology is simulated at the family, the superfamilies and the fold level, respectively. A 2-fold cross-validation is performed. At the family level, we select the families that contain at least two samples, and the samples in each family are randomly divided into two groups for cross-validation. At the superfamilies (or the fold level), we select the superfamilies (or folds) that contain at least two families (or superfamilies), and the samples of the same family (superfamily) are placed into the same group for cross-validation, so the training samples and the testing samples come from different families (superfamilies). Testing on the whole dataset is unfair for the taxonomic fold recognition methods because many categories at each level contain extremely small numbers of samples. Most of the taxonomic methods cannot construct effective classifiers with so small numbers of samples. So the categories at each level are further selected by specifying that the number of samples in each group should be larger than or equal to a threshold N_{min}.

Table 7 lists the results of five taxonomic fold recognition methods and five threading methods. The results of taxonomic methods are obtained by 2-fold cross-validation, and the results of threading methods are taken from the references. When the minimum number of samples in each category is 2 (N_{min}=1), the performance of the taxonomic methods is lower than that of the threading methods. As the number of samples increases, the performance of the taxonomic methods get better. Of course, the number of categories decreases as N_{min} increases, where homology detection seems to be easier for the taxonomic methods.
fold recognition method, has shown the state-of-the-art performance, when compared with the existing taxonomy-based methods. Thus, we think it is potentially useful and has considerable academic significance.

4 CONCLUSIONS

In this study, a novel method (ACCFold) that combines SVM with ACC is introduced for taxonomic protein fold recognition. ACC transformation is used to convert the PSSMs into fixed-length vectors, which are input to SVM for fold recognition. Testing on the D-B dataset and the extended D-B dataset has shown that the ACCFold method outperforms existing taxonomic methods. When only 27 folds are considered, the proposed method gets an overall accuracy of 70.1% on the D-B dataset and 87.6% on the extended dataset. Furthermore, the method achieves an overall accuracy of 80.9% for 86 folds and 77.2% for 199 folds. The results obtained here stand for the state-of-the-art performance of taxonomic protein fold recognition. We think that ACC transformation can be a valuable tool for protein-level studies.

Funding: National Natural Science Foundation of China (No. 60873040 and No. 60873070); Shanghai Leading Academic Discipline Project (No. B114); China Postdoctoral Science Foundation (No. 200804040572); the Program for New Century Excellent Talents in University of China (NCET-06-0376 to J.G.).

Conflict of Interest: none declared.

REFERENCES


Table 8. Performance comparison with threading methods on the F66 dataset

<table>
<thead>
<tr>
<th>Method</th>
<th>Family level</th>
<th>Superfamily level</th>
<th>Fold level</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of sequences</td>
<td>2533</td>
<td>3851</td>
<td>3397</td>
</tr>
<tr>
<td>No. of categories</td>
<td>117</td>
<td>92</td>
<td>42</td>
</tr>
<tr>
<td>ACCFold_ACC</td>
<td>86.6</td>
<td>43.5</td>
<td>42.2</td>
</tr>
<tr>
<td>ACCFold_AC</td>
<td>81.6</td>
<td>34.5</td>
<td>34.4</td>
</tr>
<tr>
<td>SWPSSM</td>
<td>86.2</td>
<td>38.9</td>
<td>42.1</td>
</tr>
<tr>
<td>Shamim</td>
<td>48.2</td>
<td>20.1</td>
<td>31.3</td>
</tr>
<tr>
<td>LA</td>
<td>35.7</td>
<td>14.6</td>
<td>25.1</td>
</tr>
<tr>
<td>HHPred</td>
<td>87.3</td>
<td>69.3</td>
<td>37.0</td>
</tr>
</tbody>
</table>


