Available online at www.dergipark.gov.tr

INTERNATIONAL ADVANCED RESEARCHES and ENGINEERING JOURNAL

Journal homepage: www.dergipark.gov.tr/iarej

e-ISSN: 2618-575X

International Open Access

Volume 03
Issue 01

April, 2019

Research Article

Determination of highly effective attributes in fold level classification of proteins

Özlem Polat *

Department of Mechatronic Engineering, Cumhuriyet University, Sivas 58140, Turkey

ARTICLE INFO

Article history:
Received 28 February 2018
Received 31 December 2018
Accepted 13 January 2019

Keywords:
Divergence analysis
Fold recognition attributes
Neural networks
Protein fold classification

ABSTRACT

In this paper it is aimed to determine which of the protein features or attributes is the most significant for classification of proteins according to their folds. Proteins in the database used in this study are represented by six feature groups called attributes and by a 125-dimensional feature vector. The representation of proteins with very high dimensional vectors such as 125 causes increasing computational load of the classification process and extending the process time. In this study "dimension reduction" solution is offered for this negative situation. Hence, with two different approaches, the features and attributes having high classification performance are determined. In the first approach, which attribute gives higher performance is determined by testing separately each of the six attributes. In the second approach, the most significant of the 125 features are determined using Divergence Analysis method. In this study, a classic classifier KNN (K-nearest neighbor) and artificial neural network models GAL (Grow and Learn) and SOM (Self-Organizing Map) networks are used as classifier and classification performance is analyzed for reduced dimension datasets.

1. Introduction

Proteins are essential and large biological macromolecules that regulate necessary parts of living organisms to control all their living functionalities [1]. Proteins having same or similar shape in a given locus perform the same or similar functions. Structural comparison and classification of proteins is therefore have a great importance in terms of computational biology [2]. Information on all known proteins is stored in the Protein Data Bank [3]. Currently (December 31, 2018) has 147,610 protein structures experimentally identified in this database, and this number is increasing every month by adding an average of 800 new molecules. Thus, many similar structures are formed in this database. The comparison and classification of protein structures is also important in this respect. SCOP (Structural Classifications of Proteins) [4] provides comprehensive evolutionary and structural relationships among all known proteins [5]. Proteins are divided into four main structural classes according to the components of the secondary structure; all-alpha, all-beta, alpha/beta and alpha+beta; and according to SCOP, these four main classes are divided into folds, folds are divided into superfamilies, and superfamilies are divided into families. Folds represent the 3D shape of proteins and because of that the protein structure defines the protein function. Therefore, classifying proteins according to their folds is an important issue for structural biology.

There are many studies in the literature about proteins. Over the past 30 years, a wide variety of research has been conducted on the classification of protein folds. In these studies, classifiers such as artificial neural networks [5-12], Bayesian classifiers [13], k-nearest neighbors [14-17], support vector machines [18-20] were used as well as ensemble classifiers [1,21-30] using more than one classifier were used. In all these studies, different methods were used to classify the proteins at the fold level, among which only [8,26,30] tested separately the feature groups used in the classification of proteins.

One of the earliest studies on classifying protein at the fold level was performed by Reczko and Bohr [6]. Reczko

* Corresponding author. Tel: *+90 346 219 10 10*
E-mail address: ozlem.polat@cumhuriyet.edu.tr
ORCID: *0000-0002-9395-4465*
Note: Part of this study was presented at International Advanced Researches & Engineering Congress, Osmaniye /Turkey

and Bohr used a special feed-forward artificial neural network model called Cascade-Correlators. In 1999 Dubchak et al. developed a new computational method based artificial neural network to assign the protein sequence to a fold class in SCOP [5]. In 2001 Edler et al. [7] conducted a study showing the role and consequence of statistical methods in predicting protein fold classes. In the same year, Ding and Dubchak [8] applied support vector machines and artificial neural networks as the primary classifier to recognize protein folds. In these studies, Ding and Dubchak used a dataset, which they had generated in previous studies [5], containing 27 protein folds. This dataset contains a total of 694 proteins, 311 in the training set and 383 in the test set. Bologna and Appel used an ensemble of four-layer Discretized Interpretable Multi-Layer Perceptron [9] and used the dataset formed by Ding and Dubchak [8]. In 2003, Huang et al. [31] proposed a hierarchical learning architecture that separates proteins into four structural classes; as a second step, they tried to solve 27-class protein fold classification problem using MLP, GRNN, RBFN and SVM classifiers. In 2004, Okun [14] used a modified nearest neighbor algorithm called the K-Local Hyperplane Distance Nearest Neighbor. In 2005, Chinnasamy et al. [13] presented a system called TAN (Tree-Augmented Network) and based Bayesian Network for classification problem. In the same year, Huang et al. [11] used a hierarchical learning architecture based on artificial neural networks, in which the attributes were selected during learning to classify proteins at the fold level. In 2006 Nanni [22,23] used two different ensemble classifiers. Shen and Chou [15] developed a hybrid classifier called OET-KNN (K-Nearest Neighbor Optimized Evidence Theoretic) and they tested it on the 27-class dataset used in [8]. In 2007, Chen and Kurgan [24] proposed PFRES method for classifying protein folds with an automatized way. Shamim et al. [20] used a SVM-based classifier. Motivated by Shen and Chou, Guo and Gao [25] proposed a new hierarchical composite classifier called GAOEC (Genetic Ensemble Classifier Optimized Algorithm). A different study on classification of proteins at the fold level was also performed by Krishnaraj and Reddy [32]. Krishnaraj and Reddy used the AdaBoost and LogitBoost methods, which are two different variations of Boost algorithms. Later on, Damoulas and Girolami [30] used the variational Bayesian approach and the Kernel combination methodology for classification. In 2009 Shen and Chou [33] used a hybrid classifier; Hashemi et al. [1] used MLP and RBF networks; Chen et al. [26] used a new approach based on genetic algorithm and Jazebi et al. [12] used a fusion method to classify the proteins at the fold level. In 2010 Dehzangi et al. applied Random Forest [34] and Rotation Forest [35] algorithms; Wang and Gao [36] used a two-layer classifier in which OET-KNN is used in the first layer and SVM is used in the second layer. Motivated by [32,34,35] Dehzangi and Karamizadeh used a heterogeneous classifier including LogitBoost, Random Forest and Rotation Forest algorithms [37] for protein fold classification. In 2012 Suvarnavani et al. [38] applied boosting algorithm called SMOTE and used Triangle Sub division Method (TSM) to extract the feature set; then used a decision tree classifier and obtained 78.25% classification accuracy. In 2013 Bae et al. [39] tried to solve the classification problem by using multi-class linear decomposition analysis method. One of the recent studies on this area has been studied by Lin et al. [28]. They applied K-means clustering algorithm in their study. To the best of our knowledge the most recent study in the literature has been made by Aram et al. [29] in 2015. Aram et al. used a two-layer classification framework (TLCF) and a mixture of MLP, RBFN and Rotation Forest algorithms for classification of protein folds.

The studies mentioned above and available in the literature tried to solve protein fold classification problem. Only three of these studies [8,26,30] have evaluated the feature groups separately to classify the proteins according to their folds.

In this study, three different classifiers, KNN, SOM and GAL, are used to classify proteins at the fold level. In addition, in this study, the feature groups (attributes) are evaluated individually in order to determine the most significant attributes for protein fold classification. Also the Divergence Analysis method is used to array the features according to their effectiveness and determine the most significant features for protein fold classification. Thus, in the literature, the feature size in the dataset which is frequently used in the classification of proteins is reduced to a value lower than 125 and the computation load is reduced, therefore the computation time is reduced.

In Section 2 Materials and methods are introduced. In Section 3 experiments and performances are presented and conclusions are brought in Section 4.

2. Materials and Methods

2.1 Dataset and Features

The dataset used in this paper was taken from [8] and still available at http://ranger.uta.edu/~chqding/bioinfo.html. The original training and test sets have 311 and 383 proteins, respectively. This dataset consists of 27 folds. These folds are shown in Table 1.

To cope with the fold classification problem, Ding and Dubchak formed the following six attributes from protein sequences; amino acid composition, predicted secondary structure, hydrophobicity, normalized van der Waals volume, polarity and polarizability [8]. Of the above six attributes, only amino acid composition has 20 components, each

component states the occurrence frequency of one of the 20 native amino acids in a given protein. The other five attributes have 21 components.

The occurrence frequencies of the 20 native amino acids in a particular protein form the components of the composition vector of that protein. The 20 amino acids are denoted with letters in alphabetic order each one is represented as AA_i .

Table 1. The 27 protein folds, structural classes and the number of proteins contained in training and test sets [8]

Fold Number	Fold name	Structural Class	Train set	Test set
1	Globin-like	all-α	13	6
2	Cytochrome c		7	9
3	DNA-binding 3-helical bundle		12	20
4	4-helical up-and-down bundle		7	8
5	4-helical cytokines		9	9
6	Alpha; EF-hand		6	9
7	Immunoglobulin-like β-sandwich	all-β	30	44
8	Cupredoxins		9	12
9	Viral coat and capsid proteins		16	13
10	ConA-like lectins/glucanases		7	6
11	SH3-like barrel		8	8
12	OB-fold		13	19
13	Trefoil		8	4
14	Trypsin-like serine proteases		9	4
15	Lipocalins		9	7
16	TIM-barrel	α/β	29	48
17	FAD (also NAD)-binding motif		11	12
18	Flavodoxin-like		11	13
19	NAD(P)-binding Rossmann-fold		13	27
20	P-loop containing nucleotide		10	12
21	Thioredoxin-like		9	8
22	Ribonuclease H-like motif		10	12
23	Hydrolases		11	7
24	Periplasmic binding protein-like		11	4
25	B-grasp	α+β	7	8
26	Ferredoxin-like		13	27
27	Small inhibitors, toxins, lectins		13	27

The number of occurrences of AA_i in the given sequence is shown as n_i (i=1,2,…,20). Then, the components of the composition vector are introduced as:

$$\frac{n_1}{L}, \frac{n_2}{L}, \dots \frac{n_{19}}{L}, \frac{n_{20}}{L} \quad (1)$$

where, L denotes the length of the sequence [24].

Three identifiers are calculated for each of the five attributes; composition (C), transition (T) and distribution (D). Composition introduces the histogram related to the three groups in a protein. Transition shows the percent frequencies related to the change between the groups. Distribution indicates the distribution of the attributes in the sequence. For each of these five attributes, totally 3(C)+3(T)+5×3(D)=21 scalar components are formed. As a result, the dimension of feature vector is 125 [5].

2.2 K-Nearest Neighbor Algorithm

K nearest neighbor algorithm is a supervised learning method usually used to classify any data and its implementation is simple. In this method, the distances between the samples in the training set and the samples in the test set are calculated one by one. After the distances have been calculated, the closest K neighbors to the one whose class is to be specified is determined. The class of the sample being tested is determined by the majority method [40]. If the number of samples related to the class is in the majority in K neighbors, the class of tested protein is determined as the class of the majority. Different distance metrics can be used in the KNN method and the user usually defines the value of K from small positive integers. In this study, Euclidean metric was used as distance metric. For the K constant, several different values have been tested in order to evaluate the classification performance, for example 1, 3, 5, 7 and 9 values for K constant are tested, but for the K=3 value a high classification performance has been achieved.

2.3 Self-Organizing Maps

SOM, a model of artificial neural networks and using the unsupervised learning method, was developed in 1982 by Tuevo Kohonen [41]. SOM uses competitor learning algorithm indicated in [42]. In this way, the neurons of the network compete each other to become active and ultimately only one neuron win the race. Here, the basic goal is to transform adaptively the n-dimensional input space into two-dimensional map of the output nodes (see Figure 1). After training is completed, a label is assigned to the nodes with a labeling method.

In this method, *j*th node (w_j) is in the output layer and input vector x is in the input layer. The distance between the *j*th node and x is calculated as follows:

$$x = [x_1, x_2, \dots, x_n]^T \; ; \; w_j = \left[w_{j1}, w_{j2}, \dots, w_{jn}\right]^T \quad (2)$$
$$D_j = \sum_{i=1}^{n} \left(x_i - w_{ji}(k)\right)^2$$

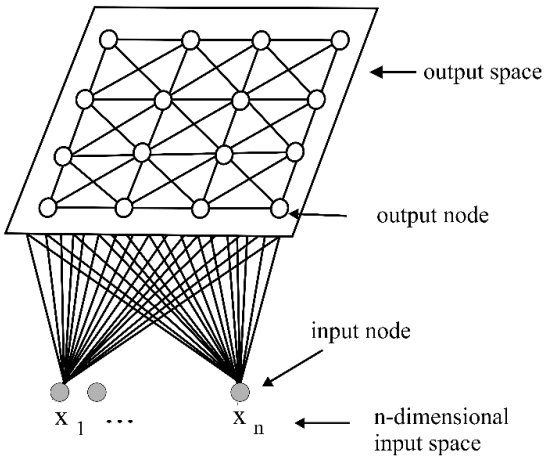

Figure 1. SOM network structure

where *n* shows the dimension of feature vector.

The weights of the output node and its neighbors are updated as follows.

$$w_{ji}(k+1) = w_{ji}(k) + \eta(k).(x_i - w_{ji}(k)) \qquad (3)$$

Here, η is learning rate and k is the number of iterations. The details of training algorithm of Kohonen's SOM network can be seen in [42].

2.4 Grow and Learn Network

In grow and learn (GAL) network [43], class boundaries are determined by the closest distance measure. The distances between the input vector and all the nodes in the network are calculated. The class of the input vector is determined as the class of the network node closest to this vector. The number of nodes in the network is automatically determined and updated during training if necessary. The network grows when it learns and becomes smaller when it forgets.

The structure of the GAL network is shown in Figure 2. The GAL network consists of two layers. The structure of the nodes in the network is expressed by the following equations.

$$D_j = \sum_{i=1}^{n} \left(x_i - w_{ji}\right)^2$$

$$E_e = \begin{cases} 1, & D_e = \min_j (D_j) \\ 0, & otherwise \end{cases} \qquad (4)$$

$$T_{ec} = \begin{cases} 1, & if\ e\ is\ an\ exemplar\ of\ class\ c \\ 0, & otherwise \end{cases}$$

$$C_c = \sum_{e} E_e.T_{ec}$$

E_e is the output of the nodes in the first layer, and T_{ec} is the link coefficient that indicates the OR operation that takes only 0 or 1 values.

The first layer is used to find the minimum distance between the node weights and the input vector, while the second layer is used to define the class to which the nodes in the network belong. The weights in the second layer initially have a value of 0, which is 1 during training. The second layer is used to logically OR the outputs of the same class.

The most important property of the GAL network is that the number of nodes can be automatically specified depending on the structure of the problem (distribution of classes) during training. The structure of the GAL network is highly dependent on the order of the input vectors initially given to the network. There are nodes in the network which are meaningful before, but which have lost meaning with the addition of new nodes to the network. These nodes created during the training algorithm are removed from the network by the forgetting algorithm.

The purpose of the forgetting algorithm is to find nodes that do not affect the success rate of the network when it is

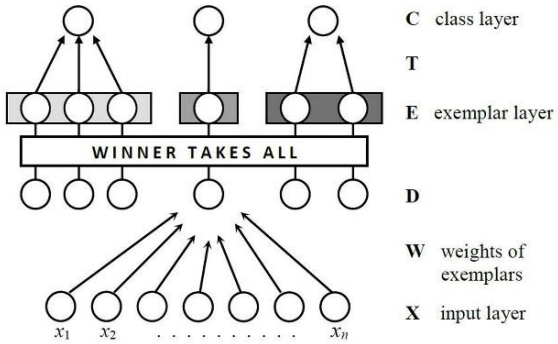

Figure 2. GAL network structure

removed from the network and to remove those nodes from the network. The training and forgetting algorithm of the GAL network [44] is described in detail.

2.5 Divergence Analysis

Divergence analysis is applied to select the best features that do not degrade performance in the desired number of all the features used in problems with two or more classes. In divergence calculations, within-class scatter matrix and between-class scatter matrix are used as criteria for class separation [45].

The selection of the best *d* feature subset ($d < n$) from *n* attributes can be done by Divergence Analysis. For example, in this method, firstly one most effective attribute is determined from *n* features. This feature is a feature that will be placed in the *d* element final set. After the most effective feature is found, second most effective feature is determined among the remaining (*n-1*) features so that these two features maximize the separability criterion. This process continues until the *d* elementary feature set is obtained.

2.6 Performance Analysis

In this paper, to improve the performance, OvO (One-versus-Others) method was used with KNN, GAL and SOM

networks. This method is used in multi-class problems and transforms K-class problem into a two-class problem. One class contains all the proteins belonging to the *i*-th fold that are labeled as positive, and the other class contains all other proteins that are labeled as negative. Thus K binary classifiers are constructed to predict the protein folds.

In the tests for classifying the protein at the fold level, to calculate individual fold success rate (IFSR), the sensitivity, which is mostly used in the literature, was used as seen in Equation (5).

$$IFSR = \frac{\sum True\ Positive}{\sum True\ Psitive + \sum False\ Negative} \quad (5)$$

In order to calculate the overall success rate (OSR), the sensitivity formula is generalized for the 27 fold class and the overall success rate is calculated as seen in Equation (6).

$$OSR = \frac{\sum_{n=1}^{27} True\ Positive}{\sum_{n=1}^{27} True\ Positive + \sum_{n=1}^{27} False\ Negative} \quad (6)$$

3. Experimental Results

In the first experiment, KNN classifier was tested for the K=3 value to determine the most significant attribute for protein fold classification. For this purpose, firstly only C (amino acid composition) attribute was used, namely the feature vector only consisted of C attribute and the number of feature vector components was 20. Then S and other four attributes were used individually to determine which attribute is more effective; and the number of feature vector components were 21 for these five attributes. The classification performances obtained for these tests are shown in Table 2. The last column in Table 2 demonstrates the results with the contribution of all six attributes.

According to Table 2 the most significant attribute is C; and the second most significant attribute is S.

In the second experiment, SOM network was used. To train SOM network 15×15 neurons was used; and the neighborhood spread was selected as σ=1, learning rate was determined as η=0.5 , lastly iteration number was defined as λ=3000. For each test the average performance over 50 runs is reported. In the last experiment GAL network was used as classifier to determine the most significant attribute for protein fold classification. In the test process, iteration number was determined as 1500. For each test the average performance over 50 runs is given. The results related to performance of SOM and GAL networks are shown in Table 3 and 4, respectively.

According to Table 3 it is observed that attribute C is the most effective and S is the second effective attribute for fold classification. Same conclusions can be made for Table 4.

Table 2. Performance of KNN for each attribute at 27-class protein fold classification

Fold	C	S	H	P	V	Z	All
1	83.3	66.6	66.6	66.6	50	33.3	50
2	33.3	22.2	33.3	22.2	11.1	22.2	33.3
3	35	35	35	20	25	25	40
4	37.5	12.5	25	25	25	25	25
5	77.7	66.6	33.3	55.5	33.3	44.4	66.6
6	33.3	0	22.2	11.1	0	0	0
7	40.9	31.8	31.8	31.8	38.6	36.3	43.1
8	16.6	33.3	8.3	16.6	33.3	16.6	25
9	53.8	15.3	30.7	23	38.4	46.1	46.1
10	33.3	16.6	0	16.6	16.6	16.6	16.6
11	12.5	0	25	37.5	50	37.5	25
12	10.5	21	15.7	26.3	21	26.3	21
13	50	50	25	25	50	50	25
14	50	25	25	25	25	25	25
15	42.8	28.5	28.5	57.1	14.2	28.5	42.8
16	75	70.8	45.8	52	56.2	56.2	81.2
17	50	66.6	33.3	33.3	25	33.3	50
18	23	38.4	15.3	23	15.3	15.3	23
19	29.6	22.2	25.9	22.2	22.2	11.1	18.5
20	41.6	25	50	58.3	33.3	41.6	41.6
21	25	37.5	25	37.5	12.5	25	25
22	41.6	50	50	33.3	50	50	66.6
23	57.1	42.8	42.8	42.8	57.1	28.5	57.1
24	25	50	75	25	75	25	50
25	25	37.5	12.5	12.5	25	37.5	37.5
26	25.9	14.8	14.8	25.9	18.5	14.8	14.8
27	88.8	59.2	25.9	22.2	40.7	18.5	44.4
Suc. Rate	**44.4**	**37.3**	**30.3**	**31.9**	**33.2**	**30.5**	**40.5**

When considered Table 2, 3 and 4, it can be said that the best classification performance is obtained with GAL; and KNN algorithm is not very well for protein fold classification.

Also, SOM has remarkable results for protein classification at the fold level. Also, it is obvious that the highest success rate can be achieved by the contribution of all six attributes.

Above, to determine the most significant attribute and decrease the dimension of the feature vector the feature blocks including 20 or 21 features were considered. Here, each of the 125 features was individually considered with dynamic programming and the divergence values of each one was calculated. Then, 125 features were put in order according to their significance and; the proteins were classified using the best 10, 30 and 50 features with GAL. The classifier wasn't performed for more than 50 features because as seen from Table 5 very close classification performance (80.9%) to that of Table 4 (81.2%) was obtained. Test results are shown in Table 5.

Table 3. Performance of SOM for each attribute at 27-class protein fold classification

Fold	C	S	H	P	V	Z	All
1	100	83.3	83.3	83.3	83.3	83.3	100
2	77.7	55.5	88.8	88.8	88.8	66.6	88.8
3	55	50	50	45	45	55	80
4	87.5	37.5	75	75	75	62.5	87.5
5	88.8	66.6	77.7	66.6	66.6	66.6	100
6	55.5	55.5	55.5	33.3	33.3	66.6	66.6
7	61.3	54.5	54.5	54.5	54.5	59	72.7
8	50	41.6	41.6	50	50	50	58.3
9	84.6	61.5	53.8	61.5	61.5	53.8	84.6
10	83.3	33.3	50	50	50	50	50
11	62.5	37.5	62.5	62.5	62.5	75	75
12	52.6	36.8	57.8	47.3	47.3	47.3	52.6
13	100	50	50	75	75	75	100
14	100	50	75	50	50	50	75
15	100	57.1	85.7	71.4	71.4	71.4	85.7
16	75	68.7	62.5	62.5	62.5	58.3	72.9
17	66.6	66.6	50	50	50	50	66.6
18	53.8	46.1	46.1	38.4	38.4	38.4	61.5
19	66.6	44.4	37	37	37	37	63
20	75	58.3	58.3	41.6	41.6	58.3	66.6
21	62.5	87.5	62.5	50	62.5	50	87.5
22	66.6	66.6	58.3	58.3	58.3	58.3	75
23	71.4	71.4	71.4	71.4	71.4	57.1	71.4
24	100	50	100	100	100	75	75
25	62.5	62.5	37.5	50	50	50	62.5
26	44.4	55.5	37	44.4	44.4	33.3	55.5
27	100	74	48.1	48.1	48.1	48.1	100
Suc. Rate	**69.7**	**57.2**	**55.6**	**54.1**	**54.3**	**53.8**	**73.4**

4. Conclusion

In this paper protein fold classification is considered and the most significant attribute and features are determined for protein classification at fold level. Six attributes are tested individually with KNN, SOM and GAL classifiers to determine the effectiveness of them. In KNN, examples are classified based on the class of their nearest neighbors.

SOM uses competitive learning algorithm. In this algorithm the network neurons compete to be activated and eventually only one neuron wins the race. The goal is to train the network and to project the high dimensional data on to low dimensional map in an adaptive way. GAL is an incremental neural network for supervised learning, the number of nodes in the network is automatically determined and updated during training if necessary. The network enlarges when it learns and becomes smaller when it forgets. The results show that the amino acid composition (C) attribute is the most effective attribute and predicted secondary structure (S) is the second most effective one for all three classifiers at protein fold classification. C attribute gives a reasonable success rate of 71.3% even tested alone with GAL. Also it is obtained that GAL has better classification performance than SOM and KNN.

Table 4. Performance of GAL for each attribute at 27-class protein fold classification

Fold	C	S	H	P	V	Z	All
1	83.3	83.3	66.6	100	83.3	83.3	100
2	77.7	66.6	66.6	66.6	77.7	44.4	100
3	60	75	50	80	90	40	80
4	87.5	75	25	100	75	62.5	87.5
5	100	88.8	55.5	55.5	88.8	77.7	100
6	66.6	44.4	44.4	55.5	66.6	22.2	77.7
7	65.9	93.1	65.9	63.6	68.1	63.6	70.5
8	50	83.3	75	41.6	25	33.3	75
9	92.3	92.3	76.9	46.1	61.5	53.8	84.6
10	66.6	16.6	83.3	33.3	66.6	83.3	66.7
11	62.5	62.5	25	87.5	75	62.5	87.5
12	36.8	47.3	42.1	47.3	78.9	42.1	52.6
13	75	100	75	50	50	100	100
14	75	25	25	25	25	50	75
15	85.7	57.1	85.7	71.4	71.4	57.1	85.7
16	79.1	60.4	83.3	79.1	50	79.1	85.4
17	66.6	75	83.3	75	41.6	50	75
18	61.5	76.9	84.6	69.2	38.4	30.7	84.6
19	66.6	40.7	55.5	59.2	44.4	44.4	81.5
20	83.3	58.3	83.3	66.6	50	100	75
21	62.5	50	62.5	62.5	62.5	62.5	87.5
22	91.6	75	75	58.3	66.6	66.6	91.7
23	85.7	85.7	71.4	85.7	100	71.4	100
24	75	25	50	100	75	75	100
25	50	50	62.5	37.5	37.5	37.5	75
26	51.8	55.5	44.4	48.1	37	25.9	66.7
27	100	70.3	81.4	51.8	37	33.3	100
Suc. Rate	**71.3**	**66.6**	**65.3**	**63.4**	**58.0**	**54.8**	**81.2**

In addition the effectiveness of the 125 features were studied. In order to reduce the feature vector dimension divergence analysis was applied. The goal of divergence analysis application is that the performance ratio does not change. This method computes divergence values of the features and sorts them by their importance. Here, after divergence analysis, the most significant 10, 30 and 50 features were determined, and they were presented to GAL network. For the most significant 50 features, 80.9% classification performance was achieved, therefore no longer tested. As a result, the feature vector dimension was decreased from 125 to 50 without much decreasing success rate, and so the computational load was reduced.

In the next step of the study, classification performance related to protein folds can be enhanced. The methods can be tested on larger datasets. For better results, new features and new network structures can be analyzed.

Table 5. Performance of the GAL classifier using different dimensional feature vectors formed by divergence analysis

Fold No	Dim=10	Dim=30	Dim=50
1	83.3	100	100
2	88.8	100	100
3	75	85	90
4	87.5	100	87.5
5	100	100	100
6	44.4	77.7	77.7
7	68.1	70.4	70.4
8	33.3	75	75
9	69.2	92.3	92.3
10	66.6	83.3	100
11	75	87.5	87.5
12	47.3	68.4	57.8
13	75	100	100
14	50	75	75
15	57.1	71.4	71.4
16	68.7	87.5	81.2
17	58.3	91.6	91.6
18	61.5	76.9	76.9
19	59.2	74	70.3
20	58.3	83.3	75
21	75	87.5	75
22	83.3	83.3	100
23	57.1	71.4	100
24	50	75	100
25	62.5	75	100
26	37	59.2	55.5
27	81.4	88.8	96.2
Suc. Rate	**65.0**	**80.2**	**80.9**

Acknowledgment

This work is supported by the Scientific Research Project Fund of Cumhuriyet University under the project number TEKNO-011, Turkey.

References

1. Hashemi, H.B., Shakery, A., Naeini, M.P, *Protein fold pattern recognition using Bayesian ensemble of RBF neural networks*, in SOCPAR2009: Malaysia. p. 436-441.
2. Cantoni, V., Ferone, A., Ozbudak, O. and Petrosino, A., *Searching structural blocks by SS exhaustive matching*, Lecture Notes in Bioinformatics. Leif Peterson, Giuseppe Russo, Francesco Masulli (Eds.), 2013. p. 57-69.
3. Protein Data Bank, http://www.rcsb.org, last access date: 31.12.2018.
4. Murzin, A.G., Brenner, S.E., Hubbard, T. and Chothia, C., *SCOP: A structural classification of proteins database for the investigation of sequences and structures*, Journal of Molecular Biology, 1995. 247(4), p. 536–540.
5. Dubchak, I., Muchnik, I., Mayor, C., Dralyuk, I. and Kim, S.H., *Recognition of a protein fold in the context of the structural classifications of proteins (SCOP) classification*, Proteins: Structure, Function and Bioinformatics, 1999. 35(4), p. 401–407.
6. Reczko, M. and Bohr, H., *The DEF data base of sequence based protein fold class predictions*, Nucleic acids research, 1994. 22(17), p. 3616-3619.
7. Edler, L., Grassmann, J. and Suhai, S., *Role and results of statistical methods in protein fold class prediction*, Mathematical and Computer Modelling, 2001. 33(12), p. 1401–1417.
8. Ding, C.H.Q. and Dubchak, I., *Multi-class protein fold recognition problem using support vector machines and neural networks*, Bioinformatics, 2001. 17(4), p. 349–358.
9. Bologna, G. and Appel, R.D., *A comparison study on protein fold recognition*, Proceedings of the 9th International Conference on Neural Information Processing, 2002. volume 5, IEEE, p. 2492–2496.
10. Igel, C., Gebert, J. and Wiebringhaus, T., *Protein fold class prediction using neural networks with tailored early-stopping*, , Proceedings of IEEE International Joint Conference on Neural Networks,2004. volume 3, p. 1693–1697.
11. Huang, C.D., Liang, S.F., Lin, C.T. and Wu, R.C., *Machine learning with automatic feature selection for multi-class protein fold classification*, Journal of information science and engineering, 2005. 21(4), p. 711–720.
12. Jazebi, S., Tohidi, A. and Rahgozar, M., *Application of classifier fusion for protein fold recognition*, Sixth International Conference on Fuzzy Systems and Knowledge Discovery, 2009. volume 7, p.171–175.
13. Chinnasamy, A., Sung, W.K. and Mittal, A., *Protein structure and fold prediction using tree-augmented naive Bayesian classifier*, Journal of Bioinformatics and Computational Biology, 2005. 3(04), p. 803–819.
14. Okun, O., *Protein fold recognition with k-local hyperplane distance nearest neighbor algorithm*, Proceedings of the Second European Workshop on Data Mining and Text Mining in Bioinformatics, 2004. Pisa, Italy, Citeseer, p. 51–57.
15. Shen, H.B. and Chou, K.C., *Ensemble classifier for protein fold pattern recognition*, Bioinformatics, 2006. 22(14), p. 1717–1722.
16. Kavousi, K., Moshiri, B., Sadeghi, M., Araabi, B.N. and Moosavi-Movahedi, A.A., *A protein fold classifier formed by fusing different modes of pseudo amino acid composition via PSSM*, Computational Biology and Chemistry, 2011. 35(1), p. 1–9.
17. Kavousi, K., Sadeghi, M., Moshiri, B. and Araabi, B. N.and Moosavi-Movahedi, A.A., *Evidence theoretic protein fold classification based on the concept of hyperfold*, Mathematical Biosciences, 2012. 240(2), p. 148–160.
18. Markowetz, F., Edler, L. and Vingron, M., *Support vector machines for protein fold class prediction*, Biometrical Journal, 2003. 45(3), p. 377–389.
19. Shi, S.Y.M., Suganthan, P.N. and Deb, K., *Multiclass protein fold recognition using multiobjective evolutionary*

algorithms, Proceedings of the IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology, 2004. p. 61–66.

20. Shamim, M.T.A., Anwaruddin, M. and Nagarajaram, H.A., *Support Vector Machine-based classification of protein folds using the structural properties of amino acid residues and amino acid residue pairs*, Bioinformatics, 2007. 23(24), p. 3320–3327.
21. Bindewald, E., Cestaro, A., Hesser, J., Heiler, M. and Tosatto, S.C.E., *MANIFOLD: protein fold recognition based on secondary structure, sequence similarity and enzyme classification*, Protein Engineering, 2003. 16(11), p. 785–789.
22. Nanni, L., *A novel ensemble of classifiers for protein fold recognition*, Neurocomputing, 2006. 69(16), p. 2434–2437.
23. Nanni, L., *Ensemble of classifiers for protein fold recognition*, Neurocomputing, 2006. 69(7), p. 850–853.
24. Chen, K. and Kurgan, L., *PFRES: protein fold classification by using evolutionary information and predicted secondary structure*, Bioinformatics, 2007. 23(21), p. 2843–2850.
25. Guo, X. and Gao, X., *A novel hierarchical ensemble classifier for protein fold recognition*, Protein Engineering Design and Selection, 2008. 21(11), p. 659–664.
26. Chen, P., Liu, C., Burge, L., Mahmood, M., Southerland, W. and Gloster, C., *Protein fold classification with genetic algorithms and feature selection*, Journal of bioinformatics and computational biology, 2009. 7(05), p. 773–788.
27. Yang, T., Kecman, V., Cao, L., Zhang, C. and Huang, J.Z., *Margin-based ensemble classifier for protein fold recognition*, Expert Systems with Applications, 2011. 38(10), p. 12348–12355.
28. Lin, C., Zou, Y., Qin, J., Liu, X., Jiang, Y., Ke, C. and Zou, Q., *Hierarchical classification of protein folds using a novel ensemble classifier*, PloS one, 2013. 8(2), e56499.
29. Aram, R.Z. and Charkari, N.M., *A two-layer classification framework for protein fold recognition*, Journal of Theoretical Biology, 2015. 365, p. 32–39.
30. Damoulas, T. and Girolami, M.A., *Probabilistic multi-class multi-kernel learning: on protein fold recognition and remote homology detection*, Bioinformatics, 2008. 24(10), p. 1264–1270.
31. Huang, C.D., Lin, C.T. and Pal, N.R., *Hierarchical learning architecture with automatic feature selection for multiclass protein fold classification*, NanoBioscience, IEEE Transactions on, 2003. 2(4), p. 221–232.
32. Krishnaraj, Y. and Reddy, C.K., *Boosting methods for protein fold recognition: an empirical comparison*, IEEE International Conference on Bioinformatics and Biomedicine, 2008.
33. Shen, H.B. and Chou, K.C., *Predicting protein fold pattern with functional domain and sequential evolution information*, Journal of Theoretical Biology, 2009. 256(3), p. 441–446.
34. Dehzangi, A., Amnuaisuk, S.P. and Dehzangi, O., *Using random forest for protein fold prediction problem: An empirical study*, J. Inf. Sci. Eng., 2010. 26(6), p. 1941–1956.
35. Dehzangi, A., Amnuaisuk, S.P., Manafi, M. and Safa, S., *Using rotation forest for protein fold prediction problem: An empirical study*, 8th European Conference on Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics, 2010. p. 217–227.
36. Wang, R. and Gao, X., *A Two-Layer Learning Architecture for Multi-Class Protein Folds Classification*, Interdisciplinary Research and Applications in Bioinformatics, Computational Biology, and Environmental Sciences, 2010.
37. Dehzangi, A.and Karamizadeh, S., *Solving protein fold prediction problem using fusion of heterogeneous classifiers*, INFORMATION, An International Interdisciplinary Journal, 2011. 14(11), p. 3611–3622.
38. Suvarnavani, K., Rafiah, S.B. and Kamisetti, N.R., *Multiclass classification for protein fold prediction using Smote*, International Journal of Advanced Research in Computer Science and Software Engineering, 2011. 2(11), p. 290–296.
39. Bae, S.E., Jung, S., Ahn, I. and Son, H.S., *Protein fold classification with backbone torsional characters using multi-class linear discriminant analysis*, J Proteomics Bioinform, 2013. 6, p. 148–152.
40. Duda RO, Hart PE. *Pattern Classification and Scene Analysis*. John-Wiley&Sons. Inc. 1973.
41. Kohonen, T., *Self-Organized Formation of Topologically Correct Feature Maps*, Biological Cybernetics, 1982. 43(1), p. 59-69.
42. Polat O. and Dokur Z., *Protein fold recognition using self-organizing map neural network*, Current Bioinformatics, 2016. 11, p. 451-458.
43. Alpaydın E., *Neural models of incremental supervized and unsupervized learning*, Ds. Thesis, Ecole Polytecnique Federale De Lausanne, Switzerland, 1990.
44. Polat O. and Dokur Z., *Protein fold classification with grow-and-learn network*, Turk J Elec Eng & Comp Sci, 2017. 25, p. 1184-1196.
45. Ölmez, T., Dokur, Z. *Uzman Sistemlerde Örüntü Tanıma: Yapay Sinir Ağları, Genetik Algoritmalar, Bulanık Mantık, Makine Öğrenmesi* ders notu.