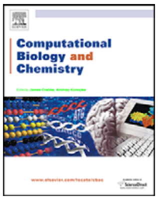
Contents lists available at ScienceDirect

Computational Biology and Chemistry

journal homepage: www.elsevier.com/locate/compbiolchem



Supervised machine learning algorithms for protein structure classification

Pooja Jain[a], Jonathan M. Garibaldi[b], Jonathan D. Hirst[a,*]

[a] School of Chemistry, The University of Nottingham, University Park, Nottingham, NG7 2RD, UK
[b] School of Computer Science and IT, The University of Nottingham, Jubilee Campus, Nottingham, NG8 1BB, UK

ARTICLE INFO

Article history:
Received 13 October 2008
Received in revised form 25 March 2009
Accepted 23 April 2009

Keywords:
Supervised learning
Structure classification
Random forest

ABSTRACT

We explore automation of protein structural classification using supervised machine learning methods on a set of 11,360 pairs of protein domains (up to 35% sequence identity) consisting of three secondary structure elements. Fifteen algorithms from five categories of supervised algorithms are evaluated for their ability to learn for a pair of protein domains, the deepest common structural level within the SCOP hierarchy, given a one-dimensional representation of the domain structures. This representation encapsulates evolutionary information in terms of sequence identity and structural information characterising the secondary structure elements and lengths of the respective domains. The evaluation is performed in two steps, first selecting the best performing base learners and subsequently evaluating boosted and bagged meta learners. The boosted random forest, a collection of decision trees, is found to be the most accurate, with a cross-validated accuracy of 97.0% and F-measures of 0.97, 0.85, 0.93 and 0.98 for classification of proteins to the *Class*, *Fold*, *Super-Family* and *Family* levels in the SCOP hierarchy. The meta learning regime, especially boosting, improved performance by more accurately classifying the instances from less populated classes.

1. Introduction

One of the major aims of structural biology is to classify known protein structures to reflect their structural, evolutionary, and functional relatedness. These relationships are embodied in hierarchical protein classification schemes, such as Structural Classification of Proteins (SCOP) (Murzin et al., 1995). However, there are challenges in hierarchical protein structure classification. To organise protein structures, the elements at the top levels of the hierarchy are defined based on the similarity in structural content and topology. Therefore, these elements are limited (Govindarajan et al., 1999). The lower levels, on the other hand, tend to be dense and overlapping to accommodate the prevalent evolutionary and functional similarity. Human expertise has been indispensable in organising the complex universe of known protein structures. However, despite the experience of the creator(s) of SCOP, for a significant number of proteins there are differences in classification between any two release versions of SCOP and a large number of proteins remain listed under the *Not a True Class* level. There is keen interest in the automation of structure classification (Andreeva et al., 2008; Oakley et al., 2008) to help experts and to cope with the speed with which new structures are being generated.

Protein structure classification can be cast as a problem of data mining, in which a supervised machine learning (ML) algorithm classifies new structures based on what it learns from the classification already available in SCOP. Work on protein structural classification has been ongoing for over a decade using supervised ML algorithms, such as neural networks (Cai et al., 2001; Ding and Dubchak, 2001; Chung et al., 2003; Klein and Delisi, 2004; Vinga et al., 2004; Ie et al., 2005) and support vector machines (SVMs) (Chen and Kurgan, 2007; Shamim et al., 2007; Melvin et al., 2007; Kurgan and Chen, 2007; Gewehr et al., 2007). These studies have reported accuracy in the range of 55–70% in predicting the appropriate SCOP level of a protein.

Sequence and structural comparison methods (Cheek et al., 2004; Rufino and Blundell, 1994) and amalgamations thereof (Çamoğlu et al., 2005; Cheng and Baldi, 2006; Shen and Chou, 2006) have also proved to be successful. In particular, SCOPMap (Cheek et al., 2004) uses sequence comparison (BLAST (Altschul et al., 1990), PSI-BLAST (Altschul et al., 1997), RPS-BLAST (Marchler-Bauer et al., 2002) and COMPASS (Sadreyev and Grishin, 2003)) and structure comparison methods (DALI (Holm and Sander, 1993) and MAMMOTH (Ortiz et al., 2002)) and a manually set similarity threshold to make assignments to super-families. Using similarity scores for a pair of proteins from methods such as CE (Shindyalov and Bourne, 1998), VAST (Madej et al., 1995), DALI, PSI-BLAST and HMMER (Eddy, 1998), a decision tree based ensemble classifier was proposed by Çamoğlu et al., assigning proteins to the existing fam-

[*] Corresponding author. Tel.: +44 115 951 3478; fax: +44 115 951 3562.
E-mail address: jonathan.hirst@nottingham.ac.uk (J.D. Hirst).

1476-9271/$ – see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compbiolchem.2009.04.004

ily, super-family and fold with an accuracy of 83%, 45% and 31%, respectively (Çamog˘lu et al., 2005).

Using a previously published data set (Ding and Dubchak, 2001), Shen and Chou (2006) attempted to assign proteins to one of the 27 fold selected from SCOP. They proposed a weighted voting scheme to combine SVM based classifiers trained independently on features, such as pseudo amino acid composition (Chou, 2001), secondary structure state, hydrophobicity, polarity, polarizability and normalised van der Waals volume. The classification accuracy of 62.1% was shown to be better than other studies (Ding and Dubchak, 2001; Chung et al., 2003). Recently, Zhao et al. have applied ensemble learning to protein structure classification (Zhao et al., 2008). Although such approaches improve the classification accuracy to 75% (Cheng and Baldi, 2006), they remain limited to the well represented SCOP levels in the data set.

SCOP classifies proteins to different structural levels based on the constituent structural domains. By definition, a structural domain is the part of a polypeptide chain that can fold, evolve and function independently of rest of the protein. Therefore, a multidomain protein can be assigned to more than one sub-tree in SCOP, depending on its constituent domains. Multiple classification is not possible with previously reported methods, which do not focus on domain structures. Moreover, the majority of the studies classify a protein structure to one or two of the top four levels (*Class, Fold, Super-Family* and/or *Family*) of the SCOP hierarchy. No effort has yet been made to discover the classification tree, spanning these four levels, for a structure. Therefore, a method that assigns a protein to a known class and to the nested fold, super-family and family is needed. The discovered classification tree could suggest a new fold, super-family or a family level and perhaps could suggest a rearrangement of the existing hierarchy to accommodate newly determined protein structures.

Our approach is different from previous studies in three aspects. Firstly, we use domains to learn the structure classification. Secondly, our approach considers a pair of domains: for one of the domains the structure classification is unknown and for the other it is known. Every domain structure is represented by structural and sequence based properties of the constituent secondary structure elements (SSEs). The higher the similarity among properties, the deeper the predicted common structural level in the SCOP hierarchy for the two domains. Based on these predictions, the structure classification for the known domain can be assigned to the unclassified domain. This similarity-based protein domain classification can be seen as a form of *Entity Resolution*, an established research area in text categorisation and pattern recognition (Bhattacharya and Getoor, 2007). The concept of transferring structure classification is analogous to *annotation transfer*, which has been widely accepted in genomics for the functional annotation of novel genes or proteins (Friedberg, 2006; Hegyi and Gerstein, 2001; Levy et al., 2005; Mistry et al., 2007; Valencia, 2005). Thirdly, we present a systematic assessment of different supervised ML algorithms for the task of hierarchical protein structure classification.

2. Supervised learning

For a classification task, supervised ML algorithms take a training set of instances $\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$ to learn the dependence of a categorical or nominal class label y_i on a vector x_i of measurable features, minimising the error over the entire set. In the present work, we evaluate classifiers from five categories of supervised ML algorithms, namely naïve Bayes learner (naïve Bayes and Bayesian network) the neural networks (SimpleLogistic, Multilayer perceptron and Radial basis function network), the decision tree learners (J48, NBTree, REPTree and SCART), the rule learners (JRip, PART and OneR) and the SVMs, as implemented in the WEKA machine learning package developer version 3.5.7 (Witten and Frank, 2005). Henceforth, these are collectively termed as base learners. Below is an overview of these base learners and the meta learning techniques used. Different textbooks (Quinlan, 1993; Witten and Frank, 2005) and respective publications can be consulted for further details.

A naïve Bayes learner combines Bayesian reasoning with the assumption of independence among the measurable features. It uses standard probability distribution methods to learn relative frequencies of different classes and feature values in the training data to estimate the class probability and the conditional probability distribution of a class given the feature values. The Bayesian network, one of the naïve Bayes learners that we evaluate, is the synergistic combination of Bayes theorem and belief networks (Witten and Frank, 2005) and can be considered as a directed acyclic graph, where features are employed as the nodes and their interdependence as the directional edges. A Bayesian network follows a different approach to estimate the conditional probability distribution, where the relative frequency of each class is represented as a leaf and the instances that a particular leaf covers decide the probability distribution for the class.

A neural network is an interconnected structure of nodes, specialised for parallel learning. The nodes mimic neurons, the fundamental units of a brain. In a neural network, every node receives an input and returns an output. To minimise the difference in the actual output and the desired output, a weight is assigned to the input–output association. Different approaches to calculating weights and different architectures lead to different types of neural networks, such as multi-layer perceptrons (MLPs), typically characterised by one or more layer of hidden neurons between input and output layers, and Radial Basis Function Networks (RBFNet).

A decision tree learner follows a divide-and-conquer strategy for training. To define a node, the most information rich feature is selected. A univariate (single feature) split of the data follows with respect to a predetermined constant for the selected feature. The instances are divided iteratively and the procedure is repeated for every child node, until all of the instances are assigned. The resultant terminal node is called a leaf node, which suggests the classification for all of the instances that reach it. We evaluate decision trees J48 (Quinlan, 1993), REPTree, Naïve Bayes Tree (NBTree) (Kohavi, 1996) and SCART (Kramer and Widmer, 2000; Kramer et al., 2001).

A rule learner induces a set of rules explaining a subset of the training instances, separates these instances, and recursively classifies the remaining training instances by learning more disjoint rules until all of the instances are covered. Such a rule extraction heuristic is termed as the separate-and-conquer strategy (Pagallo and Haussler, 1990). This learning strategy leads to fewer rules for classification as compared to a decision tree. We have evaluated three commonly used rule learners JRip, based on the propositional rule learner RIPPER (Repeated Incremental Pruning to Produce Error Reduction) (Cohen, 1995), PART, PARtial decision Trees (Frank and Witten, 1998), and OneR (Holte, 1993).

SVMs (Vapnik, 1998, 2000) find an optimal hyper-plane or decision boundary in a high-dimensional feature space using a non-linear kernel function to maximise the separation between the classes. We have evaluated the SVM classifier implemented in WEKA that uses sequential minimal optimisation (SMO) algorithm for faster optimisation (Platt, 1999).

We have evaluated meta learners by applying two ensemble learning techniques, Bagging (Breiman, 1996) and AdaBoostM1 (Freund and Schapire, 1995, 1996) as implemented in WEKA. These approaches train a base learner on different samples of the training data, thereby reducing the prediction bias and variance. Bias measures the difference in actual and predicted classification and variance measures the consistency of predicted classification across different samples of the training set (Manning et al., 2008). Bagging learns on bootstrap versions (random sampling with replacement), in which instances are drawn randomly according to the specified class probabilities from the original training set and replaced into a new training sample of the same size as the original training set. The bagged predictions are combined by a majority voting scheme. AdaBoostM1, on the other hand, trains the base learners sequentially and deterministically by re-weighting the misclassified instances from the previous trials to ensure they are classified correctly in the next trials. It uses a weighted voting scheme on predictions from each of the trials to reach the final prediction.

We also evaluate random forest (Breiman, 2001), which is a relatively new technique for ensemble learning. It is a collection of decision trees (e.g., REPTrees, as implemented in WEKA), each of which is trained using the bootstrap sample of the entire data set. As opposed to other decision tree learners, the tree splitting is multivariate and uses a set of m randomly selected features from the input features M, where $m \ll M$. Theoretically, $m = M$ should give the lowest error rate. However, it is counteracted by the high correlation among classification trees, which increases the error rate (Breiman, 2001). For the present evaluation, we have set m to $\log M + 1$ and the number of trees in the forest to 10. Each of these trees is the unpruned low-bias, high variance model. However, an ensemble of these unpruned trees, as is typical of a random forest, reduces variance. Thus, a random forest gives predictions with low bias and variance. For a test instance, the predicted class is that assigned by the majority of the trees in the forest.

3. Methodology

The main motivation for comparing various supervised ML algorithms is to identify the best algorithm for automatic protein structure classification. We aim to classify known protein structures up to the *Family* level within the SCOP hierarchy. To begin with, we develop a one-dimensional representation of three-dimensional domains, built from measurable sequence and structural descriptors (features). To maximise the learning accuracy, one has to ensure that the selected features can represent the structural diversity and evolutionary relationships among protein domains and thus encode the information needed for their classification. We formulate the protein structure classification problem as *"For a pair of domains, learn from its paired profile, the level they share in the SCOP structural hierarchy and predict for the unseen pairs of domains if they share any of the levels learned"*. The top level in the hierarchy, *Class*, groups domains with a general structural architecture having similar type of secondary structure elements (SSEs). *Fold* groups domains that may not share a common evolutionary origin, but the topological arrangements and connections between their SSEs. The *Super-Family* level groups domains having structural and functional similarity, implying divergent evolution, but which do not necessarily have detectable sequence homology. The *Family* level groups domains with definite evidence of common descent and sequence identity of 30% or higher. We have considered five classes (*CL*, *FO*, *SF*, *FA* and *NA*) to represent pairs of domains, which share the same class, fold, super-family, family or none of these levels, respectively. Depending on the level of similarity between the profiled representations for the two domains, the classifier assigns the pair to one of the five classes. The assignment to a class deeper in the SCOP hierarchy is more difficult. However, prediction to a deeper and more specific level in the hierarchy automatically assigns all the level(s) above it in the same classification sub-tree.

3.1. Domain data set

From the ASTRAL compendium (Brenner et al., 2000) the structures for all the single domain proteins classified in the SCOP database version 1.69 were obtained. For these domains, the secondary structure assignments from DSSP (Dictionary of Secondary Structure of Proteins) (Kabsch and Sander, 1983) were used to select the domains consisting of three SSEs in a single polypeptide chain. We considered small domains, primarily, to evaluate a variety of ML algorithms in a reasonable time and with the available computational power. The redundant domains from multi-meric proteins were removed giving a set of 1394 domains and a total of 970,921 pairs of domains. To reduce the redundancy further, domain pairs with greater than 35% sequence

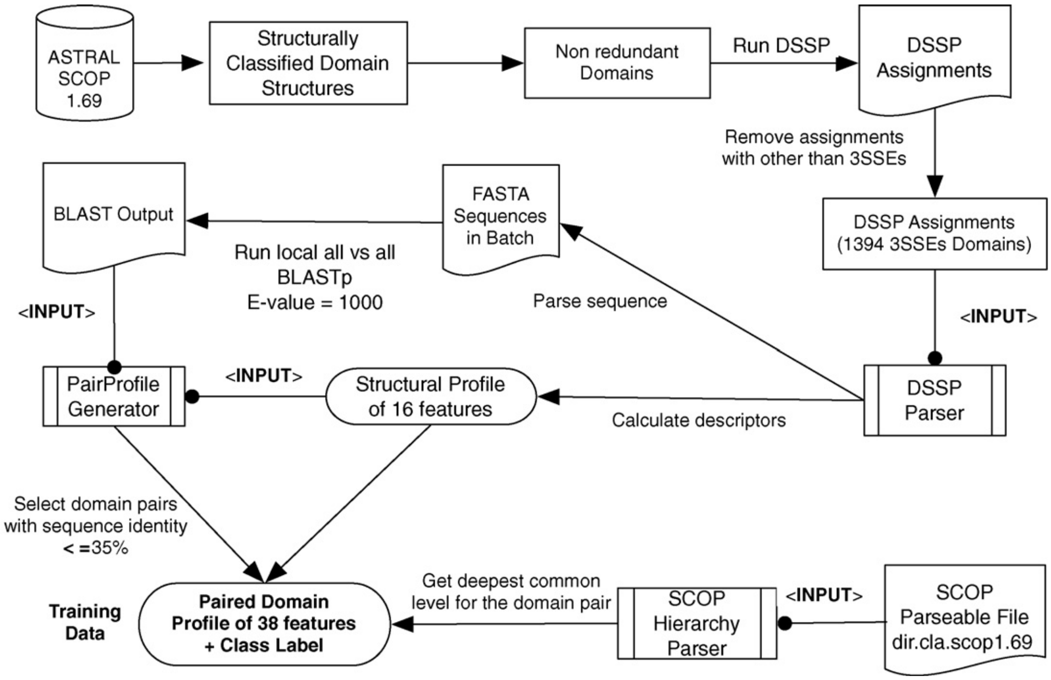

ASTRAL SCOP 1.69
Structurally Classified Domain Structures
Non redundant Domains
Run DSSP
DSSP Assignments
Remove assignments with other than 3SSEs
DSSP Assignments (1394 3SSEs Domains)
FASTA Sequences in Batch
BLAST Output
Run local all vs all BLASTp E-value = 1000
Parse sequence
<INPUT>
<INPUT>
<INPUT>
PairProfile Generator
Structural Profile of 16 features
Calculate descriptors
DSSP Parser
Select domain pairs with sequence identity < =35%
Training Data
Paired Domain Profile of 38 features + Class Label
Get deepest common level for the domain pair
SCOP Hierarchy Parser
<INPUT>
SCOP Parseable File dir.cla.scop1.69

Fig. 1. The process for the selection of domain structures and generation of the training data.

identity were removed. Fig. 1 presents the selection process for the domain structures and their subsequent use in the training set generation. The training set consists of 11,630 instances (4791, 323, 282, 1597 and 4639 instances with *CL*, *FO*, *SF*, *FA* and *NA* labels, respectively), representing domain pairs from 1394 domains with sequence identity of at most 35%. Clearly, the training set is not balanced and is rich in domain pairs sharing a class or a family or those without a common structural level. We termed all these instances the majority class. The instances sharing the same fold and super-family are the minority class. These instances represent 6 Classes, 7 Folds, 8 Super-Families and 24 Families from SCOP.

Each instance in the training set is a pair of domains and consists of 38 measurable features characterising the respective SSEs. The DSSP secondary structure assignments are used to calculate five different structural descriptors using the C_α co-ordinates of the SSE. These descriptors can be grouped into two categories, the pairwise descriptors and the individual descriptors. Two real-valued pairwise descriptors are calculated for every possible pair of the SSEs in a domain: the separation, ρ, and the relative orientation, θ. For any two SSEs i and j, the separation of their centres of mass (COM) is:

$$\rho_{ij} = \sqrt{\sum_{c=X,Y,Z} (\mathrm{COM}_{ci} - \mathrm{COM}_{cj})^2} \tag{1}$$

The relative orientation of the SSEs is the angle measured between the axes that pass through the terminal C_α atoms of the SSEs. For any two SSEs i and j, represented as vectors $\vec{V}_i$ and $\vec{V}_j$ along the respective axis, the relative orientation is:

$$\theta_{ij} = \cos^{-1} \frac{|\vec{V}_i||\vec{V}_j|}{\vec{V}_i \cdot \vec{V}_j} \tag{2}$$

Three individual descriptors for each SSE are calculated. The average solvent accessibility, δ, is the arithmetic mean of the solvent accessibilities assigned by DSSP to each of the amino acids in a SSE. The total number of residues constituting a SSE is the length descriptor, η, and the SSE type, κ, is a binary descriptor: 0 if α-helix, 1 otherwise. These features total $(3 \times 5) \times 2 = 30$. For the paired domains, four values representing the root mean square difference (RMSD) for the distance, orientation, solvent accessibility and length descriptors, calculated as below, increase the number of features to 34.

$$D_{rmsd} = \sqrt{\sum_{n,m>n} (D^a_{nm} - D^b_{nm})^2} \tag{3}$$

where D can be any of the descriptors from separation, orientation, length and average solvent accessibility for the paired domains a and b, and n is the maximum number of SSEs in the paired domains, i.e., 3.

Four additional features characterising sequence-only properties for each of the domains are also considered. These include the polypeptide chain lengths for the two domains, the difference in the chain lengths and the level of sequence identities shared by the two domains as calculated by BLAST. Finally, a class label was assigned from *C* L, *FO*, *SF*, *FA* or *NA* that shows the deepest common SCOP level for the pair. The class label assignment does not differentiate among different sub-levels nested under the four principal SCOP levels considered. Table 1 explains the class label assignment.

Table 1
Assignment of the class labels. For a domain pair assigned to the class label *FA*, each of the paired domains has the common *Class*, *Fold*, *Super-Family* and *Family*, e.g., domain pairs 1–3. Analogously, the domain pairs from 4–6, 7–9, 10–12 are assigned to the class label *SF*, *FO* and *CL*, respectively. The domain pairs having none of the structural levels in common are assigned to the class label *NA*, e.g., the domain pairs 13–15.

Domain pair	Common level				Class label
	Class	Fold	Super-Family	Family	
1	cl1	cl1.fo1	cl1.fo1.sf2	cl1.fo1.sf2.fa4	*FA*
2	cl1	cl1.fo2	cl1.fo2.sf6	cl1.fo2.sf6.fa3	*FA*
3	cl2	cl2.fo3	cl2.fo3.sf4	cl2.fo3.sf4.fa1	*FA*
4	cl2	cl2.fo7	cl2.fo7.sf1	–	*SF*
5	cl5	cl5.fo2	cl5.fo2.sf4	–	*SF*
6	cl4	cl4.fo2	cl4.fo2.sf3	–	*SF*
7	cl4	cl4.fo2	–	–	*FO*
8	cl2	cl2.fo7	–	–	*FO*
9	cl1	cl1.fo1	–	–	*FO*
10	cl1	–	–	–	*CL*
11	cl4	–	–	–	*CL*
12	cl3	–	–	–	*CL*
13	–	–	–	–	*NA*
14	–	–	–	–	*NA*
15	–	–	–	–	*NA*

3.2. Comparison of classifiers

Due to the large difference in the number of parameters to tune the performance of each algorithm, we did not explore the respective parameter spaces and all were trained and tested with their defaults. Our assessment of the 15 different classifiers was planned in two phases. In the first phase, the base learners were evaluated by 10-fold stratified cross-validation, where the entire data set is divided randomly into ten disjoint subsets of equal size and then alternately each subset is used for testing and the other nine sets are used for training. Alternatively, the algorithms were also trained on a 66% random split of the data followed by 10-fold stratified cross-validation, while the remaining portion of the data was used to test the cross-validated model. The former protocol is expected to give the better estimate of predictive accuracy. However, since the base learners may over-fit the minority class instances, we used the latter as a confirmatory test. In the second phase, the best base classifiers from each of the five categories of algorithms were taken forward, to evaluate further through two meta learning regimes, AdaBoostM1 and Bagging to select the best meta learner.

The overall performance of a classifier in terms of classifying all of the instances correctly is generally considered as the accuracy. For the present work, it can be defined as the percentage of domain pairs classified correctly as sharing some level (or having no level in common) in the SCOP hierarchy. Although we report in Table 2 the overall percent accuracy for different classifiers, we do

Table 2
Performance of different base learners in two classification tests. MLP indicates multi-layer perceptron, PolyK and RBFK indicate the SVM trained with the polynomial kernel and the radial basis function kernel, respectively.

Category	Base learners	%Accuracy	
		Cross-validation	66% Split test
Naïve Bayes	Naïve Bayes	62.5	61.9
	Bayes Net	84.0	82.3
Neural network	Simple logistic	78.9	78.8
	MLP	90.6	89.6
	RBFNet	70.5	71.0
Decision trees	DS	51.0	50.6
	J48	93.6	91.8
	NBTree	93.6	92.2
	REPTree	90.8	89.0
	SCART	92.7	91.3
Rule learners	JRip	93.6	92.4
	PART	93.6	92.1
	OneR	69.2	68.3
SVM	PolyK	78.9	78.7
	RBFK	72.0	70.2

not base our selection on the pairwise two sided *t*-test, as this gives a high Type-I error (Dietterich, 1998), i.e., falsely indicating a classifier's performance significant over others when it is not. When a classifier has to learn the multi-categorical classification, especially on an imbalanced data set, as is the case with the present work, the accuracy might be an unrealistic assessment of classifier's performance, due to the correct classification to majority classes. The performance should be assessed using metrics giving an unbiased estimate of classifier's accuracy across all the classes. Therefore, we have taken into account additional performance metrics for individual classes, including the true positive rate (TPR), false positive rate (FPR), F-measure and g-means, which are commonly used to assess the classification of imbalanced data (Ertekin et al., 2007).

For a given class, TPR, also known as the sensitivity or recall, is the ratio of *TP*, the number of true positives (instances correctly classified as belonging to the class), to the total number of positives in the data set, i.e., $TP/(TP+FN)$, where *FN* is the number of false negatives. The FPR is the ratio of the number of false positives (the instances misclassified) to the total number of negatives in the data set, i.e., $FP/(FP+TN)$, where *TN* is the total number of true negatives. The FPR can also be given by $(1-specificity)$, where specificity is the true negative rate (TNR), i.e., $TN/(TN+FP)$. The F-measure, from the field of information retrieval, is useful where the performance is expressible by the per-class precision and recall. It is the weighted harmonic mean of the precision and recall. A balanced F-measure is $(1+\alpha)(Precision \cdot Recall)/\alpha(Precision+Recall)$, where α is set to 1 to weight the precision, i.e., $TP/(TP+FP)$ and recall equally. The TPR and FPR, reflect the discriminatory power of a classifier (Sonego et al., 2008). Additionally, the g-means, i.e., the square root of the product of the sensitivity and the specificity of a classifier is used to assess classification performance for the minority class instances (Ertekin et al., 2007).

For the base learners, the reported accuracy is the averaged accuracy of the 10-fold cross-validation tests on the entire data set. For the 66% split test the accuracy is averaged over the ten runs of 10-fold cross-validation tests on the randomly split 66% training set. The reported performance measures for the meta learners represent the average values over the ten runs of meta learning.

4. Results

4.1. Selection of base learners

The classification accuracies from the 10-fold stratified cross-validation and the 66% split training tests are presented in Table 2. The accuracies of base learners in the split test are within 2% of those observed in the cross-validation test. The g-means for different base learners across the five classes are shown in Fig. 2. For most of the base classifiers, the g-means corresponding to the minority classes, *FO* and *S* F are below 0.8, indicate their inability to handle the imbalanced data set. This is also reflected in the F-measure as shown in Table 3. We chose the best performing base learners, based on the various metrics for all of the five classes. In particular, for g-means, a threshold of 0.7 was set. Bayes Net, multi-layer perceptron and SVM-PolyK were selected. Comparable performance was observed for the rule learners JRip and PART, and the decision tree learners, J48 and NBTree. Thus, all four were considered for the meta learning. The same set of seven best performing base learners can be selected based on the overall accuracy and F-measure too, suggesting a low prediction bias across various measures. This enabled us to select the appropriate base learners for subsequent evaluation as the meta learners.

4.2. Selection of meta learner

In general, the meta learners were more accurate (Table 4) than the respective base learners (Table 3). The performance of the respective meta learners for the minority classes *FO* and *SF* was much improved in terms of TPR and F-measure. This shows the advantage of meta learning for an imbalanced data set. Boosting, in particular, was advantageous. The three decision tree learners (J48, NBTree, random forest) outperformed the rest of the meta learners in classifying minority class instances, bringing the TPR close to 0.8 and 0.9 for *FO* and *SF*, respectively, while maintaining the FPR at zero.

In the 10-fold cross-validation tests, for all of the five classes, the boosted Bayes Net performed better than the respective bagged and base learners, whereas bagged multi-layer perceptron performed better than the respective base and boosted learners (Tables 3 and 4). Boosting did not improve the SVM-PolyK, whereas

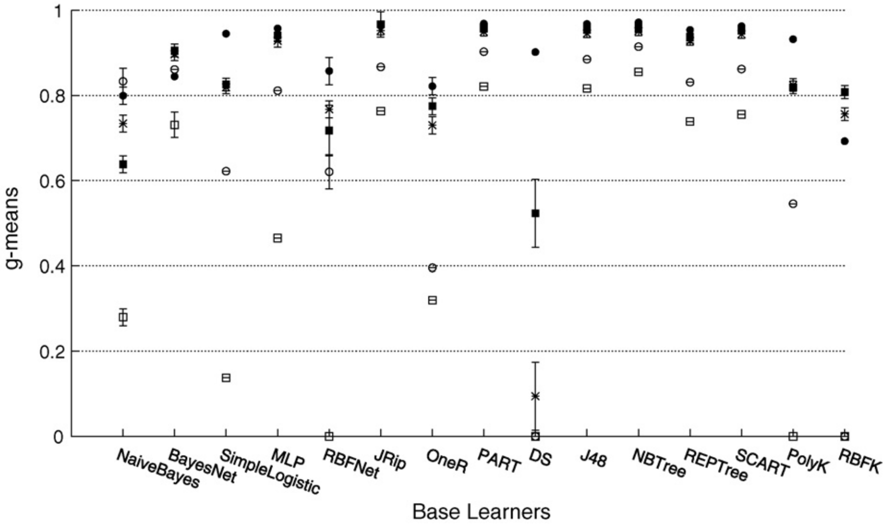


Fig. 2. g-Means for different base classifiers following 10-fold CV. Similar g-means were observed in the 66% split training. A g-means of zero indicates a high true negative rate and failure to detect any true positive instance, the case with the minority class instances mainly FO. (*) *CL*, (□) *FO*, (○) *SF*, (●) *FA*, (■) *NA*. The error bars represent the standard deviations in g-means.

Table 3
The 10-fold cross-validated performance of different base classifiers. TPR: true positive rate; FPR: false positive rate; FM: F-measure.

Base learner	%Accuracy	Class			Fold			Super-Family			Family			NA		
		TPR	FPR	FM	TPR	FPR	FM	TPR	FPR	FM	TPR	FPR	FM	TPR	FPR	FM
Bayes Net	84.0	0.87	0.08	0.88	0.55	0.04	0.39	0.76	0.02	0.58	0.73	0.03	0.78	0.87	0.06	0.89
MLP	90.6	0.93	0.07	0.91	0.22	0.00	0.32	0.67	0.00	0.72	0.93	0.01	0.93	0.94	0.05	0.93
J48	93.1	0.94	0.05	0.93	0.63	0.00	0.70	0.80	0.00	0.81	0.94	0.01	0.95	0.94	0.03	0.95
NB	93.7	0.95	0.05	0.94	0.75	0.01	0.72	0.82	0.00	0.83	0.95	0.00	0.95	0.94	0.03	0.95
JRip	93.6	0.95	0.05	0.94	0.54	0.01	0.60	0.77	0.00	0.80	0.95	0.01	0.94	0.96	0.03	0.96
PART	93.6	0.94	0.05	0.94	0.70	0.00	0.73	0.81	0.00	0.81	0.95	0.00	0.94	0.95	0.03	0.95
SVM-PolyK	78.9	0.90	0.24	0.80	0.00	0.00	0.00	0.30	0.00	0.45	0.90	0.03	0.87	0.72	0.07	0.80

Table 4
The 10-fold cross-validated performance of seven base learners in two meta learning regimes along with the ensemble learner random forest. The performance of random forest without meta learning is shown at the top of the table. The boldface numbers show the best predictions across the five classes, in the order of lowest FPR, highest TPR and highest F-measure (FM).

Meta learner	%Accuracy	Class			Fold			Super-Family			Family			NA		
		TPR	FPR	FM	TPR	FPR	FM	TPR	FPR	FM	TPR	FPR	FM	TPR	FPR	FM
Random forest	96.1	0.98	0.04	0.96	0.71	0.00	0.80	0.85	0.00	0.89	0.97	0.00	0.97	0.96	0.01	0.97
AdaBoostM1																
Bayes Net	90.0	0.90	0.06	0.90	0.61	0.01	0.58	0.87	0.01	0.74	0.93	0.01	0.93	0.91	0.04	0.92
MLP	90.7	0.94	0.08	0.91	0.22	0.00	0.31	0.66	0.00	0.72	0.93	0.01	0.93	0.93	0.09	0.93
J48	96.5	**0.98**	**0.03**	**0.97**	**0.76**	**0.00**	**0.85**	0.89	0.00	0.91	0.97	0.00	0.97	0.97	0.02	0.97
NBTree	96.5	0.98	0.03	0.96	0.75	0.00	0.84	0.89	0.00	0.92	0.97	0.00	0.97	0.97	0.02	0.97
Random forest	**96.7**	**0.98**	**0.03**	**0.97**	**0.75**	**0.00**	**0.85**	**0.90**	**0.00**	**0.93**	**0.97**	**0.00**	**0.98**	**0.97**	**0.01**	**0.97**
JRip	96.2	0.97	0.03	0.96	0.68	0.00	0.78	0.83	0.00	0.89	0.96	0.00	0.97	0.98	0.02	0.97
PART	95.6	**0.98**	**0.03**	**0.98**	0.74	0.00	0.83	0.87	0.00	0.91	**0.97**	**0.00**	**0.98**	0.97	0.02	0.97
SVM-PolyK	78.9	0.90	0.24	0.80	0.00	0.00	0.00	0.30	0.00	0.45	0.89	0.03	0.87	0.72	0.07	0.80
Bagging																
Bayes Net	85.2	0.89	0.07	0.89	0.55	0.03	0.42	0.75	0.02	0.60	0.74	0.03	0.78	0.89	0.06	0.90
MLP	92.2	0.96	0.08	0.93	0.20	0.00	0.32	0.68	0.00	0.77	0.94	0.01	0.94	0.94	0.03	0.94
J48	95.1	0.97	0.04	0.95	0.68	0.00	0.78	0.82	0.00	0.87	0.96	0.01	0.96	0.96	0.02	0.96
NBTree	96.5	**0.98**	**0.03**	**0.97**	**0.76**	**0.00**	**0.84**	0.89	0.00	0.91	0.96	0.00	0.97	**0.96**	**0.01**	**0.97**
Random forest	96.4	0.98	0.04	0.97	0.71	0.00	0.81	0.88	0.00	0.92	**0.97**	**0.00**	**0.98**	0.97	0.02	0.97
JRip	96.1	0.98	0.03	0.96	0.60	0.00	0.71	0.78	0.00	0.86	0.96	0.00	0.96	0.98	0.02	0.98
PART	96.0	0.98	0.04	0.96	0.69	0.00	0.79	0.84	0.00	0.90	0.97	0.00	0.97	0.96	0.02	0.97
SVM-PolyK	79.1	0.90	0.24	0.80	0.00	0.00	0.00	0.31	0.00	0.46	0.89	0.03	0.87	0.73	0.07	0.80

bagging did. For the bagged SVM-PolyK the F-measure for the five classes remained lower than any of the base learners or the meta learners. For JRip, PART and the other two decision tree learners J48 and NBTree, the two meta learning regimes, bagging and boosting performed comparably. Interestingly, the boosted random forest was the best. Similar performance was observed from the bagged and boosted NBTree.

We have also assessed the meta learners in the 66% split test. The overall trend was similar to the 10-fold cross-validation test. However, for rule learners and decision tree learners including random forest, the meta learning regimes performed alike, but with lower precision and recall than in the cross-validation test (data not shown). This could be due to the decrease in the instances available for training.

To assess relative performance reliably, the built-in statistical significance test based on corrected resampled *t*-test (Nadeau and Bengio, 2003) from the WEKA experimenter interface was used. The F-measure was used to base the significance estimate for the pairwise wins or losses for the meta-learners. The null hypothesis tested was that there is no significant difference in the performance of meta learners for all of the five classes considered. Table 5 summarises the differences in the number of significant wins and losses for the meta learners for classification to a given class at a confidence interval of 95%. In the boosted and bagged meta learning regimes, among other meta learners the performance of boosted random forest was found to be statistically significant, thus rejecting the null hypothesis. In general, the next best performing classifier was NBTree. None of the pairwise comparisons found the SVM-PolyK outperforming others even the Bayes Net and MLP. For the boosted meta learning regime the random forest was never superseded by the NBTree, but for the bagged meta learning regime the NBTree performed significantly better than any other for classification to the minority class *FO* and a bit better than the bagged random forest for the majority class *FA*.

Table 5
Relative performance of different meta learners based on the difference in pairwise wins and losses estimated statistically based on F-measure with a 95% confidence interval.

Meta learner	Wins–losses				
	CL	*FO*	*SF*	*FA*	*NA*
Boosting					
Random forest	3	4	5	4	3
NBTree	3	4	4	4	3
J48	3	4	4	3	3
JRip	3	−1	0	1	3
PART	3	4	2	3	3
MLP	−4	−5	−4	−4	−3
Bayes Net	−4	−3	−4	−4	−5
SVM-PolyK	−7	−7	−7	−7	−7
Bagging					
Random forest	5	3	6	6	3
NBTree	5	7	5	5	4
J48	−1	3	1	0	−1
JRip	4	−1	0	1	7
PART	2	3	3	3	2
MLP	−3	−5	−3	−3	−3
Bayes Net	−5	−3	−5	−5	−5
SVM-PolyK	−7	−7	−7	−7	−7

5. Discussion

Automation of structure classification not only minimises human intervention, but might also avoid errors in classification accumulated due to the multiple tools and techniques employed currently (Andreeva et al., 2008). As in previous work, we have used a one-dimensional representation of three-dimensional protein structures, however, using a different set of structure and sequence based properties, to train a supervised ML algorithms for automatic structural classification. Employing the concept of entity resolution, we have proposed the transfer of structure classification from one structural domain to another, guided by the supervised ML algorithm. We have evaluated 15 base learners, including popular methods, such as neural networks and SVMs, and some methods previously untested in protein structure classification, such as rule learners and decision tree learners.

We studied the performance of the best performing base learners through the ensemble learning and observed an improved accuracy. We have shown the importance of ensemble learning for dealing with an imbalanced data set. It improves the generalisation by the constructive combination of base learners trained on the different parts of the data. The decision tree learners J48, NBTree and random forest are the most successful in predicting the common SCOP level for a pair of protein domains consisting of three SSEs. Generally, a meta learner needs a long time to train, but once trained, its testing on the unseen data is quick. On average, boosted random forest was faster to train than J48 and NBTree meta learners (Table 6). We, therefore, propose random forest suitable for protein structure classification.

Although the overall prediction accuracy of base learners and the respective meta learners was similar (Tables 3 and 4), the latter reduced the classification bias to majority class instances and improving the TPR and FPR for minority classes. Further, the better performance of boosted as compared to the bagged meta learners can be explained by the sampling with replacement involved in bagging that would have removed some of the instances important for defining the classification. Boosting not only preserves such instances, but also weights the misclassification of instances, thereby minimising the misclassification.

Based on the class-wise performance of meta-learners as reported in Table 4, the results from the significance test based on F-measure shown in Table 5 and the time required for the training (Table 6), we select random forest as the best performing meta learner. It classifies protein structures sharing any of the top four structural levels in the SCOP hierarchy and segregates those that do not share any. In particular, the boosted random forest gives the highest overall accuracy (96.7%) and the class-wise highest TPR, lowest FPR and highest F-measure (as is bold faced in Table 4). Also the random forest is quicker to train than the competing NBTree and J48 decision tree learners.

Table 6
Training time for different meta learners in boosted 10-fold cross-validation test performed on a Power Mac G5 with 2.3GHz PowerPC dual core CPU running Mac OS X 10.4 (Tiger).

Meta learner	Average training time for meta learners (min)				
	Class	Fold	Super-Family	Family	*NA*
Bayes Net	39	36	37	37	37
MLP	1145	949	1131	947	1112
J48	207	154	205	143	216
NBTree	5189	5175	5018	5344	5195
Random forest	67	256	235	50	96
JRip	742	728	765	742	740
PART	747	746	734	730	755
SVM-Poly	359	360	350	179	357

6. Conclusions

The present work addresses three issues for a prototypical set of protein domains containing three SSEs. Firstly, is there a way to represent the three-dimensional structure of proteins in one dimension while preserving the functional and structural complexity of proteins? Secondly, is it possible to use this one-dimensional representation to help organise the known protein universe? Thirdly, how is it possible to automate such an organisation for the increasing pool of known protein structures? We propose a one-dimensional profile of the descriptors, which can preserve the structural and evolutionary relatedness of a pair of domains. A ML algorithm learns this profile and predicts if the paired domain share a structural level. Random forest is found to be the most accurate classifier.

The descriptors used are not correlated significantly with each other and some of them may be less informative. The impact of discarding such features through established feature selection approaches to reduce the time and cost for protein structure classification is a future avenue to pursue. The domain data set used contains proteins up to 300 residues in length. It would be interesting to utilise domains with more than 4 SSEs to cover larger proteins and evaluate generalisation of the proposed approach. We also plan to test our strategy on a set of domain pairs identified by the unique SCOP identifiers for the level they share. It would then be possible to identify a new sub-tree for a protein in the SCOP hierarchy and reflect the possible rearrangements required in SCOP.

Acknowledgements

We acknowledge the HPC facility at the University of Nottingham. We thank Craig Bruce for technical assistance. PJ is supported by the BIOPTRAIN Marie Curie Action MEST-CT-2004-7597 under the Sixth Framework Programme of the European Community.

References

Altschul, S.F., Gish, W., Miller, W., Myers, E.W., Lipman, D.J., 1990. Basic local alignment search tool BLAST. J. Mol. Biol. 215, 403–410.

Altschul, S.F., Madden, T.L., Schaffer, A.A., Zhang, J., Zhang, Z., Miller, W., Lipman, D.J., 1997. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Res. 25, 3389–3402.

Andreeva, A., Howorth, D., Chandonia, J.M., Brenner, S.E., Hubbard, T.J., Chothia, C., Murzin, A.G., 2008. Data growth and its impact on the SCOP database: new developments. Nucleic Acids Res. 36, D419–D425.

Bhattacharya, I., Getoor, L., 2007. Collective entity resolution in relational data. ACM Trans. Knowl. Discov. Data 1, 5.

Breiman, L., 1996. Bagging predictors. Mach. Learn. 24, 123–140.

Breiman, L., 2001. Random forests. Mach. Learn. 45, 5–32.

Brenner, S.E., Koehl, P., Levitt, M., 2000. The ASTRAL compendium for protein structure and sequence analysis. Nucleic Acids Res. 28, 254–256.

Cai, Y., Liu, X., Xu, X., Zhou, G., 2001. Support vector machines for predicting protein structural class. BMC Bioinformatics 2, 3.

Çamoğlu, O., Can, T., Singh, A.K., Wang, Y., 2005. Decision tree based information integration for automated protein classification. J. Bioinform. Comput. Biol. 3, 717–742.

Cheek, S., Qi, Y., Krishna, S.S., Kinch, L.N., Grishin, N.V., 2004. SCOPmap: automated assignment of protein structures to evolutionary superfamilies. BMC Bioinformatics 5, 197–222.

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

Cheng, J., Baldi, P., 2006. A machine learning information retrieval approach to protein fold recognition. Bioinformatics 22, 1456–1463.

Chou, K., 2001. Prediction of protein cellular attributes using pseudo-amino acid composition. Proteins: Struc. Funct. Genet. 43, 246–255.

Chung, I., Huang, C., Shen, Y., Lin, C., 2003. Recognition of structure classification of protein folding by NN and SVM hierarchical learning architecture. In: ICANN. pp. 1159–1167.

Cohen, W.W., 1995. Fast effective rule induction. In: ICML. Morgan Kaufmann, pp. 115–123.

Dietterich, T.G., 1998. Approximate statistical tests for comparing supervised classification learning algorithms. Neural Comput. 10, 1895–1923.

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

Eddy, S.R., 1998. Profile hidden markov models. Bioinformatics 14, 755–763.

Ertekin, S., Huang, J., Bottou, L., Giles, C.L., 2007. Learning on the border: active learning in imbalanced data classification. In: CIKM. ACM, pp. 127–136.

Frank, E., Witten, I.H., 1998. Generating accurate rule sets without global optimization. In: ICML. Morgan Kaufmann, pp. 144–151.

Freund, Y., Schapire, R.E., 1995. A decision-theoretic generalization of on-line learning and an application to boosting. In: ECCLT. Springer-Verlag, pp. 23–27.

Freund, Y., Schapire, R. E., 1996. Experiments with a new boosting algorithm. In: ICML. Morgan Kaufmann, pp. 148–156.

Friedberg, I., 2006. Automated protein function prediction—the genomic challenge. Brief Bioinform. 7, 225–242.

Gewehr, J.E., Hintermair, V., Zimmer, R., 2007. AutoSCOP: automated prediction of scop classifications using unique pattern-class mappings. Bioinformatics 23, 1203–1210.

Govindarajan, S., Recabarren, R., Goldstein, R.A., 1999. Estimating the total number of protein folds. Proteins: Struct. Funct. Bioinform. 35, 408–414.

Hegyi, H., Gerstein, M., 2001. Annotation transfer for genomics: measuring functional divergence in multi-domain proteins. Genome Res. 11, 1632–1640.

Holm, L., Sander, C., 1993. Protein structure comparison by alignment of distance matrices. J. Mol. Biol. 233, 123–138.

Holte, R.C., 1993. Very simple classification rules perform well on most commonly used datasets. Mach. Learn. 11, 63–90.

Ie, E., Weston, J., Noble, W.S., Leslie, C., 2005. Multi-class protein fold recognition using adaptive codes. In: ICML. ACM, pp. 329–336.

Kabsch, W., Sander, C., 1983. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. Biopolymers 22, 2577–2637.

Klein, P., Delisi, C., 2004. Prediction of protein structural class from the amino acid sequence. Biopolymers 25, 1659–1672.

Kohavi, R., 1996. Scaling up the accuracy of Naive-Bayes classifiers: a decision-tree hybrid. In: ICKDDM. AAAI Press, pp. 202–207.

Kramer, S., Pfahringer, B., Widmer, G., Groeve, M.D., 2001. Prediction of ordinal classes using regression trees. Fundam. Inf. 47, 1–13.

Kramer, S., Widmer, G., 2000. Relational Data Mining: Inducing Classification and Regression Trees in First Order Logic. Springer-Verlag.

Kurgan, L., Chen, K., 2007. Prediction of protein structural class for the twilight zone sequences. Biochem. Biophys. Res. Commun. 357, 453–460.

Levy, E.D., Ouzounis, C.A., Gilks, W.R., Audit, B., 2005. Probabilistic annotation of protein sequences based on functional classifications. BMC Bioinformatics 14, 302.

Madej, T., Gibrat, J.F., Bryant, S.H., 1995. Threading a database of protein cores. Proteins: Struct. Funct. Bioinform. 23, 356–369.

Manning, C.D., Raghavan, P., Schütze, H., 2008. Introduction to Information Retrieval. Cambridge University Press.

Marchler-Bauer, A., Panchenko, A.R., Shoemaker, B.A., Thiessen, P.A., Geer, L.Y., Bryant, S.H., 2002. CDD: a database of conserved domain alignments with links to domain three-dimensional structure. Nucleic Acids Res. 30, 281–283.

Melvin, I., Ie, E., Kuang, R., Weston, J., Noble, W.S., Leslie, C., 2007. SVM-fold: a tool for discriminative multi-class protein fold and superfamily recognition. BMC Bioinformatics 8, 2.

Mistry, J., Bateman, A., Finn, R.D., 2007. Predicting active site residue annotations in the pfam database. BMC Bioinformatics 8, 298.

Murzin, A.G., Brenner, S.E., Hubbard, T., Chothia, C., 1995. SCOP: a structural classification of proteins database for the investigation of sequences and structures. J. Mol. Biol. 247, 536–540.

Nadeau, C., Bengio, Y., 2003. Inference for the generalization error. Mach. Learn. 52, 239–281.

Oakley, M.T., Barthel, D., Bykov, Y., Garibaldi, J.M., Burke, E.K., Krasnogor, N., Hirst, J.D., 2008. Search strategies in structural bioinformatics. Curr. Prot. Peptide Sci. 9, 260–274.

Ortiz, A.R., Strauss, C.E.M., Olmea, O., 2002. MAMMOTH (matching molecular models obtained from theory): an automated method for model comparison. Protein Sci. 11, 2606–2621.

Pagallo, G., Haussler, D., 1990. Boolean feature discovery in empirical learning. Mach. Learn. 5, 71–99.

Platt, J.C., 1999. In Advances in Kernel Methods—Support Vector Learning: Fast Training of Support Vector Machines Using Sequential Minimal Optimization. MIT Press.

Quinlan, J.R., 1993. C4.5: Programs For Machine Learning. Morgan Kaufmann.

Rufino, S.D., Blundell, T.L., 1994. Structure-based identification and clustering of protein families and superfamilies. J. Comput. Aided Mol. Des. 8, 5–27.

Sadreyev, R.I., Grishin, N.V., 2003. COMPASS: a tool for comparison of multiple protein alignments with assessment of statistical significance. J. Mol. Biol. 326, 317–336.

Shamim, M.T.A., Anwaruddin, M., Nagarajaram, H.A., 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.

Shen, H., Chou, K., 2006. Ensemble classifier for protein fold pattern recognition. Bioinformatics 22, 1717–1722.

Shindyalov, I.N., Bourne, P.E., 1998. Protein structure alignment by incremental combinatorial extension (CE) of the optimal path. Protein Eng. 9, 739–747.

Sonego, P., Kocsor, A., Pongor, S., 2008. ROC analysis: applications to the classification of biological sequences and 3D structures. Brief Bioinform., Bbm064+.

Valencia, A., 2005. Automatic annotation of protein function. Curr. Opin. Struct. Biol. 15, 267–274.

Vapnik, V.N., 1998. Statistical Learning Theory, 2nd edition. John Wiley & Sons.

Vapnik, V.N., 2000. The Nature of Statistical Learning Theory, 2nd edition. Springer-Verlag, New York.

Vinga, S., Gouveia-Oliveira, R., Almeida, J.S., 2004. Comparative evaluation of word composition distances for the recognition of SCOP relationships. Bioinformatics 20, 206–215.

Witten, I.H., Frank, E., 2005. Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann.

Zhao, X.M., Li, X., Chen, L., Aihara, K., 2008. Protein classification with imbalanced data. Proteins: Struct. Funct. Bioinform. 70, 1125–1132.