Using Rotation Forest for Protein Fold Prediction Problem: An Empirical Study

Abdollah Dehzangi, Somnuk Phon-Amnuaisuk, Mahmoud Manafi, and Soodabeh Safa

Center of Artificial Intelligence and Intelligent computing, Faculty of Information Technology,
Multi Media University, Cyberjaya, Selangor, Malaysia
abdollah.dehzangi07@mmu.edu.my, somnuk.amnuaisuk@mmu.edu.my
mahmoud.manafi09@mmu.edu.my, soodabeh.safa07@mmu.edu.my

Abstract. Recent advancement in the pattern recognition field has driven many classification algorithms being implemented to tackle protein fold prediction problem. In this paper, a newly introduced method called Rotation Forest for building ensemble of classifiers based on bootstrap sampling and feature extraction is implemented and applied to challenge this problem. The Rotation Forest is a straight forward extension of bagging algorithms which aims to promote diversity within the ensemble through feature extraction by using Principle Component Analysis (PCA). We compare the performance of the employed method with other Meta classifiers that are based on boosting and bagging algorithms, such as: AdaBoost.M1, LogitBoost, Bagging and Random Forest. Experimental results show that the Rotation Forest enhanced the protein folding prediction accuracy better than the other applied Meta classifiers, as well as the previous works found in the literature.

Keywords: Protein Fold Prediction Problem, Ensemble Classifier, Rotation Forest, Feature Extraction, Principal Component Analysis, Decision Tree, C4.5, Random Forest, AdaBoost.M1, LogotBoost, Bagging.

1 Background

Prediction of the tertiary structure of a protein from its primary structure is a challenging task in bioinformatics and biological science. Recently, due to tremendous advancements in pattern recognition, many classifiers have been implemented and applied to challenge this task. In this paper, Rotation Forest, as a newly proposed method for building an ensemble of classifiers employed to tackle the protein fold prediction problem.

The Rotation Forest, by Rodriguez and his co-workers [1], is based on bagging algorithm [2] that aims to build a more accurate and diverse classifier. Rotation Forest uses bootstrap samples of training dataset to train a group of decision trees as well as bagging, but dissimilar to bagging, to reinforce diversity within classifiers ensemble; it splits the feature set to randomly K subset, then runs *Principal Component Analysis (PCA)* on each of them, and finally rebuilds the feature set of N linear extracted features by combining all principle components. In this way it transforms

C. Pizzuti, M.D. Ritchie, and M. Giacobini (Eds.): EvoBIO 2010, LNCS 6023, pp. 217–227, 2010.

the feature set to the new $M(where M \leq N)$ dimensional feature spaces. This process is repeated to extract new set of features to train each of the base learners in parallel. At last, Rotation Forest combines the results of all base classifiers using majority voting.

Rotation Forest has been widely applied for different benchmarks and in many cases outperformed other Meta Classifiers such as Adaboost.M1 [3], or other classifiers such as Support Vector Machine which is considered as the state-of-the-art in machine learning ([1], [4], and [5]). To the best of our knowledge, Rotation Forest has never been applied to deal with the protein folding task. Experimental results demonstrated that the Rotation Forest enhanced the prediction accuracy as well as reducing time consumption of the classification task better than the previous related works found in the literature.

One of the most important factors that affect the performance of the Rotation Forest is the number of base classifiers. Therefore, in this paper, to study the sensitivity of the Rotation Forest to the number of base classifiers for the protein fold prediction problem, six different numbers of base classifiers in the range between 10 and 200 were employed (10, 20, 50, 100, 150, and 200). Finally, the Rotation Forest compared with the best-of-the-shelf Meta classifiers that based on boosting and bagging methods, namely: *Multi Class Adaptive Boosting (AdaBoost.M1), LogitBoost, Random Forest (RF) and Bagging* which demonstrated better results compared to other similar methods ([1], [4], [6] and [7]).

Recently, many efforts have been made to challenge the protein fold prediction problem ([8], [9], [10], [11], [12], and [13]). Most of the classification methods, used for this task were based on Artificial Neural Network (ANN) ([14], [15], [16], and [17]) and Support Vector Machine (SVM) ([18], [19], [20], and [21]). In 2001, Ding and Dubchak used three SVM based multi-class classification methods *(one-versus-others (OvO), unique one-versus others (uOvO), and all-versus-all (AvA)*), with six feature groups named: *Composition of amino acids (C), Predicted secondary structure (X), polarity (P), polarizability (V), hydrophobicity (H) and van der vaals volume (V)* [23]. They reported 56% prediction accuracy using the AvA SVM.

Motivated by the work of Ding and Dubchak [22], Bologna and Appel [14] used ensemble of four-layer *Discretized Interpretable Mulri Layer Perceptron (DIMLP)* trained with the dataset produced by Ding and Dubchak. Different to Ding and Dubchak, in their work, each classifier learned all folds simultaneously. To the best of our knowledge, they reported the highest prediction accuracy (61.1%) using same set of features introduced by Dubchak and her co-workers [23].

NNs and SVMs classifiers used again by Chung and his co-workers as basic building blocks of two-level classifier for the protein folding task. In their work, each NN or SVM was a multi-class classifier [24]; hence, the number of classifiers that they used compared to other works had been greatly reduced. In their work, the common and most popular NN based models with a single hidden layer name: *Multi Layer Perceptron (MLP), Radial Basis Function Network (RBFN), and General Regression Neural Network (GRNN*) were used. However, in Chung and his co-workers and also in their previous works, it was observed that the model constructed by using neural networks and SVMs, perform badly due to the imbalanced proportion of the data which caused high rate of false positive error.

To address this problem, Nanni used non-density-based Fisher's linear classifier (FLC) and an ensemble of Hyper-plane K-Nearest Neighbor classifier (HKNN) [12]. FLC were used to find the linear discrimination function between the classes in the dataset by minimizing the errors in the least-squares sense, and HKNN were used to find a decision surface, by separating different classes of the data, in input space. However, HKNN as a kind of K-Nearest Neighbor (KNN) (Instance Based Learner) based method, suffers from curse of dimensionality while dealing with small dataset contains high amount of features [25].

To conquer inefficiencies of the mentioned methods, and also to reduce computational complexity of the protein fold classification task, Krishnaraj and Reddy employed Boosting approaches as kind of Meta classifiers to tackle the protein fold prediction problem [26]. They employed the AdaBoost.M1 [3] and the LogitBoost [6] to tackle this task. Boosting approaches and generally bootstrap sampling based classifiers avoid false positive error and build robust prediction rules by combining weak learners [3]. They reported comparable prediction accuracy in dramatically lower time complexity (60.3% compared to 61.1% achieved by Bologna and Appel [14]) with other works have been conducted in the literature. Despite all the advantages of the boosting algorithms, they suffer from over-fitting problem while dealing with noisy and high dimensional datasets [27].

Inspired by Krishnaraj and Reddy and in order to exploit the merits of Meta classifiers, we employed the Rotation Forest which illustrated better performance for different benchmarks compared to the other Meta classifiers ([1], [4], and [28]). As like as the Random Forest [7], the Rotation Forest overcome the over-fitting problem by providing a proper method to approximate missing data when dealing with noisy data or in case which large numbers of data are missing ([1] and [7]). Results showed that the Rotation Forest outperformed previous methods developed in the literature for the protein fold prediction problem.

The rest of this paper is organized as follows: in section (2), we introduced the Rotation Forest, how it works and tools which were used in this experiment. In section (3), we introduced the dataset and the features that used in this study. Section (4), concerned about the results and discussion achieved and finally followed by section (5), where the conclusions and future works were explained.

2 Rotation Forest

The Rotation Forest is a recently proposed method based on bootstrap sampling and Principal Component Analysis (PCA) [29]. It builds a group of independent trained decision trees to build an ensemble of classifiers in a parallel manner [1]. Rotation Forest is formulated based on the Random Forest idea [7]. The base classifiers independently built decision trees, but instead of using decision trees for random set of features, each tree in the Rotation Forest is trained on the whole set of dataset in a rotated feature space. It splits feature set (total N features) randomly into K (K is the parameter of the algorithm) subsets and then applied principal component analysis separately to each subset. Finally, based on all principal components the data is transformed linearly into new feature space and make new set of ($M \leq N$ in case where some of Eigen Values are zero [1]) linear feature set by combining all K transformed feature subsets [4].

In Rotation Forest as in the bagging algorithm, bootstrap samples are taken as the training set for the individual classifiers [30]. It performs transformation of feature set for each of the base classifiers, trains each classifier with a boot strap sample of train dataset and transformed feature set, and finally combines all independent base classifiers by using majority voting. In the Rotation Forest classifier, diversity within the classifier ensemble and individual prediction accuracy of the base learners are considered, simultaneously. In this method, diversity is enhanced through feature extraction for each base classifier better than the Random Forest which just uses feature selection to encourage diversity within ensemble classifier [7]; and individual accuracy is also pursued by maintaining all principal components and also using whole dataset to train each base classifier [4].

One of the useful characteristics of the Rotation Forest is that it can be used with almost any base classifier which makes it more flexible than the Random Forest which is capable to be used with Decision Trees as base classifier [7]. Therefore, a lot of possible improvements and modifications can be considered in the Rotation Forest [1]. However, in this paper, decision trees were chosen because of its sensitivity to the rotation of the feature axe.

Data mining toolkit WEKA (Waikato Environment for Knowledge Analysis) version 3.6.0 is used for the classification. WEKA is an open source toolkit and it consists of a collection of machine learning algorithms for solving data mining problems [30]. In this experiment, J48 (WEKA's own version of C4.5 [31]) decision tree algorithm is used as a base classifier. C4.5 is an algorithm used to generate a decision tree.

C4.5 uses the fact that each attribute of the data can be used to make a decision that splits the data into smaller subsets. C4.5 examines the normalized information gain (difference in entropy) that results from choosing a feature for splitting the data [31].

$$SplitInfo_x T = -\sum_{i=1}^{n} \frac{T_i}{T} \log_2 \frac{T_i}{T} \quad (1)$$

$$GainRatio_x(T) = \frac{Gain_x(T)}{SplitInfo_x T} \quad (2)$$

Where $SplitInfo_x T$ represents the potential information provided by dividing dataset, T, into n partition corresponding to the outputs of attributes x, and $Gain_x(T)$ is how much gain would achieve by branching on x.

3 Dataset and Features

To compare our results with the previous work have done by the literature, we used the dataset introduced by Ding and Dubchak [22]. This dataset contains a train and a test dataset. The training dataset comprises of 313 protein belong to the 27 most populated protein folds in *Structural Classification of Protein (SCOP)* protein databank [32], [33]. Each fold contains seven or more proteins. The dataset represents all major structural classes (α, β, α/β, and α + β). The original test dataset is based on the *Protein Data Bank (PDB)* protein databank [34]; it is also developed by the authors of the SCOP database. This dataset contains 385 proteins. Over 90% of our data in the test set has less than 20% sequential similarity with the proteins in test set. Among these proteins, two proteins (2SCMC and 2GPS) in the training dataset and two proteins (2YHX_1 and 2YHX_2) in the testing dataset excluded due to insufficient sequence information. As a result, there are 311 and 383 proteins remain respectively in training and testing dataset.

In this paper, six feature groups were introduced by Dubchak and her co-workers were used [23]. These feature groups were extracted from the proteins amino acid-sequence based on physical, chemical and physiochemical properties of amino acids, named: *amino acids composition (C), predicted secondary structure based on normalized frequency of a-helix residue (S), hydrophobicity (H), normalized Van Der Waals volume (V), polarity (P), and polarizability (Z).* In particular, the first feature represents a vector of the percentage composition of the 20 amino acids in the sequence. The other feature vectors properties are based on three descriptors: composition, percent composition of three constituents (polar, neutral and hydrophobic residues); transition, the transition frequencies (polar to neutral, neutral to hydrophobic, etc.): and distribution, the distribution pattern of constituents (where the first residue of a given constituent is located, and where 25%, 50%, 75%, and 100% of that constituent are contained). Therefore, there are 20 features in composition feature vector and 21 features for other feature vectors. More detail can be found in the literature ([11], [22], and [35]). The length of the amino acid plays an important role in the protein folding task ([14], [35], and [36]. Thus, it is included in every combination of feature groups for experiments.

4 Results and Discussion

The proposed method was evaluated for eleven different combinations of feature groups compared to six combinations of the feature groups used by Ding and Dubchak [22], and Krishnaraj and Reddy [26]. New combinations of the feature groups were applied to find proper combination of features and also investigate the effectiveness of each feature group to the achieved prediction accuracy. In addition, the length of proteins was also added to the all combinations of the feature groups due to its discriminatory information ([11], and [14]).

To study the sensitivity of the Rotation Forest to the number of base classifiers, the employed method with six different numbers of base classifiers in range between 10 and 200 were used for the applied dataset (10, 20, 50, 100, 150, and 200). As shown in Table.1, the best result was achieved by applying the Rotation Forest with 100 base classifiers to the combination of the all feature groups, and the lowest prediction accuracy was obtained by using 10 base classifiers. As we can see in Table.1, by raising the number of base classifiers from 10 to 100, the prediction accuracy of the Rotation Forest also increased significantly, but differences in prediction accuracy between 100, 150 and 200 was nontrivial. Therefore, using 100 base classifiers can be addressed to the future works as an appropriate number of the base classifiers for the applied dataset or more generally for the protein fold prediction task (in similar cases).

According to the results, by using the Rotation Forest with 100 base classifiers, we achieved a 62.4% prediction accuracy which is 1.3% higher than the result reported by Bologna and Appel [14] and 2.1% higher than Nanni [12] and Krishnaraj and Reddy [26] (Table.2). We also achieved a 56.9% prediction accuracy, using the Rotation Forest with 50 base classifiers by employing the composition of amino acid feature group (20-dimensional feature group) which is slightly better than the result achieved by Ding and Dubchak [22] using the AvA SVM and the combination of four feature groups (*Amino Acids Composition, Predicted Secondary Structure, Hydrophobicity, Polarity*).

As shown in Table.1, the Rotation Forest classifier was capable of achieving to the high prediction accuracy depends on the using the appropriate number of base classifiers. The computational complexity of this method was also crucially depended on the number of base classifiers. Therefore, using the Rotation Forest classifier with appropriate number of the base classifiers can achieve to the high prediction accuracy as well as reducing the computational complexity compared to the SVM or ANN based classifiers ([1] and [4]). Despite using PCA algorithms as a feature extraction approach, having parallel structure and using simple and fast base learner (C4.5); made the Rotation Forest classifier as fast as the other Meta classifiers that were based on boosting algorithm (AdaBoos.M1 and LogitBoost). In this paper, the highest result achieved by using 100 base classifiers for Rotation Forest in a comparable computational complexity to the AdaBoost.M1 using the same number of base classifiers ([26]).

Table 1. Comparison of the results achieved (in percentage) by using the Rotation Forest with six different numbers of base classifiers for eleven combinations of the feature groups

Number of Base Classifiers	C	CS	CSV	CSZ	CSP	CSH	CSH V	CSH P	CSH PV	CSH PZ	CSHP ZV
10	50.1	54.3	54.8	55.9	53.5	54.0	53.0	54.8	52.0	54.3	50.1
20	54.3	57.2	55.1	55.4	56.9	56.9	57.2	55.9	56.9	59.0	58.0
50	56.9	59.3	**60.1**	60.1	59.5	60.1	60.6	60.6	60.0	60.6	60.0
100	56.7	58.0	60.8	60.3	59.3	60.6	58.8	60.8	58.7	61.4	**62.4**
150	56.7	60.1	61.1	60.3	**62.1**	60.6	59.5	61.4	61.4	60.3	59.8
200	56.7	60.6	59.8	57.4	61.6	60.8	59.8	60.3	60.8	61.1	**62.3**

The other remarkable result achieved by applying the Rotation Forest for the combination of three feature groups (*Composition of Amino Acid, Predicted Secondary Structure and Polarity feature groups in addition to the length feature*). We achieved 62.1% prediction accuracy, which was 1% higher than the result reported by Bologna and Appel [14] for the independent test dataset.

Table 2. Results achieved by using the Rotation Forest (in percentage) compared to the results achieved by the related works found in the literature for the protein fold prediction problem

[22]	OvO (SVM)	C+S+H	45.2
[22]	Unique OvO (SVM)	C+S+H	51.1
[22]	AvA(SVM)	C+S+H+P+Z+V	56.4
[24]	MLP-Based HLA	C+S	48.6
[24]	RBFN-Based HLA	C+S+H+P+Z+V	56.4
[24]	SVM-Based HLA	C+S+H+P+Z+V	53.2
[26]	AdaBoost.M1	C+S+H	58.2
This Paper	Rotation Forest (150 Decision Trees)	C+S+H	62.1
[26]	LogitBoost	C+S+H+P+V	60.3
[14]	DIMLP	C+S+H+P+Z+V	61.1
[10]	HKNN	C+R+H+P+Z+V	57.4
[12]	RS1_HKNN_K125	C+S+H+P+Z+V	60.0
[12]	RS1_KHNN_K25	C+S+H+P+Z+V	60.3
[11]	BAYESPROT	C+S+H+P+Z+V	58.8
[9]	MOFASA	C+S+H+P+Z+V	60.0
[8]	ALH	C+S+H+P+Z+V	60.8
[38]	RBF Majority voting Fuse	C+S+H+P+Z+V	49.7
[38]	RBF Bayesian Fuse	C+S+H+P+Z+V	59.0
This Paper	Rotation Forest (100 Decision Trees)	C+S+H+P+Z+V	***62.4***

In a different task, the employed method compared to the other Meta classifiers, such as the AdaBoost.M1 that is declared to be the best-of-the-shelf Meta classifier [1], [27], the Logitboost that has been successfully applied for different tasks [6], the Bagging as one of the most popular Meta classifiers has been applied for different machine learning tasks, and the Random Forest, recent modified version of the bagging that showed remarkable results compared to the other meta classifiers ([7], and [37]). Each Meta classifier was tested with the combination of the all feature groups.

For all of Meta classifiers default parameters applied except for the base classifiers and the number of base classifiers. In this paper, the J4.8 and the Decision Stump were respectively employed as the base learners for the AdaBoost.M1 and the LogitBoost based on the experiment were conducted by Krishnaraj and Ready [26]. The J4.8 was also used as the base classifier for the bagging to compare how the modifications made in the Rotation Forest would affect its performance compared to bagging by using the same base learner. The numbers of base learners for all cases were set to 100 as well as the Rotation Forest.

Table 3. Results achieved by using the Rotation Forest with 100 base classifiers for each individual fold, compared to the AdaBoost.M1, LogitBoost, Bagging and the Random Forest as best-of-the-shelf Meta classifiers based on boosting and bagging approaches. For most of the fold, the Rotation Forest demonstrated better results compared to the other Meta classifiers

Index	Fold	N-test	Rotation Forest	AdaBoost .M1	LogitBoost	Bagging	Random Forest
		α					
1	Globin-like	6	83.3%	83.3%	83.3%	83.3%	83.3%
3	Cytochrome c	9	100.0%	77.8%	55.6%	77.8%	88.9%
4	DNA-Binding 3-Helical	20	85.0%	70.0%	55.0%	55.0%	60.0%
7	4-helical up-and-down bundle	8	25.0%	75.0%	37.5%	62.5%	25.0%
9	4-helical cytokines	9	100.0%	88.9%	77.8%	88.9%	100.0%
11	Alpha; EF-hand	9	33.3%	22.2%	22.2%	11.1%	33.3%
		B					
20	Immunoglobulin-like	44	72.7%	70.5%	77.3%	63.6%	84.1%
23	Cuperdoxins	12	16.7%	41.7%	16.7%	16.7%	25.0%
26	Viral coat and capsid	13	69.2%	76.9%	76.9%	76.9%	76.9%
30	ConA-like lectins/glucanases	6	33.3%	33.3%	33.3%	33.3%	33.3%
31	SH3-like barrel	8	75.0%	75.0%	62.5%	75.0%	62.5%
32	OB-fold	19	26.3%	21.1%	36.8%	21.1%	31.6%
33	Trefoil	4	50.0%	75.0%	50.0%	75.0%	50.0%
35	Trypsin-like serine proteases	4	25.0%	25.0%	50.0%	25.0%	25.0%
39	Lipocalins	7	42.9%	28.6%	57.1%	28.6%	57.1%
		α/β					
46	(TIM)-barrel	48	87.5%	81.3%	81.3%	75.0%	91.7%
47	FAD (also NAD)	12	58.3%	58.3%	41.7%	50.0%	58.3%
48	Flavodoxin-like	13	61.5%	46.2%	46.2%	46.2%	46.2%
51	NAD(P)-binding Rossmann fold	27	40.7%	33.3%	51.9%	33.3%	25.9%
54	P-loop containing nucleotide	12	58.3%	33.3%	33.3%	41.7%	33.3%
57	Thioredoxin-like	8	62.5%	37.5%	50.0%	50.0%	50.0%
59	Ribonuclease H-like motif	12	66.7%	50.0%	58.3%	66.7%	58.3%
62	Hydrolases	7	71.4%	28.6%	57.1%	57.1%	28.6%
69	Periplasmic binding	4	25.0%	0.0%	25.0%	0.0%	25.0%
		α+ β					
72	β-grasp	8	37.5%	25.0%	37.5%	25.0%	25.0%
87	Ferredoxin-like	27	29.6%	48.1%	55.6%	37.0%	37.0%
110	small inhibitors	27	100.0%	100.0%	85.2%	100.0%	96.3%
	TOTAL	383	**62.4%**	58.5%	59.0%	55.4%	59.8%

The overall results were shown in Table.3. Based on the results (Table.3), the Rotation Forest achieved at least, more than 2% higher prediction accuracy compared to other Meta classifiers based on boosting and bagging. It also achieved to the highest prediction accuracy for 16 folds compared to the other employed classifiers, which shows the ability to enhance the prediction accuracy of the Rotation Forest for each fold separately (Table.3). It outperformed Random Forest, the other modification of the bagging classifier by more than 2% of the prediction accuracy, as well as the AdaBoost.M1 classifier by more than 3% of prediction accuracy. Our experimental results showed that the Rotation Forest outperformed the methods which have been used for the protein fold prediction task as well as other Meta classifiers based on boosting and bagging algorithms (AdaBoost.M1, LogitBoost, Bagging, and Random Forest).

5 Conclusion and Future Works

In this paper, an empirical study on the performance and advantages of using the Rotation Forest to solve the protein fold recognition problem were conducted. We also studied the sensitivity of the Rotation Forest to the number of base classifiers by using six different numbers of base classifiers in range between 10 and 200. Finally, employed method compared to the other Meta classifiers based on boosting and bagging approaches which have showed remarkable results on different benchmarks ([1], [2], [3], [6], and [7]).

The ensemble classifier built using the Rotation Forest with 100 base classifiers achieved better results compared to the previous works found in the literature as well as the other best-of-the-shelf Meta classifiers, namely: the AdaBoost.M1, LogitBoost, Bagging and the Random Forest. The proposed method achieved a 62.4% prediction accuracy which is 1.3% higher than the result achieved by Bologna and Appel [14] who used an ensemble of DIMLP, more than 2% better than Nanni [12] who used ensemble of HKNN with FCA and Krishnaraj and Reddy [26] who used the AdaBoost.M1 and the LogitBoost with the same number of base classifiers.

High prediction performance as well as low computational complexity and time consumption of the Rotation Forest shows the potential of this method for further researches. The Rotation Forest is capable to be used by any classifier as a base classifier, being used in hierarchical structure or as part of an ensemble of heterogeneous classifiers to achieve better results for the protein fold prediction problem and other classification tasks.

Acknowledgements

We would like to thank professor Ding and professor Dubchak for letting us use their datasets. We also would like to thank Omid Dehzangi, Kien-Huei Chan, and Keng-Hoong Ng for helpful discussions.

References

1. Rodríguez, J.J., Kuncheva, L.I., Alonso, C.J.: Rotation forest: A new classifier ensemble method. IEEE Transactions on Pattern Analysis and Machine Intelligence 28(10), 1619–1630 (2006)
2. Breiman, L.: Bagging Predictors. Machine Learning 24(2), 123–140 (1996)
3. Freund, Y., Schapier, R.E.: A Short Introduction to Boosting. Journal of Japanese Society for Artificial Intelligence 14(5), 771–780 (1997)

4. Kuncheva, L.I., Rodríguez, J.J.: An Experimental Study on Rotation Forest Ensembles. In: Haindl, M., Kittler, J., Roli, F. (eds.) MCS 2007. LNCS, vol. 4472, pp. 459–468. Springer, Heidelberg (2007)
5. Stiglic, G., Kokol, P.: Effectiveness of Rotation Forest in Meta-learning Based Gene Expression Classification. In: Proceedings of Twentieth IEEE International Symposium on Computer-Based Medical Systems (2007) ISBN: 0-7695-2905-4
6. Friedman, J., Hastie, T., Tibshirani, R.: (Published version) Additive Logistic Regression: a Statistical View of Boosting Annals of Statistics 28(2), 337–407 (2001)
7. Breiman, L.: Random Forest. Machine learning. Kluwer Academic Publishers, Dordrecht (2001) ISSN: 0885-6125
8. Kecman, V., Yang, T.: Protein Fold Recognition with Adaptive Local Hyper plane Algorithm. In: IEEE Symposium Computational Intelligence in Bioinformatics and Computational Biology, CIBCB 2009 (2009); 4925710
9. Stanley, Y., Shi, M.P., Suganthan, N.: Multiclass protein fold recognition using multiobjective evolutionary algorithms. Computational Intelligence in Bioinformatics and Computational Biology (2004); 0-7803-8728-7
10. Okun, O.G.: Protein Fold Recognition with K-local Hyperplane Distance Nearest Neighbor Algorithm. In: Proceedings in the Second European Workshop on Data Mining and Text Mining in Bioinformatics, Pisa, Italy, pp. 51–57 (2004)
11. Chinnasamy, A., Sung, W.K., Mittal, A.: Protein structure and fold prediction using tree-augmented naive Bayesian classifier. Pacific Symposium on Biocomputing. In: Pacific Symposium on Biocomputing, vol. 9, pp. 387–398 (2004)
12. Nanni, L.: Ensemble of classifiers for protein fold recognition. In: New Issues in Neurocomputing: 13th European Symposium on Artificial Neural Networks, vol. 69, pp. 850–853 (2006)
13. Karplus, K.: SAM-T08, HMM-based protein structure prediction. Nucleic Acids Research 37(suppl. 2), W492–W497 (2009)
14. Bologna, G., Appel, R.D.: A comparison study on protein fold recognition. In: Proceedings of the Ninth International Conference on Neural Information Processing, November 2002, vol. 5, pp. 2492–2496 (2002)
15. Huang, C., Lin, C., Pal, N.: Hierarchical learning architecture with automatic feature selection for multi class protein fold classification. IEEE Transactions on Nano Bioscience 2(4), 221–232 (2003)
16. Lin, K.L., Li, C.Y., Huang, C.D., Chang, H.M., Yang, C.Y., Lin, C.T., Tang, C.Y., Hsu, D.F.: Feature Selection and Combination Criteria for Improving Accuracy in Protein Structure Prediction. IEEE Transactions on Nano Bioscience 6(2) (2008)
17. Lin, K.L., Lin, C.Y., Huang, C.D., Chang, H.M., Yang, C.Y., Lin, C.T., Tang, C.Y., Hsu, D.F.: Feature Selection and Combination Criteria for Improving Accuracy in Protein Structure Prediction. IEEE Transactions on Nano Bioscience (2007) ISSN: 1536–1241
18. Chen, C., Zhou, X.B., Tian, Y.X., Zou, X.Y., Cai, P.X.: Predicting protein structural class with pseudo-amino acid composition and support vector machine fusion network. Anal. Biochem. 357, 116–121 (2006)
19. Lewis, D.P., Jebara, T., Noble, W.S.: Support vector machine learning from heterogeneous data: an empirical analysis using protein sequence and structure. Bioinformatics 22, 2753–2760 (2006)
20. Zhang, S.W., Pan, Q., Zhang, H.C., Shao, Z.C., Shi, J.Y.: Prediction protein homo-oligomer types by pseudo amino acid composition: approached with an improved feature extraction and naive Bayes feature fusion. Amino Acids 30, 461–468 (2006)

21. Zhou, X.B., Chen, C., Li, Z.C., Zou, X.Y.: Improved prediction of subcellular location for apoptosis proteins by the dual-layer support vector machine. Amino Acids 35, 383–388 (2008)
22. Ding, C., Dubchak, I.: Multi-class protein fold recognition using support vector machines and neural networks. Bioinformatics 17(4), 349–358 (2001)
23. Dubchak, I., Muchnik, I., Kim, S.K.: Protein folding class predictor for SCOP: approach based on global descriptors. In: 5th International Conference on Intelligent Systems for Molecular Biology, vol. 5, pp. 104–107 (1997)
24. Chung, I.F., Huang, C.D., Shen, Y.H., Lin, C.T.: Recognition of structure classification of protein folding by NN and SVM hierarchical learning architecture. In: Artificial Neural Networks and Neural Information Processing, pp. 1159–1167 (2003)
25. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer, Heidelberg (2006) ISBN-13: 978-0387-31073-2
26. Krishnaraj, Y., Reddy, C.K.: Boosting methods for Protein Fold Recognition: An Empirical Comparison. In: IEEE International Conference on Bioinformatics and Biomedicine (2008) ISBN: 978-0-7695-3452-7
27. Cai, Y.D., Feng, K.Y., Lu, W.C., Chou, K.C.: Using LogitBoost classifier to predict protein structural classes. Journal of Theoretical Biology 238, 172–176 (2006)
28. Zhang, C.X., Zhang, J.S., Wang, J.W.: An empirical study of using Rotation Forest to improve regressors. Applied Mathematics and Computation 195, 618–629 (2007)
29. Scholkopf, B., Smola, A., Muller, K.R.: Nonlinear component analysis as a kernel eigenvalue problem. Neural Computation 10(5), 1299–1319 (1998)
30. Witten, I.H., Frank, E.: Data Mining: Practical machine learning tools and techniques, 2nd edn. Morgan Kaufmann, San Francisco (2005)
31. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, San Francisco (1993)
32. Lo Conte, L., Ailey, B., Hubbard, T.J.P., Braner, S.E., Murzin, A.G., Chothia, C.: SCOP a structural classification of proteins database 28(1), 257–259 (2000)
33. Murzin, A.G., Brenner, S.E., Hubbard, T., Chothia, C.: SCOP: a structural classification of proteins database for the investigation of sequences and structures. Journal of Molecular Biology 247, 536–540 (1995)
34. Hobohm, U., Scharf, M., Schneider, R., Sander, C.: Selection of a representative set of structure from the Brookhaven Protein Bank protein. Science 1, 409–417 (1992)
35. Duwairi, R., Kassawneh, A.: A Framework for Predicting Proteins 3D Structures. In: Computer Systems and Applications, AICCSA 2008 (2008); 978-1-4244-1968
36. Chou, K.C., Zhang, C.T.: Prediction of protein structural classes, Critical Review. Biochem. Mol. Biol. 30(4), 275–349 (1995)
37. Livingston, F.: Implementation of Breiman's Random Forest Machine Learning Algorithm. ECE591Q Machine Learning Journal Paper (2005)
38. Hashemi, H.B., Shakery, A., Naeini, M.P.: Protein Fold Pattern Recognition Using Bayesian nsemble of RBF Neural Networks. In: International Conference of Soft Computing and Pattern Recognition, pp. 436–441 (2009)