JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 26, 1941-1956 (2010)

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

ABDOLLAH DEHZANGI, SOMNUK PHON-AMNUAISUK AND OMID DEHZANGI*
Center of Artificial Intelligence and Intelligent Computing
Faculty of Information Technology
Multimedia University
Cyberjaya, Selangor, 63100 Malaysia
E-mail: {abdollah.dehzangi07; somnuk.amnuaisuk}@mmu.edu.my
**School of Computer Engineering*
Nanyang Technological University
Nanyang Avenue, 639798 Singapore
E-mail: omid0002@ntu.edu.sg

The functioning of a protein in biological reactions crucially depends on its three-dimensional structure. Prediction of the three-dimensional structure of a protein (tertiary structure) from its amino acid sequence (primary structure) is considered as a challenging task for bioinformatics and molecular biology. Recently, due to tremendous advances in the pattern recognition field, there has been a growing interest in applying classification approaches to tackle the protein fold prediction problem. In this paper, Random Forest, as a kind of ensemble method, is employed to address this problem. The Random Forest, is a recently introduced method based on bagging algorithm that trains a group of base classifiers by randomly selecting sets of features and then, combining results obtained from base classifiers by majority voting. To investigate the effectiveness of the number of base learners to the performance of the Random Forest, twelve different numbers of base classifiers (between 30 and 600) are applied for this classifier. To study the performance of the Random Forest and compare its results with previously reported results, the dataset produced by Ding and Dubchak is used. Our experimental results show that the Random Forest enhances the prediction accuracy (using same set of features proposed by Dubchak *et al.*) as well as reduces time consumption of the protein fold prediction task, compared to the previous works found in the literature.

Keywords: protein fold prediction problem, classifier ensemble, random forest, bootstrap sampling, weak learner, feature selection, random sampling, bagging, prediction performance

1. INTRODUCTION

Understanding of how proteins fold in three-dimensional space can reveal significant information of how they function in biological reactions. In these past decades, there has been a growing interest in applying pattern recognition methods to tackle this problem. The protein fold prediction performance using pattern recognition approaches critically depends on two main criteria: features used and employed classifiers. To address the first issue, a variety of features extracted and employed to this task such as: Global physicochemical-based features [1], order-based features extracted based on bi-gram [2, 3], tri-gram [4, 5], and pseudo amino acids composition concepts [6, 7], secondary struc-

Received November 16, 2009; revised February 4, 2010; accepted May 6, 2010.
Communicated by Jorng-Tzong Horng.

ture-based features [4, 8], substitution matrices-based features [9, 10], and hydrophobicity-based feature [4, 11]. However, feature extraction approach is not the solitary effective criteria to enhance the protein fold prediction problem. Employing an appropriate classifier is also has a critical effect to this problem.

Despite all the efforts made at implementing novel classifiers [12-16], enhancement of the protein fold prediction accuracy by using novel classification techniques is still remains limited. Despite the implementation of all the complex classifiers, there is still, a variety of novel and effective classifiers have not been appropriately explored for this task. Hence, in this study, respect to the effectiveness of the features being used for the protein fold prediction problem, we aimed to enhance the protein fold prediction accuracy by addressing the second approach (classifiers being used to solve this problem). Therefore, we employed an ensemble of decision trees called Random Forest to the protein fold prediction problem.

Random Forest, by Breiman [17] is a straightforward modification of bagging algorithm [18], which aims to increase the effectiveness of diversity within the classifier ensemble to enhance its prediction performance. The Random Forest uses bootstrap samples of training dataset to train a group of decision trees as in bagging, except, it randomly selects a subset of features instead of using the whole set of feature vector to train each decision tree. Recently, the Random Forest has been widely applied to different benchmarks and in many cases, outperforms other Meta classifiers such as Adaboost.M1 [19] and Logitboost [20] (which are considered as the best-of-the-shelf Meta Classifiers) or other classifiers such as Support Vector Machine (SVM) which is considered as the state-of-the-art in machine learning [17, 21]. Our experimental results demonstrated that the employed classifier has enhanced the prediction accuracy as well as reduced the time consumption of the classification task (which is studied empirically) as compared to the previous related works found in the literature.

The rest of this paper is organized as follows: section 2 reviews the previous works found in the literature; in section 3, we introduce the Random Forest and the tools were used in our experiments; in section 4, we introduce the datasets and the features were used in this study; in section 5, the performance measurement is introduced and explained; section 6 is concerned with the results and discussion; and eventually, section 7 presents the conclusion and suggested future works.

2. LITERATURE REVIEW

In the past three decades, many efforts have been made to predict how proteins fold in a three-dimensional space. A variety of classification methods have been applied to this task such as Bayesian Classifiers [12, 22], Fuzzy clustering [23, 24], K-nearest neighbor [13, 25], boosting and bagging [26-28], Hidden Markov Model (HMM) [29-32], Artificial Neural Network (ANN) [15, 33-35], and Support Vector Machine [6, 7, 36, 37].

In general, most of the classifiers have been used for the protein fold classification were based on Artificial Neural Network and Support Vector Machine. In 2001, Ding and Dubchak implemented three layer feed forward neural network and three ensemble approaches based on SVM classifier [38] namely: *One-Versus-One* (*OvO*), *Unique One-Versus-One* (*uOvO*) and *All-Versus-All* (*AvA*) used with six feature groups namely: *Com-*

position of amino acids (*C*), *Predicted secondary structure* (*X*), *polarity* (*P*), *polarizability* (*V*), *hydrophobicity* (*H*) *and van der vaals volume* (*V*) to solve this problem [1]. They reported 56% prediction accuracy using AvA SVM for combination of three feature groups (Composition of amino acids, Predicted secondary structure, and hydrophobicity).

Motivated by Ding and Dubchak, Bologna and Appel used an ensemble of four-layer *Discretized Interpretable Multi Layer Perceptron* (DIMLP) [15] trained with the dataset produced by Ding and Dubchak [39]. In addition, they added the length of the amino acid sequence as an effective feature to each feature group. Different from Ding and Dubchak's work, in [15], each classifier learned all the folds simultaneously. It was observed in Ding and Dubchak, and Bologna and Appel's work that the ANN and the SVM performed badly due to the imbalanced proportion of the data that caused a high rate of false positive error.

To solve this problem, Shi *et al*. formulated feature selection problem into Three-objective optimization problem and proposed *Multi-Objective Evolutionary Algorithm* (MOEA) to tackle the protein fold prediction problem [39]. They also employed OvO SVM as their base classifier to get a leverage of the discrimination ability of the SVM classifier. However, the SVM suffered from two main problems; the generalization of this method for multi-class classification task, and the choice of the optimal set of the kernel parameters.

To handle the false problem rate of SVM and ANN for an imbalance proportion of data and to decrease the time consumption of the protein fold classification task, Krishnaraj and Reddy employed boosting-based approaches as a kind of Meta classifiers to deal with this problem [28]. They used AdaBoost.M1 [19] and LogitBoost [20] to handle this task. Boosting-based approaches avoid false positive error by combining many weak learners constructed from different training datasets obtained using sampling with replacement [19]. They reported comparable prediction accuracy in dramatically lower time consumption to the other works found in the literature. However, they could not enhance the protein fold prediction performance better than their previous works (60.3% compared to 61.1% achieved by Bologna and Appel using Dubchak *et al.* feature groups [15]). Despite all the advantages of boosting algorithms, they suffer from over fitting problem while dealing with noisy data with high dimensionality feature vector.

To address boosting based approaches inefficiencies, the Random Forest as a kind of bagging based approach employed by Chen and Kurgan [40], and Jain *et al*. [41] to tackle the protein fold prediction problem. To overcome the over fitting problem of boosting-based approaches while dealing with noisy data, Random Forest provides a proper method to approximate missing data when dealing with noisy data and in the case where large amount of data is missing [17, 21, 42]. In 2007, Chen and Kurgan employed Random Forest using 250 based learners and compare this method to the other state-of-the-art classifiers which showed its promising performance for this task. Recently, Jain *et al.* derived a comparison study to assess the performance of different classifiers (fifteen classifiers from five different categories of pattern recognition algorithms) for the protein fold prediction problem. Their experimental results showed that the Random Forest with using 10 based achieved better results compared to the other classifiers that they employed.

Despite the promising results achieved from the Random Forest, the prediction performance of this classifier has not been studied appropriately for the protein fold predic-

tion problem. In both cases, Random Forest with a fixed and specific number of base classifiers (10 and 250 base classifiers) was used without studying the effectiveness of this parameter to the Random forest's prediction performance. In addition, in none of the cases, their results did not directly compare to the results reported by other works found in literature.

Therefore, Motivated by Krishnaraj and Reddy, to inherit the merits of the Meta classifiers [28], and based on the studies have been conducted by Chen and Kurgan [40], and Jain *et al*. [41] that showed the promising performance of the Random Forest for the Protein fold prediction problem compare to the other Meta classifiers; we derived an empirical study to investigate the performance of the Random Forest based on the number of base learners applied for this classifier, and different combinations of feature groups to assess the sensitivity of the Random Forest to these two different parameters, separately. To compare our results with the previously reported results in the literature, the dataset produced by Ding and Dubchak and the feature groups introduced by Dubchak *et al.* were used (which have been widely used by previous works). Our results showed that the Random Forest has outperformed previous methods developed in the literature as well as reduced the time consumption of the protein fold prediction problem.

3. RANDOM FOREST

Random Forest is a recently proposed classifier by Breiman that generates a classifier ensemble based on feature selection [17]. It is a straightforward modification of bagging method that enforces the diversity between base classifiers in an ensemble method. Bagging takes samples from the training dataset multiple times with replacement for training a group of base classifiers called weak learners [43]. The classifier votes for a combined group of base classifiers (unweighted majority voting). To enhance the prediction accuracy of the classifier, bagging focuses more on the individual accuracy of the base classifiers. For bagging, the only factor which enforces the diversity within a classifier ensemble is the random sampling of the training dataset to train each base classifier. Thus, leaning on just bootstrap sampling to train base classifiers, appears to lead bagging to discourage diversity compared to other ensemble methods which works based on the combination of base classifiers similar to bagging such as the AdaBoost.M1 (Boosting-based approach) [19] and the Rotation Forest (bagging-based approach) [44].

To overcome bagging inefficiencies, two main modifications are considered in the Random Forest classifier. Firstly, the Random Forest uses only decision tree as a base classifier (why it is called Forest). Since decision tree classifiers are usually unstable, therefore they should perform better by bagging base classifiers as base learners [45]. Secondly, each decision tree is trained with the best features among a set of M randomly chosen features among F features contained in features vector (why it is called Random) to encourage diversity between base classifiers. These small changes enforce the diversity of an ensemble of trees without really compromising the prediction accuracy of the decision trees (to encourage diversity within classifier ensemble and individual accuracy of the base learners simultaneously). The Random Forest is considered as an appropriate model to handle large number of input dataset, imbalance dataset, and provide an empirical approach to trace variable interactions [21].

The best features are selected based on the Gini Index between the M selected features. Consider a training set $D = \{(x_i, y_i)\}$ where $i = 1, \ldots, m$ in which each sample $T_i = (x_i, y_i)$ is described by an input attribute vector $x_i = (x_{i,1}, x_{i,2}, x_{i,3}, \ldots, x_{i,F})$ and sample label y_i, while A is a binary feature of F features in the feature vector:

$$Gini_A(D) = \frac{|n_1|}{|n|} Gini(D_1) + \frac{|n_2|}{|n|} Gini(D_2). \quad (1)$$

Where D_1 and D_2 are the binary partitions of D by A which each has n_1 and n_2 records respectively;

$$\text{where } n = n_1 + n_2, D_1 \cup D_2 = D_1 \cap D_2 = \phi \quad (2)$$

and

$$Gini(D) = 1 - \sum_{i=1}^{n} p(y_i = C_i), \text{ where } p(y_i = C_i) = \frac{|C_{i,D}|}{|D|}. \quad (3)$$

Where $C_{i,D}$ is the number correctly classified samples for the class i. In this study, the Random Forest implemented in data mining toolkit WEKA (Waikato Environment for Knowledge Analysis) version 3.6.0 was used for classification task. WEKA is an open source toolkit and it consists of a collection of machine learning algorithms for solving data mining problems [46]. In this experiments, we applied the Random Forest with twelve different number of decision trees to investigate the effectiveness of the number of decision trees on the performance of the Random Forest (in the range of 30 to 600) for the protein fold prediction problem. Our experiments was performed on a PC with a 2.6 GHz Intel CPU and 1 GB RAM, using windows XP.

4. DATASET AND FEATURES

The training and testing datasets studied here were taken from Ding and Dubchak [38]. The original train set was based on the *Protein Data Bank* (*PDB*) [47]. This dataset contained 313 proteins with less than 35% sequential similarity were chosen from 27 most populated folds in PDB. The original test set was based on the *Structural Classification of Proteins* (*SCOP*) databank [48, 49]. This dataset contained 385 proteins that had less than 40% sequence similarity. Recently, two proteins (*i.e.* 2SCMC and 2GPS) in the train set and two proteins (2YHX_1 and 2YHX_2) in the test set were removed from this dataset due to the lack of sequence information. Accordingly, a new version of dataset contains 311 proteins for the train set and 383 proteins for the test set.

The proteins are alphabetic sequences of amino acid of various lengths. Therefore, to apply machine learning algorithms, amino acid sequences are commonly transformed to numerical feature vectors of an equal length [50]. To provide this prerequisite, varieties of features have been extracted from the sequence of amino acids to tackle the protein folding problem [1, 4, 8]. In this paper, the feature groups introduced by Dubchak *et al.* were used [1]. Based on the sequential and physiochemical properties of amino acids, they extracted six feature groups as global descriptors of proteins namely: *amino acid*

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 group of the composition percentage of the 20 amino acid in sequence (as sequential or compositional based feature). The other feature group properties are based on three descriptors: composition: composition percentage 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).

Table 1. The description of six feature groups which used by this task and their dimension size.

Symbol	Feature Group's Name	Dimension
C	Amino Acids Composition	20
S	Predicted Secondary Structure (Normalized frequency of a-Helix Residue)	21
H	Hydrophobicity	21
P	Polarity	21
V	Van der Waals Volume	21
Z	Polarizability	21

Therefore, there are 20 features in the composition feature group and 21 features for other feature groups (as physicochemical based features) [1]. More details of how to extract these features from the proteins can be found in the literature [12, 38, 51]. In this study, the length of the amino acid was also included in each combination of feature groups due to its effectiveness considering its simplicity (a single feature) to the protein folding task [15, 28, 39].

5. ACCURACY MEASUREMENT

In this paper, the Standard *Q* percentage as a popular accuracy measurement was used to make it easier to compare this study to previous works found in the literature [38, 52]. Standard *Q* percentage can be defined as follows; if the number of test proteins belonging to the w_ith fold is n_i, but the tested classifier only recognizes c_i proteins as the w_ith fold, then the accuracy rate of this tested classifier is set as c_i/n_i for the w_ith fold. In addition to the calculation of individual accuracy rate, the total accuracy rate is calculated as follows:

$$N = n_1 + n_2 + n_3 + \ldots + n_i, \text{ for } i = 27 \text{ and } N = 383 \tag{4}$$

$$C = c_1 + c_2 + c_3 + \ldots + c_i \tag{5}$$

$$Q = \frac{C}{N}. \tag{6}$$

Where *N* is the total number of test proteins, *C* is the total number of correctly classified proteins among test set, and *Q is* the classification accuracy.

6. RESULTS AND DISCUSSION

The proposed method was evaluated with eleven different combinations of feature groups instead of six combinations of feature groups used by previous works, to provide more understanding about the effectiveness of each feature group as well as finding better features combinations [28, 38]. As it mentioned earlier, for all the combinations, the length of the amino acid sequence was also added to each combination of the feature groups as an effective feature.

We applied the Random Forest classifier with twelve different numbers of decision trees to investigate the effectiveness of this parameter to the performance of the employed method as well as finding an appropriate number of decision trees (in the range of 30 to 600) for the protein fold prediction problem (30, 50, 70, 80, 100, 150, 200, 250, 300, 400, 500 and 600) (Table 2). As shown in Table 2, we achieved 62.7% prediction accuracy using the Random Forest with 80 decision trees for a combination of five feature groups (C + S + H + P + Z) that has not been used in previous work (Table 3).

The results in Table 3 illustrate that the proposed method outperformed other related methods in the literature in many cases. The proposed method also achieved 58.8% prediction accuracy by using amino acid composition feature group, which was higher than most of the reported results achieved by previous works found in the literature (Table 2).

To investigate the performance of the Random Forest and compare its results to the previous works for each fold individually; the results of using the Random Forest with 80 decision trees for each individual fold compared to the results reported by Ding and Dubchak using AvA SVM [38] (Table 4). As shown in Table 4, the Random Forest outperforms the results achieved by using AvA SVM for different folds containing different numbers of proteins.

Table 2. Results were achieved by using the random forest on independent test dataset with twelve different numbers of decision trees and eleven different combinations of feature groups (%).

Number of decision trees / Features combination	30	50	70	80	100	150	200	250	300	400	500	600
C	57.2	58.5	57.2	56.7	58.2	58.5	58.8	58.8	58.0	58.2	58.0	58.8
C+S	57.4	59.8	61.1	61.1	60.3	60.3	60.8	60.1	59.5	59.5	60.3	60.6
C+S+V	56.4	56.4	57.4	59.3	59.3	60.6	59.8	59.7	59.5	60.6	61.1	60.6
C+S+Z	56.9	58.5	60.6	60.1	60.1	60.3	61.1	60.1	59.8	60.6	60.8	59.8
C+S+P	56.1	58.8	59.0	59.3	59.0	60.3	60.1	61.4	61.1	60.3	59.8	60.1
C+S+H	57.4	59.3	59.3	59.8	59.0	59.8	60.1	61.4	60.6	61.4	61.4	61.6
C+S+H+V	57.4	56.9	59.8	58.2	58.5	59.3	59.5	60.3	60.8	61.1	60.8	60.5
C+S+H+P	58.0	58.8	60.3	60.6	60.6	61.4	59.8	60.6	61.4	**62.1**	62.4	61.4
C+S+H+P+V	55.4	56.7	57.8	58.8	59.3	59.8	60.3	59.8	60.3	60.1	60.8	61.1
C+S+H+P+Z	58.0	60.1	62.1	**62.7**	61.6	**62.1**	61.1	61.1	59.5	60.8	60.6	61.1
C+S+H+P+Z+V	55.1	59.0	60.3	60.8	59.8	59.5	59.1	60.6	60.6	61.6	61.9	61.4

Table 3. Results were achieved by the random forest compared to the results achieved by related works (in percentage %) found in the literature (the most remarkable results achieved by using same set of features proposed by Dubchak *et al.*).

[38]	OvO (SVM)	C+S+H	45.2
[38]	Unique OvO (SVM)	C+S+H	51.1
[38]	OvO(ANN)	C+S+H+P+Z+V	41.8
[38]	AvA(SVM)	C+S+H+P+Z+V	56.4
[33]	MLP-Based HLA	C+S	48.6
This Paper	Random Forest (80 Decision Trees)	C+S	61.1
[33]	RBFN-Based HLA	C+S+H+P+Z+V	56.4
[33]	SVM-Based HLA	C+S+H+P+Z+V	53.2
[18]	AdaBoost.M1	C+S+H	58.2
This Paper	Random Forest (250 Decision Trees)	C+S+H	61.4
[28]	LogitBoost	C+S+H+P+V	60.3
[15]	DIMLP	C+S+H+P+Z+V+ Length of amino acids	**61.1**
[25]	HKNN	C+R+H+P+Z+V	57.4
[14]	BS2_FLC_K125	C+S+H+P+Z+V+ Length of amino acids	59.2
[14]	RS1_HKNN_K125	C+S+H+P+Z+V+ Length of amino acids	60.0
[14]	RS1_KHNN_K25	C+S+H+P+Z+V+ Length of amino acids	60.3
[12]	BAYESPROT	C+S+H+P+Z+V+ Length of amino acids	58.8
[39]	MOFASA	C+S+H+P+Z+V+ Length of amino acids	60.0
[13]	ALH	C+S+H+P+Z+V	60.8
[31]	SHMM	Amino Acids Composition Based Feature	51.6
[35]	MLP Majority Voting Fuse	C+S+H+P+Z+V	40.5
[35]	MLP Bayesian Fuse	C+S+H+P+Z+V	44.5
[35]	RBF Majority Voting Fuse	C+S+H+P+Z+V	49.7
[35]	RBF Bayesian Fuse	C+S+H+P+Z+V	59.0
This Paper	Random Forest (80 Decision Trees)	C+S+H+P+Z	**62.7**

Besides enhancing the prediction accuracy, our experimental results showed that the Random Forest reduced the time consumption of the protein folding considerably. For the Random Forest, each base learner works independently. They build their models on independent bootstrap samples of training data that makes it possible for the Random Forest to build all models for base learners in parallel structures. Then, the results of all base classifiers are combined with the majority voting to build a final ensemble model. Building simple models in parallel structures reduces time consumption of the Random Forest classifier considerably. In addition to that, it allows the Random Forest to increase the number of base classifiers without increasing the time consumption of building its model, significantly.

In our experiments, testing time for the Random Forest took less than five seconds for employed dataset and different combinations of feature groups. It also took less than a minute to train the model for the combinations of all the feature groups with 600 base classifiers and less than eight seconds for 80 base classifiers, which is much faster than ANN or SVM (which took more than ten minutes) as well as AdaBoost.M1 and LogitBoost (which took more than 5 minutes with using 100 base classifiers) [28, 38].

However, parallel structure and independence modeling of the base classifiers cause the Random Forest to occupy bigger memory than boosting-based methods which build their models by sequentially applying their base classifiers. Considering the prediction accuracy of the Random Forest as well as its time consumption; makes the Random Forest classifier as a suitable classifier to be used for the relatively medium size datasets.

Table 4. Results were achieved by using the random forest with 80 decision trees for each individual fold compared to the All-versus-All method with the SVM classifier, used by Ding and Dubchak (in percentage %) [39]. For most of the folds, the random forest demonstrates better than (for fifteen cases) or equals to (for eight cases) the results of the SVM (AVA).

Index	Fold		N-test	Random Forest (80 Decision Trees)	SVM (AvA)
		α			
1	Globin-like		6	83.3%	83.3%
3	Cytochrome c		9	88.9%	77.8%
4	DNA-Binding 3-Helical bundle		20	60.0%	35.0%
7	4-helical up-and-down bundle		8	37.5%	50.0%
9	4-helical cytokines		9	100.0%	100.0%
11	Alpha; EF-hand		9	22.2%	66.7%
		β			
20	Immunoglobulin-like β-sandwich		44	81.8%	71.6%
23	Cuperdoxins		12	16.7%	16.7%
26	Viral coat and capsid proteins		13	69.2%	50.0%
30	ConA-like lectins/glucanases		6	33.3%	33.3%
31	SH3-like barrel		8	75.0%	50.0%
32	OB-fold		19	36.8%	26.3%
33	Trefoil		4	50.0%	50.0%
35	Trypsin-like serine proteases		4	25.0%	25.0%
39	Lipocalins		7	71.4%	57.1%
		α/β			
46	(TIM)-barrel		48	89.6%	77.1%
47	FAD (also NAD)-binding motif		12	58.3%	58.3%
48	Flavodoxin-like		13	69.2%	48.7%
51	NAD(P)-binding Rossmann fold		27	40.7%	61.1%
54	P-loop containing nucleotide		12	41.7%	36.1%
57	Thioredoxin-like		8	37.5%	50.0%
59	Ribonuclease H-like motif		12	58.3%	35.7%
62	Hydrolases		7	57.1%	71.4%
69	Periplasmic binding protein-like		4	25.0%	25.0%
		$\alpha+\beta$			
72	β-grasp		8	25.0%	12.5%
87	Ferredoxin-like		27	48.1%	37.0%
110	small inhibitors, toxins, lectins		27	96.3%	83.3%
	AVERAGE		383	62.7%	56.0%

Despite achieving high prediction accuracy by applying the Random Forest on the independent test set, experiments on independent test set do not provide sufficient information about the effectiveness of the number of decision trees on the prediction accuracy. Thus, to reveal this relation, we evaluated the Random Forest classifier with 10-fold cross validation on train set. 10-fold cross validation separated the train dataset equally to 10 parts. It trains an employed method using nine parts of 10 parts and evaluates it with the remaining part. It repeats this process for all 10 parts and then, combines all the results with together as the output [53]. Repeating the experiments for 10 times, make this evaluation method more capable to provide more robust and general results compared to use of the test dataset to evaluate an employed method [54].

Table 5. Results were achieved by using random forest using 10-fold cross validation evaluation criteria for twelve different numbers of decision trees and eleven different combinations of feature groups.

Number of decision trees / Features combination	30	50	70	80	100	150	200	250	300	400	500	600
C	40.8	43.1	43.1	43.1	43.4	41.8	43.7	43.7	43.7	46.0	44.7	45.3
C+S	51.5	53.4	55.0	54.3	55.3	55.3	54.0	55.0	55.0	55.6	56.0	55.3
C+S+V	49.5	50.8	53.4	53.7	54.3	54.0	55.0	55.6	55.3	56.0	55.0	54.7
C+S+Z	49.2	49.5	53.4	54.3	53.7	54.3	53.4	55.0	56.3	**56.3**	55.6	55.0
C+S+P	48.6	47.9	50.5	53.4	53.4	53.4	52.4	55.3	54.7	56.0	**56.6**	55.0
C+S+H	50.2	49.8	52.7	52.7	53.7	53.7	55.3	54.7	53.7	53.4	56.0	**56.9**
C+S+H+V	46.0	49.5	51.5	52.0	54.0	55.0	54.0	53.7	53.7	54.7	55.6	54.3
C+S+H+P	45.7	50.5	53.1	53.4	53.4	53.1	54.0	54.0	53.1	52.4	53.1	51.8
C+S+H+P+V	41.5	45.7	47.0	47.3	48.2	48.2	49.8	51.5	50.5	50.8	50.5	50.5
C+S+H+P+Z	46.3	46.6	46.6	49.2	50.8	50.5	52.1	51.8	50.8	51.8	52.8	51.8
C+S+H+P+Z+V	43.4	46.0	46.0	45.7	47.3	47.9	48.9	50.8	51.1	51.5	50.5	49.2

As it is shown in Table 5, the results were achieved by evaluating Random Forest with using 10-fold cross validation as its evaluation criterion, revealed general information of the relation between the number of decision trees and the prediction accuracy for the protein fold prediction task (Table 5). They also show that the Random Forest outperformed previously reported results by using the 10-fold cross validation evaluation method, as well as the enhancement achieved by applying the Random Forest for independent test set. We achieved 56.9% using combination of three feature groups (amino acid composition, predicted secondary structure and hydrophobicity) which is more than 3% better than 53.8% prediction accuracy reported by Krishnaraj and Reddy using combination of all six feature groups [28].

As it was mentioned earlier, the sequential similarity among the proteins contained in the train dataset is less than 35% which is 5% lower than the sequential similarity among the proteins in the test dataset (which is less than 40%) which reduce the prediction performance using 10-fold cross validation compare to use of test dataset to evaluate employed methods (sequential similarity is considered as critical criteria on the prediction performance of the protein fold prediction problem, in the way that decreasing the sequential similarity, dramatically reduce the protein fold prediction performance as well).

According to Brieman's research, the number of appropriate decision trees is dramatically dependent on the features of the problem and can be calculated experimentally [17]. To find the appropriate number of decision trees, twelve decision trees in the range of 30 to 600 were tested and the average of probabilities for eleven combinations of feature groups were calculated and compared separately for independent test sets and 10-fold cross validation (Fig. 1).

In Fig. 1 two plots according to the average of prediction accuracy of the Random Forest for each number of decision trees, for the independent dataset and 10-fold cross validation evaluation criteria compared and analyzed. As illustrated in Fig. 1 the plot of the average of probabilities for each number of decision tress achieved by 10-fold cross validation demonstrates two local maximums for 100 and 250 decision trees and a global maximum for 500 decision trees in the specified range of the number of decision trees

were studied in this paper. In general, the results were achieved by using 10-fold cross validation evaluation criteria showed higher rate of the prediction accuracy increment by increasing the number of base learners (Fig. 1) which emphasis on the effectiveness of the number of base learners on the prediction performance of the Random Forest.

Fig. 1 shows a comparison of the average of the prediction accuracy of using the Random Forest for each number of decision trees, using independent test dataset and 10-fold cross validation evaluation methods (average of each column in Tables 1 and 2). As we can see, both of the plots almost have a same shapes (increasing the number of base classifiers would increase the averaged of prediction accuracy). However, the plot according to the average of prediction accuracy which was evaluated by using 10-fold cross validation, shown higher sensitivity to the number of base classifiers compared to the plot which is driven based on the average of prediction accuracy which was evaluated by using independence test set for the Random Forest classifier.

To provide more information of the performance of the Random Forest based on the number of base classifiers, the intensity plot of the using test set and 10-fold cross validation criteria are shown in Figs. 2 and 3. Comparison of the intensity plots (Figs. 2 and 3) demonstrate that two plots tend to make similar shapes by increasing the number of

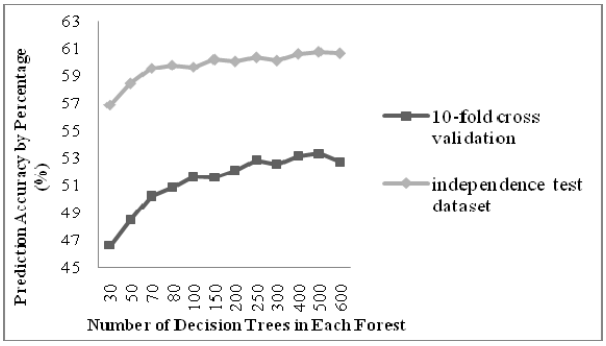

Fig. 1. A comparison of the average of the prediction accuracy.

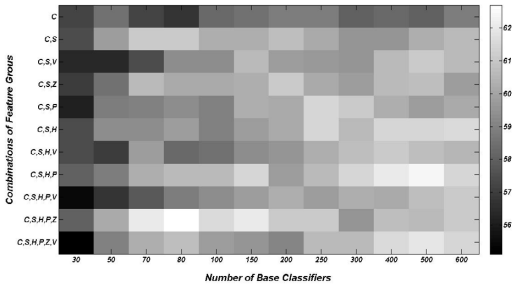

Fig. 2. A intensity plot of the results were achieved by evaluating the random forest, using test dataset. As it shows, increasing the number of classifiers and number of features almost monotonically increase the prediction performance.

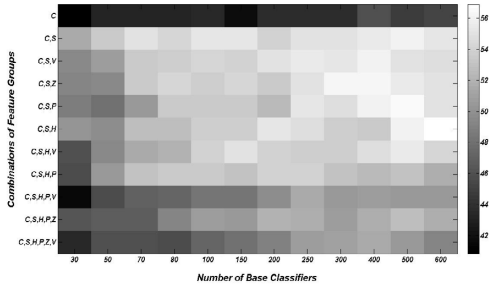

Fig. 3. A intensity plot of the results were achieved by evaluating the random forest, using 10-fold CV. As it shows, increasing the number of classifiers monotonically enhance the prediction performance until using 500 classifiers.

weak learners. Increasing number of base learners enhanced the prediction accuracy of Random Forest using both evaluation criteria (independent test set and 10-fold cross validation) [55].

7. CONCLUSSION AND FUTURE WORKS

In this paper, an empirical study on the performance and advantages of using the Random Forest to solve the protein fold recognition problem has been conducted. To study the sensitivity of the Random Forest to the number of base classifiers, twelve different numbers of decision trees in the range of 30 to 600 were employed and applied in this paper. Our results showed, the prediction accuracy of the Random Forest is almost monotonically increased by increasing the number of base classifiers (local maximum in 500 between 30 and 600 number of base classifiers). It also provided an approximation of the appropriate number of base classifiers for the Random Forest for the employed benchmark. Experimental results also, demonstrate that the employed method enhanced the prediction accuracy with lower time consumption compared to the previous methods proposed in the literature [15, 28]. We achieved 62.7% prediction accuracy for independent test sets better than 61.1% prediction accuracy reported by Bologna and Appel using ensemble of DIMLP [15]. We also achieved 56.9% prediction accuracy for 10-fold cross validation better than 53.8% prediction accuracy reported by Krishnaraj and Reddy, using AdaBoost.M1 [28].

Random Forest is flexible as it can be implemented in a hierarchical structure, or as components in ensemble classifiers which should enhance the accuracy of the protein fold prediction. The results achieved by Random Forest as well as the reduction in time consumption and the simplicity of this method show the potential of this method which can be considered in future research.

ACKNOWLEDGEMENT

We would like to thank professor Ding and professor Dubchak for letting us use their datasets. We also would like to thank Arash Dehzangi, Kein Huei Chan, and Abdelrahman Osman Elfaki for helpful discussions.

REFERENCES

1. I. Dubchak, I. Muchnik, and S. K. Kim, “Protein folding class predictor for SCOP: approach based on global descriptors,” in *Proceedings of the 5th International Conference on Intelligent Systems for Molecular Biology*, Vol. 5, 1997, pp. 104-107.
2. C. D. Huang, C. T. Lin, and N. R. Pal, “ Hierarchical learning architecture with automatic feature selection for multiclass protein fold classification,” *IEEE Transactions on Nanobioscience*, Vol. 2, 2003, pp. 221-232.
3. K. L. Lin, C. Y. Li, C. D. Huang, H. M. Chang, C. Y. Yang, C. T. Lin, C. Y. Tang, and D. F. Hsu, “Feature selection and combination criteria for improving accuracy in protein structure prediction,” *IEEE Transactions on Nanobioscience*, Vol. 6, 2008,

pp. 186-196.
4. N. R. Pal and D. Chakraborty, "Some new features for protein fold prediction," in *Proceedings of International Conference on Artificial Neural Networks and Neural Information Processing*, 2003, pp. 1176-1183.
5. P. Ghanty and N. R. Pal, "Prediction of protein folds: Extraction of new features, dimensionality reduction, and fusion of heterogeneous classifiers," *IEEE Transactions on Nanobioscience*, Vol. 8, 2009, pp. 100-110.
6. K. C. Chen, Y. X. Tian, X. Y. Zou, P. X. Cai, and J. Y. Mo, "Using pseudo amino acid composition and support vector machine to predict protein structural class," *Journal of Theoretical Biology*, Vol. 243, 2006, pp. 444-448.
7. K. C. Chen, X. B. Zhou, Y. X. Tian, X. Y. Zou, and P. X. Cai, "Predicting protein structural class with pseudo-amino acid composition and support vector machine fusion network," *Analytical Biochemistry*, Vol. 357, 2006, pp. 116-121
8. M. T. A. Shamim, M. Anwaruddin, and H. A. Nagarajaram, "Support vector machine-based classification of protein folds using the structural properties of amino acid residues and amino acid residue pairs," *Bioinformatics*, Vol. 23, 2007, pp. 3320-3327.
9. T. Damoulas and M. A. Girolami, "Probabilistic multi-class multi-kernel learning: on protein fold recognition and remote homology detection," *Bioinformatics*, Vol. 24, 2008, pp. 1264-1270.
10. Y. Ying, K. Huang, and C. Campbell, "Enhance protein fold recognition through a novel data integration approach," *BMC Bioinformatics*, Vol. 10, 2009, pp. 267-287.
11. H. B. Shen and K. C. Chou, "Ensemble classifier for protein fold pattern recognition," *Bioinformatics*, Vol. 15, 2006, pp. 1717-1722.
12. A. Chinnasamy, W. K. Sung, and A. Mittal, "Protein structure and fold prediction using tree-augmented naive Bayesian classifier," in *Proceedings of Pacific Symposium on Biocomputing*, Vol. 9, 2004, pp. 387-398.
13. V. Kecman and T. Yang, "Protein fold recognition with adaptive local hyper plane algorithm," in *Proceedings of IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology*, 2009, pp. 75-78.
14. L. Nanni, "Ensemble of classifiers for protein fold recognition," *New Issues in Neurocomputing: 13th European Symposium on Artificial Neural Networks*, Vol. 69, 2006, pp. 850-853.
15. G. Bologna and R. D. Appel, "A comparison study on protein fold recognition," in *Proceedings of the 9th International Conference on Neural Information Processing*, Vol. 5, 2002, pp. 2492-2496.
16. T. L. Zhang, Y. S. Ding, and K. C. Chou, "Prediction protein structural classes with pseudo-amino acid composition: approximate entropy and hydrophobicity pattern," *Journal Theoretical Biology*, Vol. 250, 2008, pp. 186-193.
17. L. Breiman, "Random forest," *Machine Learning*, Vol. 45, 2001, pp. 5-32.
18. L. Breiman, "Bagging predictors," *Machine Learning*, Vol. 24, 1996, pp. 123-140.
19. Y. Freund and R. E. Schapier, "A short introduction to boosting," *Journal of Japanese Society for Artificial Intelligence*, Vol. 14, 1999, pp. 771-780.
20. J. Friedman, T. Hastie, and R. Tibshirani, "Additive logistic regression: a statistical view of boosting," *Annals of Statistics*, Vol. 28, 2000, pp. 337-407.
21. F. Livingston, "Implementation of Breiman's random forest machine learning algo-

rithm," *Machine Learning Journal Paper*, 2005, ECE591Q.
22. A. Raval, Z. Ghramani, and D. L. Wild, "A bayesian network model for protein fold and remotehomologue recognition," *Bioinformatics*, Vol. 18, 2002, pp. 788-801.
23. Z. C. Li, X. B. Zhou, Y. R. Lin, and X. Y. Zou, "Prediction of protein structure class by coupling improved genetic algorithm and support vector machine," *Amino Acid*, Vol. 35, 2008, pp. 581-590.
24. H. B. Shen, J. Yang, X. J. Liu, and K. C. Chou, "Using supervised fuzzy clustering to predict protein structural classes," *Biochemical and Biophysical Research Communications*, Vol. 334, 2005, pp. 577-581.
25. O. G. Okun, "Protein fold recognition with K-local hyperplane distance nearest neighbor algorithm," in *Proceedings of the 2nd European Workshop on Data Mining and Text Mining in Bioinformatics*, 2004, pp. 51-57.
26. L. Dong, Y. Yuan, and T. Cai, "Using bagging classifier to predict protein domain structural class," *Journal of Biomolecular Structure and Dynamics*, Vol. 3, 2006, pp. 239-242.
27. K. Y. Feng, Y. D. Cai, and K. C. Chou, "Boosting classifier for predicting protein domain structural class," *Biochemical and Biophysical Research Communications*, Vol. 334, 2005, pp. 213-217.
28. Y. Krishnaraj and C. K. Reddy, "Boosting methods for protein fold recognition: An empirical comparison," in *Proceedings of IEEE International Conference on Bioinformatics and Biomedicine*, 2008, pp. 393-396.
29. C. Lampros, C. Papaloukasac, T. P. Exarchosab, D. I. Fotiadisad, and D. Tsalikakisae, "Improving the protein fold recognition accuracy of a reduced state-space hidden Markov model," *Computers in Biology and Medicine*, Vol. 39, 2009, pp. 907-914.
30. C. Lampros, C. Papaloukasac, T. P. Exarchos, Y. Goletsis, and D. I. Fotiadis, "Sequence-based protein structure prediction using a reduced state-space hidden Markov model," *Computers in Biology and Medicine*, Vol. 37, 2007, pp. 1211-1224.
31. D. Bouchaffra and J. Tan, "Protein fold recognition using a structural hidden Markov model," in *Proceedings of the 18th International Conference on Pattern Recognition*, 2006, pp. 186-189.
32. K. Karplus, "SAM-T08, HMM-based protein structure prediction," *Nucleic Acids Research*, Vol. 37, No. suppl_2, 2009, W492-W497.
33. I. F. Chung, C. D. Huang, Y. H. Shen, and C. T. Lin, "Recognition of structure classification of protein folding by NN and SVM hierarchical learning architecture," *Artificial Neural Networks and Neural Information Processing*, 2003, pp. 1159-1167.
34. A. Dehzangi, S. P. Amnuaisuk, and K. H. Ng, "Investigating the influence of combined features to classifiers' performance: A comparison study on a protein fold prediction problem," in *Proceedings of the 6th International Conference on Information Technology in Asia*, 2009, pp. 213-217.
35. H. B. Hashemi, A. Shakery, and M. P. Naeini, "Protein fold pattern recognition using bayesian ensemble of RBF neural networks," in *Proceedings of International Conference of Soft Computing and Pattern Recognition*, 2009, pp. 436-441.
36. M. Takata and Y. Matsuyama, "Protein folding classification by committee SVM array," *Advances in Neuro-Information Processing: 15th International Conference*, Part II, 2008, LNCS 5507, pp. 369-377.
37. X. D. Sun and R. B. Huang, "Prediction of protein structural classes using support

vector machines," *Amino Acids*, Vol. 30, 2006, pp. 469-475.
38. C. Ding and I. Dubchak, "Multi-class protein fold recognition using support vector machines and neural networks," *Bioinformatics*, Vol. 17, 2001, pp. 349-358.
39. S. Y. M. Shi and N. Suganthan, "Multiclass protein fold recognition using multiobjective evolutionary algorithms," in *Proceedings of IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology*, 2004, pp. 61-66.
40. K. Chen and L. Kurgan, "PFRES: Protein fold classification by using evolutionary information and predicted secondary structure," *Bioinformatics*, Vol. 23, 2007, pp. 2843-2850.
41. P. Jain, J. M. Garibaldi, and J. D. Hirst, "Supervised machine learning algorithms for protein structure classification," *Computational Biology and Chemistry*, Vol. 33, 2009, pp. 216-223.
42. M. Gashler, C. G. Carrier, and T. Martinez, "Decision tree ensemble: Small heterogeneous is better than large homogeneous," in *Proceedings of the 7th International Conference on Machine Learning and Applications*, 2008, pp. 900-905.
43. E. R. Schapire, "The strength of weak learn ability," *Journal of Machine Learning*, Vol. 5, 1990, pp. 197-227.
44. J. J. Rodríguez, L. I. Kuncheva, and C. J. Alonso, "Rotation forest: A new classifier ensemble method," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 28, 2006, pp. 1619-1630.
45. E. Bauer and R. Kohavi, "An empirical comparison of voting classification algorithms: Bagging, boosting, and variants," *Machine Learning*, Vol. 36, 1999, pp. 105-139.
46. I. H. Witten and F. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*, 2nd ed., Morgan Kaufmann, San Francisco, CA, 2005.
47. U. Hobohm, M. Scharf, R. Schneider, and C. Sander, "Selection of a representative set of structure from the brookhaven protein bank protein," *Science*, Vol. 1, 1992, pp. 409-417.
48. A. G. Murzin, S. E. Brenner, T. Hubbard, and C. Chothia, "SCOP: A structural classification of proteins database for the investigation of sequences and structures," *Journal of Molecular Biology*, Vol. 247, 1995, pp. 536-540.
49. L. L. Conte, S. E. Brenner, T. J. P. Hubbard, C. Chothia, and A. G. Murzin, "SCOP database in 2002: refinements accommodate structural genomics," *Nucleic Acid Research*, Vol. 30, 2002, pp. 264-267.
50. Z. P. Liu, L. Y. Wu, Y. Wang, X. S. Zhang, and L. Chen, "Bridging protein local structures and protein functions," *Amino Acids*, Vol. 35, 2008, pp. 627-650.
51. R. Duwairi and A. Kassawneh, "A framework for predicting proteins 3D structures," in *Proceedings of IEEE International Conference on Computer Systems and Applications*, 2008, pp. 37-44.
52. B. Rost and C. Sander, "Prediction of protein secondary structure at better than 70% accuracy," *Journal of Molecular Biology*, Vol. 232, 1993, pp. 584-599.
53. R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed., Wiley, New York, 2001.
54. C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer Science, Berlin, 2006.
55. N. Pater, "Enhancing random forest implementation in Weka," in *Proceedings of Conference on Machine Learning*, 2005, ECE591Q.

Abdollah Dehzangi received the B.Sc. degree in Computer Engineering-Hardware from Shiraz University, Iran in 2007. Since 2008, he is pursuing the Master degree by research in Bioinformatics at Multi Media University (MMU), Malaysia. His research interests include bioinformatics, protein fold prediction problem, data mining, and machine learning in pattern recognition.

Somnuk Phon-Amnuaisuk received his B.Eng. from King Mongkut Institute of Technology (Thailand) and Ph.D. in Artificial Intelligence from the University of Edinburgh (Scotland). He is currently a Deputy Dean for the faculty of Information Technology, Multimedia University, Malaysia, where he also leads the Music Informatics Research group. His current research works span over multimedia information retrieval, Bayesian networks, data mining and machine learning. He has also served as a committee member in many editorial boards and research grant screening committees.

Omid Dehzangi received the B.Sc. and M.Sc. degrees in Computer Engineering from Shiraz University, Iran in 2002 and 2007, respectively. Since 2007, he is pursuing the Ph.D. degree in Computer Engineering at Nanyang Technological University (NTU), Singapore. His research interests include machine learning in pattern recognition particularly speech recognition, speaker recognition and spoken language recognition.