

Structural bioinformatics

Support Vector Machine-based classification of protein folds using the structural properties of amino acid residues and amino acid residue pairs

Mohammad Tabrez Anwar Shamim, Mohammad Anwaruddin and H.A. Nagarajaram*

Laboratory of Computational Biology, Centre for DNA Fingerprinting and Diagnostics, Hyderabad 500 076, India

Received on July 5, 2007; revised on September 25, 2007; accepted on October 15, 2007

Advance Access publication November 7, 2007

Associate Editor: Burkhard Rost

ABSTRACT

Motivation: Fold recognition is a key step in the protein structure discovery process, especially when traditional sequence comparison methods fail to yield convincing structural homologies. Although many methods have been developed for protein fold recognition, their accuracies remain low. This can be attributed to insufficient exploitation of fold discriminatory features.

Results: We have developed a new method for protein fold recognition using structural information of amino acid residues and amino acid residue pairs. Since protein fold recognition can be treated as a protein fold classification problem, we have developed a Support Vector Machine (SVM) based classifier approach that uses secondary structural state and solvent accessibility state frequencies of amino acids and amino acid pairs as feature vectors. Among the individual properties examined secondary structural state frequencies of amino acids gave an overall accuracy of 65.2% for fold discrimination, which is better than the accuracy by any method reported so far in the literature. Combination of secondary structural state frequencies with solvent accessibility state frequencies of amino acids and amino acid pairs further improved the fold discrimination accuracy to more than 70%, which is ~8% higher than the best available method. In this study we have also tested, for the first time, an *all-together* multi-class method known as *Crammer and Singer* method for protein fold classification. Our studies reveal that the three multi-class classification methods, namely *one versus all*, *one versus one* and *Crammer and Singer* method, yield similar predictions.

Availability: Dataset and stand-alone program are available upon request.

Contact: han@cdfd.org.in

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

1 INTRODUCTION

The gap between the number of proteins with and without 3-dimensional (3D) structural information has been increasing alarmingly owing to the successful completion of many genome-sequencing projects. Since 3D structure is essential for understanding protein function, and as not all proteins are amenable to experimental structure determination,

computational prediction of 3D structures has, therefore, become a necessary alternative to experimental determination of 3D structures. Among the computational approaches, fold recognition/threading methods have taken central stage. In instances where detection of homology becomes difficult even when using the best sequence comparison methods such as PSI-BLAST (Altschul *et al.*, 1997), structure-based fold recognition approaches are often employed. Many methods have been developed, which are used for assigning folds to protein sequences. These can be broadly classified in three categories: (a) sequence–structure homology recognition methods such as FUGUE (Shi *et al.*, 2001) and 3DPSSM (Kelley *et al.*, 2000), (b) threading methods such as THREADER (Jones *et al.*, 1992) and (c) taxonomic methods such as PFP-Pred (Shen and Chou, 2006).

Sequence–structure homology recognition methods align target sequence onto known structural templates and calculate their sequence–structure compatibilities using either profile-based scoring functions (Kelley *et al.*, 2000) or environment-specific substitution tables (Shi *et al.*, 2001). The scores obtained for different structural templates are then ranked and the template, which gives rise to the best score, is assumed to be the fold of the target sequence. Unfortunately, these methods, although widely used, have not been able to achieve accuracies >30% at the fold level (Cheng and Baldi, 2006), which could be attributed to the fact that these methods use substitutions to detect folds that are evolutionally related. Threading methods, which use pseudo-energy based functions (Jones *et al.*, 1992) to calculate sequence–structure compatibilities also yield poor accuracies perhaps due to the difficulty of formulating reliable and general scoring functions.

On the other hand, taxonomic methods for protein fold recognition such as the one developed by Ding and Dubchak (2001) and PFP-Pred (Shen and Chou, 2006) that give prediction accuracies of ~60%, assume that the number of protein folds in the universe is limited and therefore, the protein fold recognition can be viewed as a fold classification problem, where a query protein can be classified into one of the known folds. In this classification scheme one needs to identify fold-specific features, which can discriminate between different folds. Available taxonomic methods for protein fold recognition use amino acid composition, pseudo amino acid

*To whom correspondence should be addressed.

composition, and selected structural and physico-chemical propensities of amino acids as fold discriminatory features. Ding and Dubchak (2001) used amino acid composition and features extracted from structural and physico-chemical propensities of amino acids to train the discriminatory classifier. The Ensemble classifier approach for protein fold recognition developed by Shen and Chou (2006) used different orders of pseudo amino acid composition and structural and physico-chemical propensities of amino acids as features. In general, the taxonomic approach appears very promising for protein fold recognition and hence this approach can further be explored in order to obtain higher prediction accuracies by investigating new fold discriminatory features. In this study, we investigate the discriminatory potential of the secondary structural and solvent accessibility state information of amino acid residues and amino acid residue pairs. As shown, our approach gives a fold recognition accuracy which is ~8% higher than the best published fold recognition method.

2 METHODS

2.1 Datasets for training and testing

The investigations were performed on two datasets: (a) Ding and Dubchak dataset (D–B dataset), which is same as that used in earlier studies (Ding and Dubchak, 2001; Shen and Chou, 2006) and (b) extended D–B dataset, which is formed by further populating the D–B dataset with additional protein examples.

2.1.1 D–B dataset The D–B dataset contains 311 and 383 proteins for training and testing, respectively (<http://crd.lbl.gov/~cding/protein/>) (Supplementary Table 1). This dataset has been formed such that, in the training set, no two proteins have more than 35% sequence identity to each other and each fold have seven or more proteins; and in the test set, proteins have <40% sequence identity to each other and have not more than 35% identity to the proteins of the training set (Ding and Dubchak, 2001). According to SCOP classification (Murzin *et al.*, 1995), the proteins used for training and testing belong to 27 different folds representing all major structural classes: all α , all β , α/β , $\alpha + \beta$ and small proteins.

2.1.2 Extended D–B dataset This dataset (Supplementary Table 1) was formed by merging training and testing datasets of the D–B dataset and further populating each fold with additional protein examples chosen from ASTRAL SCOP 1.71 (Chandonia *et al.*, 2004; <http://astral.berkeley.edu>), where sequences have <40% identity to each other. This dataset comprises of 2554 proteins belonging to 27 folds.

2.2 Fold classifier method

For fold classification we have used Support Vector Machine (SVM), a supervised machine-learning method first developed by Vapnik (1995) that is extensively used for classification and regression problems. Literature abounds with technical details of SVM (Larranaga *et al.*, 2006; Vapnik, 1995; Yang, 2004).

SVM has been designed primarily for binary classification. Many methods have been developed to extend SVM to a multi-class classification (Crammer and Singer, 2000; Krebel, 1999). Currently, there are two kinds of methods: (a) the Binary classification-based method (Ding and Dubchak, 2001; Hsu and Lin, 2002; Krebel, 1999), which constructs and combines several binary classifiers and (b) the All-together method (Crammer and Singer, 2000; Vapnik, 1998), which directly considers all data in one big optimization formulation.

In general, a multi-class problem is computationally more expensive than a binary problem. Since protein fold recognition is typically a multi-class problem, we used multi-class methods, namely, All-together method (referred to as *Crammer and Singer* method) and the two Binary classification-based methods: *one versus all* and *one versus one*. *One versus all* and *one versus one* methods have been used earlier for protein fold recognition (Ding and Dubchak, 2001).

All SVM computations were carried out using LIBSVM (Chang and Lin, 2001). We used the *one versus one* implementation of LIBSVM 2.83 main code, *one versus all* implementation of LIBSVM error-correcting code and *Crammer and Singer* method implementation of BSVM 2.0. Although LIBSVM provides a choice of in-built kernels, such as Linear, Polynomial, Radial basis function (RBF) and Gaussian, we used RBF kernel for this study as it gave the best results (data not shown). The SVMs were trained using different values of the cost parameter $C = [2^{11}, 2^{10}, \dots, 2^{-3}]$ and kernel parameter $\gamma = [2^{-3}, 2^{-2}, \dots, 2^{-11}]$ and only those which gave rise to the best results were retained.

2.3 Fold discriminatory features

The sequence- and structure-based features extracted for this study are listed in Table 1.

2.3.1 Sequence-based features *Amino acid composition*: amino acid composition compresses the protein information into a fixed length vector in 20-dimensional space. This feature has been used with significant success, for predicting sub cellular localization of proteins (Garg *et al.*, 2005; Guo *et al.*, 2006), classification of nuclear receptors (Karchin *et al.*, 2002) and protein fold recognition (Ding and Dubchak, 2001). The composition of an amino acid i in a protein is calculated using the formula:

$$f_i = \frac{N_i}{L}$$

where f_i = frequency of amino acid i ; N_i = number of amino acid i found in that protein; L = total number of amino acid residues found in that protein and $i = 1$ to 20.

Amino acid pair composition: amino acid pair composition or an n th order amino acid pair encapsulates the interaction between the i th and $(i+n)$ th ($n > 0$) amino acid residues and gives the local order information as well as the composition of amino acids in a protein. Amino acid pair composition is a 400 (20×20) dimensional representation of protein information, which has been shown to work well for many problems, such as subcellular localization of proteins (Garg *et al.*, 2005; Guo *et al.*, 2006); classification of G-protein-coupled receptors (Karchin *et al.*, 2002), etc. The n th order of amino acid pair composition in a protein is calculated using the formula:

$$f(D^{i,i+n})_j = \frac{N(D^{i,i+n})_j}{L - n}$$

where $f(D^{i,i+n})_j$ is the frequency of an n th order amino acid pair j ; $N(D^{i,i+n})_j$ is the number of n th order amino acid pair j ; n is the order of amino acid pair and $j = 1$ to 400.

2.3.2 Structure-based features *Secondary structural state (H, E, C) frequencies of amino acids*: these are the frequencies of amino acids found in helices (H), β -strands (E) and coils (C) in a given protein and are collectively represented as a 60 (20×3) dimensional vector. The frequencies are calculated using the formula:

$$f_i^k = \frac{N_i^k}{L}$$

where $k = (H, E, C)$; f_i^k is the frequency of amino acid i occurring in the secondary structural state k and N_i^k is the number of amino acid i found in the secondary structural state k . In this study, we have used predicted

Table 1. Different features along with their dimensions, used for training SVM classifiers

Feature Index	Feature	Dimensions
Individual features		
Sequence Features		
1	Amino acid composition	20
2	First order amino acid pair (dipeptide) composition	400
3	Second order amino acid pair (1-gap dipeptide) composition	400
Structural Features		
4	Secondary structural state frequencies of amino acids	60
5	Secondary structural state frequencies of dipeptides	1200
6	Secondary structural state frequencies of 1-gap dipeptides	1200
7	Solvent accessibility state frequencies of amino acids	40
8	Solvent accessibility state frequencies of dipeptides	1200
9	Solvent accessibility state frequencies of 1-gap dipeptides	1200
Combination of features		
10	Feature4 + Feature7	100
11	Feature5 + Feature8	2400
12	Feature6 + Feature9	2400
13	Feature4 + Feature5 + Feature6	2460
14	Feature7 + Feature8 + Feature9	2440
15	Feature4 + Feature7 + Feature5 + Feature8	2500
16	Feature4 + Feature7 + Feature6 + Feature9	2500
17	Feature5 + Feature8 + Feature6 + Feature9	4800
18	Feature4 + Feature7 + Feature5 + Feature8 + Feature6 + Feature9	4900

secondary structural information as the basis for all the calculations. The predictions were made using PSIPRED (McGuffin *et al.*, 2000) and only those with confidence level ≥ 1 were considered for calculations.

Secondary structural state frequencies of amino acid pairs: these collectively represent a 1200 (400×3) dimensional vector. An amino acid pair was considered as found in helix or β -strand, only if both the residues were found in helix or strand, respectively, otherwise the pair was considered as found in coil. Secondary structural state frequency of an n -order amino acid pair is calculated using the formula:

$$f(D_k^{i,i+n})_j = \frac{N(D_k^{i,i+n})_j}{L-n}$$

where $k = (H, E, C)$; $f(D_k^{i,i+n})_j$ is the frequencies of an n th order amino acid pair j in secondary structural state k and $N(D_k^{i,i+n})_j$ is the number of an n th order amino acid pair j found in secondary structural state k .

Solvent accessibility state (B, E) frequencies of amino acids: solvent accessibility state frequency of amino acids is a 40-dimensional representation of protein structural information and is calculated as follows:

$$f_i^k = \frac{N_i^k}{L}$$

where $k = (B, E)$; f_i^k is the frequency of amino acid i in solvent accessibility state k and N_i^k is the number of amino acid i in solvent accessibility state k . We used predicted solvent accessibility states for calculating these frequencies. ACCpro (Cheng *et al.*, 2005) was used for predicting the solvent accessibility states of amino acid residues [cut off value for relative solvent accessibilities were $\leq 10\%$ and $> 10\%$ for buried (B) and exposed (E), respectively].

Solvent accessibility state frequencies of amino acid pairs: these comprise a 1200-dimensional representation of protein structural information. An amino acid pair was considered as buried (B) or exposed (E) only if both the residues were found buried or exposed,

respectively. All other pairs were considered as partially buried (I). The solvent accessibility state frequency of an n th order amino acid pair is calculated using the formula:

$$f(D_k^{i,i+n})_j = \frac{N(D_k^{i,i+n})_j}{L-n}$$

where $k = (B, E, I)$; $f(D_k^{i,i+n})_j$ and $N(D_k^{i,i+n})_j$ are the frequency and number of the n th order amino acid pair j found in solvent accessibility state k and $L-n$ is the total number of n th order amino acid pairs.

2.4 Performance measures

The performance of fold classification by SVM was evaluated by computing overall accuracy (Q), sensitivity (Sn) and specificity (Sp). Overall accuracy is the most commonly used parameter for assessing the global performance of a multi-class problem (Ding and Dubchak, 2001; Pierleoni *et al.*, 2006), and is defined as the number of instances correctly predicted over the total number of instances in the test set:

$$Q = \frac{\sum_i z_i}{N} \times 100$$

where N is the total number of proteins (instances) in the test set, and z_i are the true positives.

Sensitivity and specificity were calculated using formulae:

$$\text{Sensitivity (Sn)} = \frac{(\text{TP} \times 100)}{(\text{TP} + \text{FN})}$$

$$\text{Specificity (Sp)} = \frac{(\text{TP} \times 100)}{(\text{TP} + \text{FP})}$$

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

The n -fold cross-validation is generally used to check the generalization and stability of a method (Bhasin and Raghava, 2004; Goutte, 1997; Wang *et al.*, 2006). In this study, we performed 2-fold

Table 2. The overall 2-fold cross-validation accuracies obtained for three multi-class methods—(a) *one versus all*, (b) *one versus one* and (c) *Crammer and Singer*

Individual features				Combination of features			
Feature index	OVA	OVO	C&S	Feature index	OVA	OVO	C&S
	Q_{cv}	Q_{cv}	Q_{cv}		Q_{cv}	Q_{cv}	Q_{cv}
1	49.0	48.8	48.8	10	60.5	59.5	59.5
2	47.5	47.3	46.5	11	53.3	47.8	52.0
3	50.3	48.9	47.6	12	55.5	49.2	54.4
4	56.9	54.2	55.7	13	54.7	51.2	52.5
5	49.2	48.6	49.5	14	55.7	48.5	54.7
6	49.9	48.9	49.3	15	56.0	50.8	54.4
7	52.0	49.0	51.5	16	58.8	51.6	57.4
8	51.1	46.0	50.1	17	55.9	46.8	55.7
9	52.5	47.0	51.4	18	54.0	46.0	55.0

The training and testing were carried out twice (2-fold cross-validation) for D–B dataset. In the first run, training was done on the training set and validation on the test set and in the second run, training was done on the test set and validation on train set. The average of the accuracies obtained in the two runs is given as the 2-fold cross-validation accuracy (Q_{cv}). The best Q_{cv} obtained for a multi-class method is shown in bold.

cross-validation using the D–B dataset and 5-fold cross-validation using the extended D–B dataset.

We also checked the classification performance of input features using a naive Bayes classifier. The classifier was downloaded from <http://www.borgelt.net/bayes.html>, and trained with default parameters using the same input features as used in SVM. The performance was evaluated using 2-fold and 5-fold cross-validation for D–B and extended D–B dataset, respectively.

3 RESULTS AND DISCUSSION

We conducted preliminary studies to test the usefulness of the different orders (n) ranging from 1 to 12, of amino acid pairs. Our studies revealed that only the amino acid pairs with the first ($n=1$) and second ($n=2$) orders give good fold prediction accuracies. The prediction accuracy declined from the third order pair onwards (please see the Supplementary Fig. 1). The decrease in prediction accuracy is due to increase uncertainty in backbone conformation as the spacing between the amino acids, i.e. ' n ' in the pair increases. Therefore, for further studies we considered only the first and second order of amino acid pairs.

3.1 Twofold cross-validation studies using D–B dataset

We analyzed individual fold discriminatory potentials of the sequence and the structure-based features as given in Table 1. The prediction accuracies yielded by the various features for the three multi-class methods and their corresponding values of C and γ are given in Supplementary Table 2. Among the nine individual features used in this study, secondary structural state frequencies of amino acids (Feature 4) gave the best overall Q_{cv} (2-fold cross-validation accuracy) value of 57% (Table 2). As mentioned earlier, we also examined the fold discriminatory potential of different combinations of the features. Of these, Feature10—the combination of secondary structural state and

solvent accessibility state frequencies of amino acids gave the best 2-fold accuracy of 60% (Table 2).

The sensitivity and specificity of the best classifier set Feature10 as obtained by the three multi-class methods are shown in Figure 1. As is evident from the figure, the sensitivity and specificity values do not remain the same for all the folds. In general, the folds, which are mostly α -helical, such as globin-like and cytochrome c, show high sensitivity and specificity. The average prediction accuracy (i.e. sensitivity) obtained for 'all α class' folds is $\sim 78\%$ as compared to $\sim 56\%$ obtained for 'all β class' folds. This difference in prediction accuracies between the two classes of folds can be attributed to the accuracies associated with the prediction of secondary structures and solvent accessibilities of the amino acids and amino acid pairs in these folds. In general, α -helices are predicted with better accuracies than the β -strands (Rost and Sander, 1993 and Table 3) and, therefore, any prediction approach, such as the one presented here, which uses predicted secondary structural information, is already biased towards better prediction of folds in α -class than β -class.

In this study, as mentioned earlier, the methods used for secondary structure and solvent accessibilities are, respectively, PSIPRED and ACCpro and their prediction accuracies are $\sim 78\%$ (McGuffin *et al.*, 2000) and $\sim 77\%$ (Cheng *et al.*, 2005), respectively. As the structures for the protein domains in the D–B dataset are known, we identified the secondary structural states using SSTRUC (Smith, 1989) and calculated the solvent accessibilities using PSA (Sali, 1991) and these were compared with the predictions (Table 3). It is interesting to note that most of the low performing folds show marked errors in their predicted secondary structural states. For example, the OB-fold shows $\sim 22\%$ error in strand prediction (E_s); trypsin-like serine proteases fold, $\sim 25\%$ error in strand prediction; ribonuclease H-like motif fold, $\sim 15\%$ error in helix prediction and $\sim 23\%$ error in strand prediction.

Similarly, the low performing folds show significant errors in the prediction of solvent accessibility states of the residues. Failure to predict the correct number of residues buried can arise in the case of domains, which form parts of multi-domain proteins. In such cases, the solvent accessibility prediction program does not give proper prediction, as contact residues between domains, which are actually buried, are predicted as exposed. We calculated the percentage of such domains, which are part of multi-domain proteins in each fold (Table 3). It turns out that most of the folds characterized by the domains from multi-domain proteins give rise to low accuracies.

In addition to the influence of incorrectly predicted features, the SVM training can also become error prone due to the sparseness of the dataset used. It is known that performance of the SVM depends on the size of the dataset used for training because it learns from the examples. The greater the number of examples (for both positives and negatives) available for learning, the better would be the model. A look at the D–B dataset (Supplementary Table 1) reveals that many folds are sparsely represented. For example, folds such as the immunoglobulin-like β -sandwich and TIM-barrel show good sensitivity but poor specificity. These are the most populated folds in the D–B dataset. Generally, training, in such cases, becomes skewed towards populous folds labeled as positive

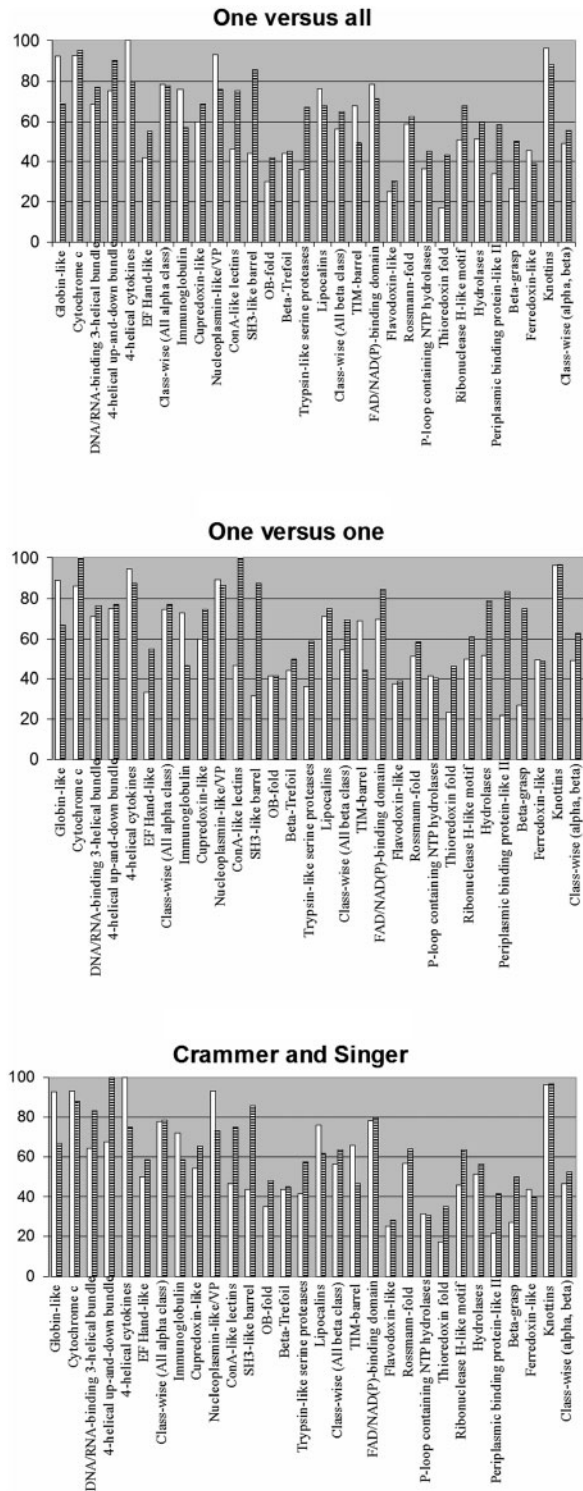


Fig. 1. Fold-wise sensitivity and specificity obtained for the best classifier feature—Feature10 for three multi-class methods: (A) *one versus all* (OVA), (B) *one versus one* (OVO) and (C) *Crammer and Singer* (C&S). As mentioned in Table 1, Feature10 is the combination of secondary structural and solvent accessibility state frequencies of amino acids. Overall prediction accuracy using Feature10 is 60.5% for *one versus all* (OVA), 59.5% for *one versus one* (OVO) and 59.5% for *Crammer and Singer* (C&S) method.

Table 3. Discrepancy (error) in secondary structural state (helix, strand) prediction and solvent accessibility state (buried) prediction for different folds

Fold	E_h	E_s	E_b	DOM_{mdp}
All α class				
Globin-like	5.5	–	19.4	0.0
Cytochrome c	12.3	–	34.7	31.2
DNA/RNA-binding 3-helical bundle	10.5	–	33.8	34.4
4-helical up-and-down bundle	7.9	–	29.8	13.3
4-helical cytokines	11.5	–	35.5	0.0
EF Hand-like	11.5	–	25.7	20.0
Class-wise average	9.9		29.8	16.5
All β class				
Immunoglobulin-like β -sandwich	–	17.8	32.8	43.2
Cupredoxin-like	–	16.9	26.3	47.6
Nucleoplasm-like/VP	–	27.1	29.8	0.0
ConA-like lectins/glucanases	–	18.9	16.0	15.4
SH3-like barrel	–	17.6	40.9	12.5
OB-fold	–	21.3	32.7	37.5
Beta-Trefoil	–	18.1	30.1	16.7
Trypsin-like serine proteases	–	25.3	16.7	0.0
Lipocalins	–	9.6	24.8	0.0
Class-wise average		19.2	27.8	19.2
α, β (α/β, $\alpha \pm \beta$ and small proteins)				
TIM beta/alpha-barrel	13.4	24.0	10.5	33.8
FAD/NAD(P)-binding domain	16.7	23.6	25.6	82.6
Flavodoxin-like	13.6	13.7	14.2	54.2
NAD(P)-binding Rossmann	7.7	16.5	21.8	72.5
P-loop containing NTH	13.6	20.0	18.3	36.4
Thioredoxin fold	10.7	10.5	23.7	47.1
Ribonuclease H-like motif	15.2	23.2	22.6	63.6
Hydrolases	17.5	22.4	8.3	16.7
Periplasmic binding protein-like II	7.9	14.7	5.5	6.7
β -grasp (ubiquitin-like)	21.6	17.2	28.8	40.0
Ferredoxin-like	16.8	27.7	33.8	30.0
Knottins	55.5	55.2	51.2	37.5
Class-wise average	17.5	22.4	22.0	43.4

The errors were calculated by comparing the assigned and predicted states. E_h and E_s are the fold-wise percentage error in helix and strand prediction. E_b is the fold-wise percentage error in buried state prediction. Last column shows percentage of domains, which are part of multi-domain proteins (MDP) in each fold (DOM_{mdp}).

rather than lesser populated folds labeled as negative; hence as a result many proteins that do not belong to the populous folds get classified as positives.

3.2 Fivefold cross-validation studies using extended D–B dataset

In order to remove any bias due to inadequate data, the D–B dataset was populated by adding representatives taken from ASTRAL SCOP 1.71 (Chandonia *et al.*, 2004). The new dataset referred to as the extended D–B dataset, is almost four times larger in size than the D–B dataset. This dataset was used to perform 5-fold cross-validation by randomly dividing the dataset into five equal size sets (I, II, III, IV and V) and in each round of cross-validation, training was carried out using four sets and testing using the remaining set. The prediction

Table 4. The overall 5-fold cross-validation accuracy (Q_{cv}) along with SD (enclosed within parentheses) obtained for all the features using three multi-class methods—*one versus all*, *one versus one* and *Cramer and Singer*

Feature index	OVA Q_{cv}	OVO Q_{cv}	C&S Q_{cv}
1	42.7 (0.9)	44.1 (0.8)	43.0 (1.3)
2	48.9 (1.9)	50.5 (1.0)	45.0 (1.9)
3	48.8 (1.4)	50.4 (0.7)	43.1 (1.5)
4	64.9 (1.1)	65.2 (1.1)	63.4 (1.4)
5	60.3 (1.0)	63.7 (0.8)	56.3 (1.8)
6	58.8 (0.4)	62.5 (1.5)	53.9 (0.6)
7	54.2 (2.0)	54.3 (1.3)	53.9 (1.6)
8	57.0 (2.0)	58.3 (1.5)	54.4 (2.1)
9	56.9 (1.7)	59.0 (1.8)	54.1 (1.5)
10	68.7 (1.7)	68.7 (2.5)	68.8 (1.9)
11	65.2 (1.5)	68.3 (1.3)	63.0 (1.3)
12	63.4 (1.3)	67.4 (1.7)	61.2 (0.6)
13	66.0 (1.5)	68.7 (1.3)	62.4 (1.6)
14	63.6 (1.5)	64.3 (1.2)	62.7 (3.5)
15	67.7 (2.0)	70.5 (0.7)	65.3 (2.1)
16	66.2 (1.6)	70.3 (1.2)	64.1 (1.2)
17	65.6 (2.0)	67.6 (1.6)	64.0 (1.5)
18	66.4 (1.2)	68.4 (0.6)	65.8 (1.9)

In the 5-fold cross-validation, the extended D-B dataset was randomly divided into five equal size sets. In each round of cross-validation four sets were used for training and the remaining set was used for testing. The best Q_{cv} obtained for a multi-class method is shown in bold.

accuracies achieved by the various features for the three multi-class methods and their corresponding values of C and γ are given in Supplementary Table 3.

Among the individual features tested, the secondary structural state frequencies of amino acids (Feature 4) gave the best 5-fold accuracy of 65% (Table 4), which is higher than the best accuracy (62%) reported in the literature, by the Ensemble classifier approach PFP-Pred (Shen and Chou, 2006). Among the feature combinations, Feature15—combination of secondary structural state and solvent accessibility state frequencies of amino acids and first-order amino acid pairs—gave the highest accuracy of 70.5% (Table 4). This corresponds to the highest accuracy reported in the literature. The feature combination, Feature10, which showed the highest accuracy in 2-fold cross-validation studies, achieved a 5-fold accuracy of around 69%.

The sensitivities and specificities obtained for the different folds are shown in Figure 2. As can be seen from the figure the prediction accuracies for many folds (i.e. EF Hand-like, immunoglobulin-like β -sandwich, TIM-barrel, trypsin-like serine proteases, etc.), have improved significantly as compared to the results obtained from 2-fold cross-validation studies. This shows that dataset size influences the quality of the training of SVM and hence the accuracy of prediction.

Another interesting result is the increase in the specificity values of the populous folds, which showed poor specificity in 2-fold cross-validation studies. This indicates that poor specificity in 2-fold cross-validation studies is due to sparseness

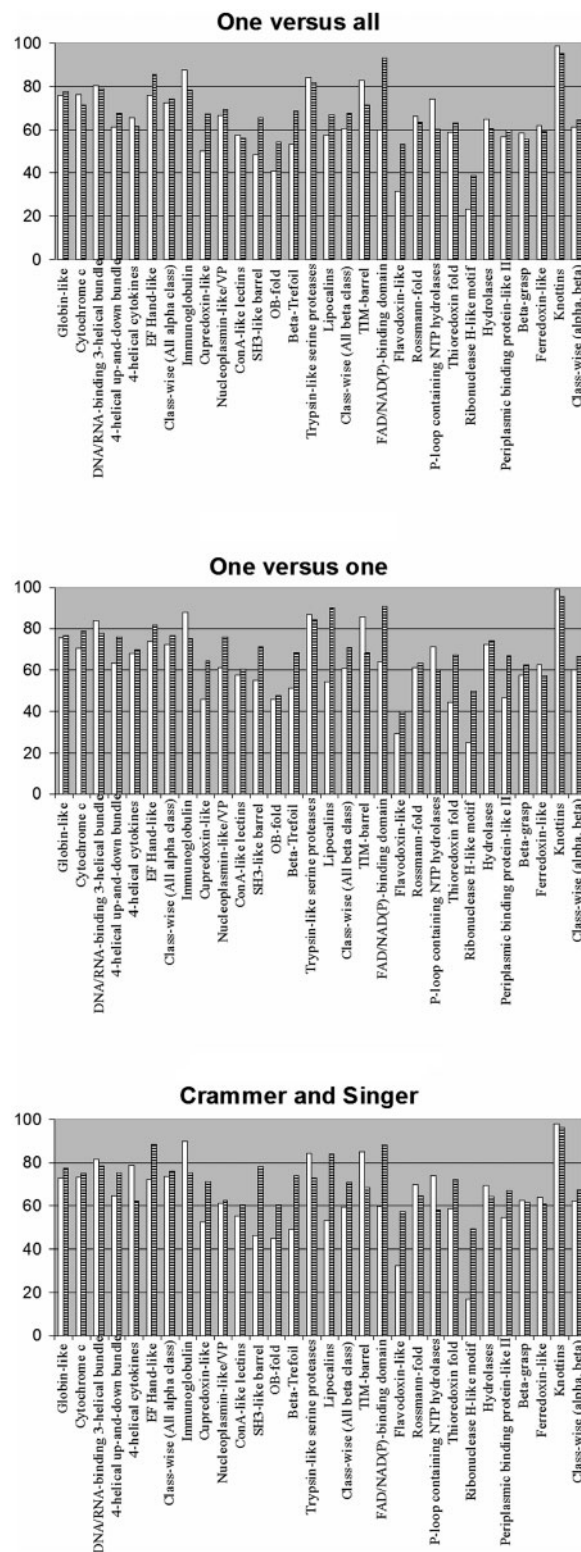


Fig. 2. Fold-wise sensitivity and specificity for the classifier feature—Feature10 using three multi-class methods: (A) *one versus all* (OVA), (B) *one versus one* (OVO) and (C) *Cramer and Singer* (C&S). Overall prediction accuracy using Feature10 is 68.7% for *one versus all* (OVA), 68.7% for *one versus one* (OVO) and 68.8% for *Cramer and Singer* (C&S) method.

of the D–B dataset. Furthermore, poor specificity of the populous folds can also be attributed to their wide spread in the ‘Fold-space’ as revealed by phylogenetic studies (data not shown).

We computed the classification accuracy at the superfamily level (Supplementary Table 4). Superfamilies having at least 20 proteins in the extended D–B dataset were selected for the study. There are a total of 33 such superfamilies in the extended D–B dataset. The 5-fold accuracy of 74.2% was obtained. The sensitivities and specificities obtained for different superfamilies are also given in the Supplementary Table 4.

Finally, we estimated the generalization performance of SVM using the leave-one-out error estimate that is commonly used for this purpose. In the literature in addition to leave-one-out error estimate, Xi-Alpha error estimate has also been used. However, it has been shown that Xi-Alpha estimator overestimates the true error rate. In fact, Xi-Alpha estimator was developed as an alternative to the leave-one-out estimator as the latter is computationally very expensive (Joachims, 2000). The leave-one-out error estimate was calculated for the Feature10 and Feature15 using the extended D–B dataset and the results are shown in the Supplementary Figure 2. The leave-one-out error estimate is very similar to average error ($100 - Q_{cv}$) obtained for the 5-fold cross-validation. This result shows that the SVM is generally working well. Furthermore, the number of support vectors in the model (Supplementary Table 5) further strengthens the fact that the SVM is not over-trained for any specific dataset.

As mentioned earlier, we also calculated the prediction accuracies using a naïve Bayes classifier for the same input features. We found that the SVM performance is much superior to the naïve Bayes classifier (Supplementary Table 6).

3.3 Comparison of multi-class methods

It has been argued that *one versus one* method performs better than the *one versus all* multi-class method (Allwein et al., 2000; Furnkranz, 2002; Hsu and Lin, 2002). However, present study reveals that all the three multi-class methods yield similar overall accuracies, sensitivities and specificities (refer Tables 2 and 4; and Figs 1 and 2) indicating that the performance of SVM for the present set of features is independent of the type of multi-class method used; but dependent on the types of discriminatory features as well as the size of the dataset used for training. It is, however, worth noting that *one versus all* method is slower than the *one versus one* and *Crammer and Singer* method, especially for large dimensional features such as the ones used in the present study and hence any one of the latter methods is more useful in terms of execution time. To the best of our knowledge, this is the first time the *Crammer and Singer* multi-class method has been tested for protein fold classification problem.

3.4 Comparison with the other fold recognition methods

We compared the performance of our approach with that of other taxonomic fold recognition methods reported in literature; details are shown in Figure 3. For the sake of completion, we have also shown the prediction accuracies of the template-based fold recognition methods as reported in the

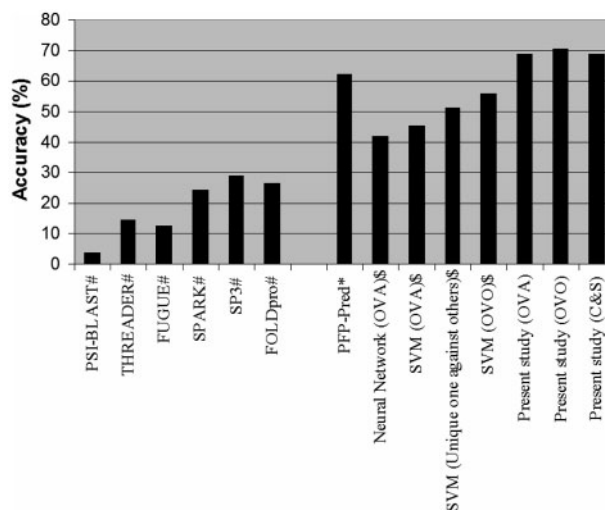


Fig. 3. The best prediction accuracy (Q) for protein fold recognition reported by different fold recognition methods. The Q-value for the template-based methods (#) corresponds to the % of top 1 hits match the correct folds. #Cheng and Baldi (2006), *Shen and Chou (2006) and \$Ding and Dubchak (2001).

literature. As evident from Figure 3, the prediction accuracy of our approach is ~8% higher than the best available method PFP-Pred. The strikingly better performance of our approach can be attributed to the more sensitive and specific fold discriminatory features as well as better trained fold-specific SVM.

4 CONCLUSIONS

In this study, we have investigated fold discriminatory potential of a number of sequence- and structure-based features using SVM. Our studies have revealed that the secondary structural and solvent accessibility state frequencies of amino acids and amino acid pairs collectively give rise to the best fold discrimination. The newly developed SVM-based approach presented in this study is stable and outperforms the other available methods and therefore can be used for fold-wise classification of unknown proteins discovered in various genomes.

ACKNOWLEDGEMENTS

H.A.N. gratefully acknowledges the core funding from CDFD. M.T.A.S. and M.A. are thankful to the Council of Scientific and Industrial Research (CSIR) for their research fellowships. Computational facilities of the SUN Centre of Excellence, CDFD is gratefully acknowledged. The authors thank Prof. Sir Tom Blundell for critically going through the manuscript and colleagues Sridhar and Pankaj for their wise help throughout this course of study. Finally, the authors gratefully acknowledge the two anonymous referees for their critical comments.

Conflict of Interest: none declared.

REFERENCES

- Allwein, E.L. *et al.* (2000) Reducing multi-class to binary: a unifying approach for margin classifiers. *J. Mach. Learn. Res.*, **1**, 113–141.
- 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.
- Bhasin, M. and Raghava, G. P. (2004) Classification of nuclear receptors based on amino acid composition and dipeptide composition. *J. Biol. Chem.*, **279**, 23262–23266.
- Chandonia, J. M. *et al.* (2004) The ASTRAL compendium in 2004. *Nucleic Acids Res.*, **32**, D189–D192.
- Chang, C.C. and Lin, C.J. (2001) LIBSVM: a library for support vector machines. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- Cheng, J. and Baldi, P. (2006) A machine learning information retrieval approach to protein fold recognition. *Bioinformatics*, **22**, 1456–1463.
- Cheng, J. *et al.* (2005) SCRATCH: a protein structure and structural feature prediction server. *Nucleic Acids Res.*, **33**, w72–w76.
- Crammer, K. and Singer, Y. (2000) On the learnability and design of output codes for multiclass problems. In *Proceedings of the Thirteenth Annual Conference on Computational Learning Theory (COLT-2000)*, pp. 35–46.
- Ding, C.H. and Dubchak, I. (2001) Multi-class protein fold recognition using support vector machines and neural networks. *Bioinformatics*, **17**, 349–358.
- Furnkranz, J. (2002) Round robin classification. *J. Mach. Learn. Res.*, **2**, 721–747.
- Garg, A. *et al.* (2005) Support vector machine-based method for subcellular localization of human proteins using amino acid compositions, their order, and similarity search. *J. Biol. Chem.*, **280**, 14427–14432.
- Goutte, C. (1997) Note on free lunches and cross-validation. *Neural Comput.*, **9**, 1211–1215.
- Guo, J. *et al.* (2006) GNBSL: a new integrative system to predict the subcellular location for Gram-negative bacteria proteins. *Proteomics*, **6**, 5099–5105.
- Hsu, C. and Lin, C. (2002) A comparison of methods for multi-class support vector machines. *IEEE Trans. Neural Netw.*, **13**, 415–425.
- Joachims, T. (2000) Estimating the generalization performance of an SVM efficiently. In *Proceedings of the Seventeenth International Conference on Machine Learning (ICML-2000)*, pp. 431–438.
- Jones, D.T. *et al.* (1992) A new approach to protein fold recognition. *Nature*, **358**, 86–89.
- Karchin, R. *et al.* (2002) Classifying G-protein coupled receptors with support vector machines. *Bioinformatics*, **18**, 147–159.
- Kelley, L.A. *et al.* (2000) Enhanced genome annotation using structural profiles in the program 3D-PSSM. *J. Mol. Biol.*, **299**, 499–520.
- Krebel, U. (1999) Pairwise classification and support vector machines. Advances in Kernel Methods- Support Vector Learning, MIT Press, Cambridge, MA. pp. 255–268.
- Larranaga, P. *et al.* (2006) Machine learning in bioinformatics. *Brief. Bioinformatics*, **7**, 86–112.
- McGuffin, L.J. *et al.* (2000) The PSIPRED protein structure prediction server. *Bioinformatics*, **16**, 404–405.
- Murzin, A.G. *et al.* (1995) SCOP: a structural classification of proteins database for the investigation of sequences and structures. *J. Mol. Biol.*, **247**, 536–540.
- Pierleoni, A. *et al.* (2006) BaCelLo: a balanced subcellular localization predictor. *Bioinformatics*, **22**, e408–e416.
- Rost, B. and Sander, C. (1993) Prediction of protein secondary structure at better than 70% accuracy. *J. Mol. Biol.*, **232**, 584–599.
- Sali, A. (1991) *Ph.D. Thesis*. University of London.
- Shen, H.B. and Chou, K.C. (2006) Ensemble classifier for protein fold pattern recognition. *Bioinformatics*, **22**, 1717–1722.
- Shi, J. *et al.* (2001) FUGUE: sequence-structure homology recognition using environment-specific substitution tables and structure-dependent gap penalties. *J. Mol. Biol.*, **310**, 243–257.
- Smith, D. (1989) *SSTRUC: A Program to Calculate Secondary Structural Summary*. Department of Crystallography, Birkbeck College, University of London.
- Vapnik, V. (1995) *The Nature of Statistical Learning theory*. Springer, New York.
- Vapnik, V. (1998) *Statistical Learning Theory*. Wiley, New York, NY.
- Wang, Y. *et al.* (2006) Better prediction of the location of α -turns in proteins with support vector machine. *Proteins Struct. Funct. Bioinformatics*, **65**, 49–54.
- Yang, Z.R. (2004) Biological applications of support vector machines. *Brief. Bioinformatics*, **5**, 328–338.