# Genetic algorithm for an optimized weighted voting scheme incorporating k-separated bigram transition probabilities to improve protein fold recognition

Harsh Saini[1], Gaurav Raicar[1], Sunil Lal[1], Abdollah Dehzangi[2], James Lyons[2], Kuldip K. Paliwal[2], Seiya Imoto[3], Satoru Miyano[3], Alok Sharma[1,2]

[1]The University of the South Pacific, Fiji
[2]Griffith University, Brisbane, Australia
[3]University of Tokyo, Japan

***Abstract*—In biology, identifying the tertiary structure of a protein helps determine its functions. A step towards tertiary structure identification is predicting a protein's fold. Computational methods have been applied to determine a protein's fold by assembling information from its structural, physicochemical and/or evolutionary properties. It has been shown that evolutionary data helps improve prediction accuracy. In this study, a scheme is proposed that uses the genetic algorithm (GA) to optimize a weighted voting system to improve protein fold recognition. This scheme incorporates k-separated bigram transition probabilities for feature extraction, which are based on the Position Specific Scoring Matrix (PSSM). A set of SVM classifiers are used for initial classification, whereupon their predictions are consolidated using the optimized weighted voting system. This scheme has been demonstrated on the Ding and Dubchak (DD) benchmarked data set.**

***Keywords—Protein fold recognition; SCOP; PSSM; Genetic Algorithm; Support Vector Machines; k-separated bigrams***

## I. Introduction

In the field of biological science, protein fold recognition refers to assigning a protein to one of a finite number of folds. This is considered a crucial transitional step in identifying a protein's tertiary structure [1]. Recognition of protein folds requires development of feature extraction and classification techniques. In literature, several feature extraction and classification techniques have been proposed. Dubchak et al. [2] have used structural and physicochemical based features for protein fold recognition. Taguchi and Gromiha [3] proposed structural features (occurrence) for protein fold recognition. Ghanty and Pal [3] have employed pairwise frequencies of amino acids, both for adjacent (PF2) and separated by one residue (PF1). Yang et al. 2011 [4] have used PF1 and PF2 in an augmented form in their study. Their features, subsequently, were of higher dimensions compared to Ghanty and Pal [3]. Nonetheless, high dimensionality can be controlled by utilizing either feature selection methods (eg. [5]–[10]) or dimensionality reduction methods (eg. [11]–[15]).

Recently in protein fold recognition, the use of evolutionary features have been showing good performance [16], [17]. Evolutionary features are extracted from Position Specific Scoring Matrix (PSSM) and are basically a relative measure of amino acid occurrence at a particular position in the protein sequence. Several researchers have used PSSM for improving protein fold recognition and some of these include auto cross-covariance [18], bi-gram [19], tri-gram [20], sequence alignment via dynamic time warping [21] and ensemble features [22], [23].

Furthermore, some of the classification techniques that have been explored include Linear Discriminant Analysis [24], K-Nearest Neighbors [25], Bayesian Classifiers [26]–[28], Support Vector Machines (SVM) [19]–[21], [28]–[30], Artificial Neural Networks (ANN) [31]–[33] and ensemble classifiers [34], [35]. Out of these mentioned classification techniques, SVM has showed promising results in protein fold recognition problem. However, it is shown in the literature that to further improve the protein folding accuracy, a good combination of features extraction technique as well as classification technique is needed [36], [37].

In this paper, a scheme is proposed where the Genetic Algorithm (GA) is utilized to determine the optimal weights for a weighted voting scheme. In this scheme, features are extracted using k-separated bigrams and classification is carried by utilizing multiple instances of SVM classifiers. The extracted information for each k-separated bigrams is used for classification by a separate SVM classifier whose individual predictions are combined using a weight voting system that has been optimized using the GA [38].

## II. Dataset

In this research, data has been used from the benchmarked Ding and Dubchak (DD) protein sequence dataset. The dataset consists of a training set for the creation of the model and an independent test set for testing queries against the model. The data set belong to 27 SCOP folds which further represent the four major structural classes – α, β, α+β, α/β. A brief listing of the DD dataset has been illustrated in **Table 1**. The training dataset consists of 311 protein sequences where any given pair of sequences do not have more than 35% sequence identity for aligned subsequences longer than 80 residues and the test set consists of 383 protein sequences where the sequence identity between any two given proteins is less than 40% [2].

TABLE 1. SUMMARY OF DD-DATASET

| Fold | Number of training vectors | Number of test vectors |
|---|---|---|
| **α** | | |
| Globin-like | 13 | 6 |
| Cytochromec | 7 | 9 |
| DNA-binding 3-helical-bundle | 12 | 20 |
| 4-Helical up-and-down bundle | 7 | 8 |
| 4-Helical cytokines | 9 | 9 |
| α EF-hand | 6 | 9 |
| **β** | | |
| Immunoglobulin-like - sandwich | 30 | 44 |
| Cupredoxins | 9 | 12 |
| Viral coat and capsid proteins | 16 | 13 |
| ConA-like lectins/glucanases | 7 | 6 |
| SH3-like barrel | 8 | 8 |
| OB-fold | 13 | 19 |
| Trefoil | 8 | 4 |
| Trypsin-like serineproteases | 9 | 4 |
| Lipocalins | 9 | 7 |
| **α/β** | | |
| (TIM)-barrel | 29 | 48 |
| FAD (also NAD)-binding motif | 11 | 12 |
| Flavodoxin-like | 11 | 13 |
| NAD(P)-binding Rossmann-fold | 13 | 27 |
| P-loop containing nucleotide | 10 | 12 |
| Thioredoxin-like | 9 | 8 |
| Ribonuclease H-like motif | 10 | 12 |
| Hydrolases | 11 | 7 |
| Periplasmic binding protein-like | 11 | 4 |
| **α+β** | | |
| β-Grasp | 7 | 8 |
| Ferredoxin-like | 13 | 27 |
| Small inhibitors, toxins, lectins | 13 | 27 |

## III. PROCEDURE

### A. *Overview*

The scheme, initially, commences with the extraction of PSSM from protein sequences using PSI-BLAST. This is succeeded by the calculation the k-separated bigrams and the features are extracted for *k=1,...,11*. Finally, the SVM classifiers provide individual sets of predictions that are consolidated using a GA optimized weighted voting system. A flow-diagram of the classification procedure is illustrated in **Figure 1**.

### B. *Feature Extraction*

The technique proposed in this study attempts to model relationships between amino acids that are non-adjacent in the protein sequence. To accomplish this, amino acid bigram probabilities are extracted from the sequential evolution probabilities in PSSM. In this study, we also explore k-separated bigrams that are non-adjacent. i.e., they are separated by other amino acids in the sequence whereby *k* determines the positional distance between the bigrams under consideration. This technique can be summarized mathematically as shown in **Equation 1**. If we let *N* be the PSSM matrix representation for a given protein, *N* will have *L* rows, where *L* is the length of the primary sequence and 20 columns since there are only 20 unique amino acids. The transition of the m$^{th}$ amino acid to the n$^{th}$ amino acid is:

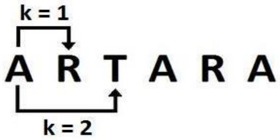
$$T_{m,n}(k) = \sum_{i=1}^{L-k} N_{i,m} N_{i+k,n} \tag{1}$$

*where* $1 \leq m \leq 20,\ 1 \leq n \leq 20$ *and* $1 \leq k \leq K$

$$T(k) = [\ T_{1,1}(k), T_{1,2}(k), \ldots, T_{1,20}(k), T_{2,1}(k), \ldots, T_{2,20}(k), \ldots, T_{20,1}(k), \ldots, T_{20,20}(k)\ ] \tag{2}$$

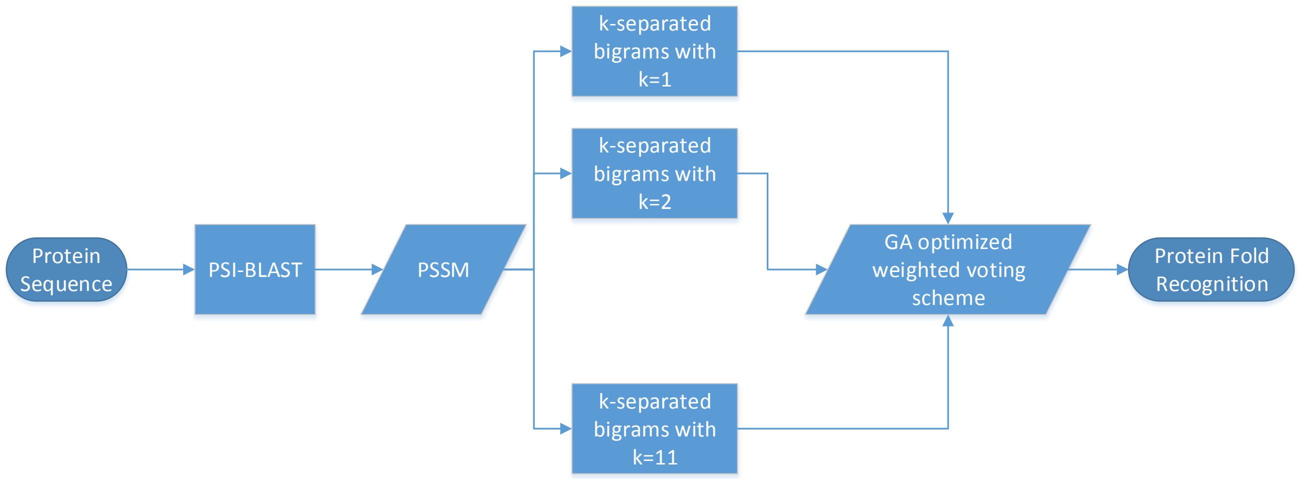
The equation stated in (**1**) constructs a matrix *T(k)* or *F(k)* (for clarity) (**2**) which contains 400 elements representing 400 amino acid transitions for just a single value of *k*. As stated previously, *k* represents the distance between the amino acid positions that are used to compute the transition probabilities. For *k=1*, the transition probabilities are computed between neighboring amino acids whereas, for *k=2*, the transition probabilities are computed between amino acids that are separated by 1 amino acid in between. Therefore, for *k=11*, the amino acids used to calculate the transition probabilities are separated by 10 amino acids. We consider k-separated bigrams only until *k=1,2,...,11* due to the computational constraints and also due to classifier performance discussed in later sections. The k-separated bigrams have been visualized in **Figure 2** for a sample protein ARTARA.

FIGURE 2 - A SAMPLE PROTEIN SEQUENCE WITH ILLUSTRATION OF K-SEPARATED BIGRAMS



We calculate k-separated bigrams only up to *k=1,…,11* due to the fact that the classification accuracy gradually decreases as the *k* gets larger and the weights assigned by the genetic algorithm to each *k* (shown in **Figure 4**) become too small to provide any significant input to the classification. Additionally, the accuracy for each feature set *F(k)* gradually drops as the value of *k* increases (highlighted in **Figure 3**).

FIGURE 1 – FLOW DIAGRAM OF THE PROPOSED CLASSIFICATION PROCEDURE


Protein Sequence
PSI-BLAST
PSSM
k-separated bigrams with k=1
k-separated bigrams with k=2
k-separated bigrams with k=11
GA optimized weighted voting scheme
Protein Fold Recognition

Upon extracting all features with *k=1,2…,11*, we get a feature set of 4400 features (400 * 11 features). Instead of considering all these features a single feature vector of length 4400, we consider features extracted for a particular value of *k* as separate and independent feature vector. This is partly due to the high computational requirements of processing the concatenated feature vector and the increased difficulty in classification with a feature vector of high dimensionality. Moreover, this approach will also clearly highlight the contribution by individual k-separated bigrams. Therefore, for each value of *k*, we consider the extracted features to be representing the bigram transition probabilities with varying k-separation between the amino acid pairs. Each of these 11 different feature sets provides input for the set of SVM classifiers discussed in the later sections.

In order to illustrate this feature extraction technique, let us consider a fictional protein sequence ARTARA of length *L=5*, and also assume that there are only 3 amino acids for the purpose of illustration. Therefore, the length of the feature vector for each value of k will be 9 (since 3*3). Assuming that the PSSM matrix for this sequence is given as shown in **Table 2**, the technique described in this paper will be utilized to extract features for *k=1* and *k=2*. If we apply the proposed feature extraction technique with *k=1*, we can compute the bigram transition probabilities as shown in **Table 3**. Similarly, the bigram transition probabilities for *k=2* can be shown in **Table 4**. As described previously, each of these feature sets are considered as independent and they provide input to specific classifier in the ensemble classification procedure.

TABLE 2 - PSSM OF PROTEIN ARTARA

| Amino Acid | A | R | T |
|---|---|---|---|
| **A** | 0.10 | 0.60 | 0.30 |
| **R** | 0.45 | 0.35 | 0.20 |
| **T** | 0.20 | 0.56 | 0.24 |
| **A** | 0.31 | 0.42 | 0.27 |
| **R** | 0.66 | 0.17 | 0.17 |
| **A** | 0.13 | 0.71 | 0.16 |

TABLE 3 - BIGRAM TRANSITION PROBABILITIES FOR PROTEIN ARTARA WITH K=1

| Amino Acid | A | R | T |
|---|---|---|---|
| **A** | 0.4874 | 0.8923 | 0.3403 |
| **R** | 0.8129 | 0.8333 | 0.4538 |
| **T** | 0.4497 | 0.4844 | 0.2459 |

TABLE 4 - BIGRAM TRANSITION PROBABILITIES FOR PROTEIN ARTARA WITH K=2

| Amino Acid | A | R | T |
|---|---|---|---|
| **A** | 0.3318 | 0.4991 | 0.2291 |
| **R** | 0.6527 | 0.8764 | 0.4009 |
| **T** | 0.3155 | 0.4845 | 0.2100 |

### *C. Classification on Individual Features*

SVM was used for classifying each of the feature sets represented by *F(k)*. SVM is a supervised learning model linked to machine learning algorithms that is used for pattern recognition. It is widely used in classification and regression analysis. In its simplest form, SVM accepts a set of inputs and then predicts for each input which of the two possible classes it falls under. For multi-class problems, SVM can still be used by reducing the problem into multiple binary classification problems. SVM aims to construct a hyper-plane in infinite-dimensional space such that a good level of separation is achieved between the classes thus lowering the generalization error of the classifier.

A different instance of SVM was used to build the classification model for every different *F(k)*. Therefore, in this scheme we have 11 different instances of SVM each of which is trained with a particular feature set, *F(k)*. As stated earlier, this is done so to mitigate the problem of high dimensionality with the concatenated feature set. The parameters for each of the SVM classifiers were tuned such that the classifiers yielded the highest training accuracies.

### *D. Fusion on Classifier Outputs*

Since the classification model consisted of several instances of SVM classifiers, all of which provided their own sets of predictions, it was important to formulate a scheme to assimilate the predictions from the classifiers efficiently to produce one consolidated set of predictions. Considerable experimentation was carried out to determine the optimal scheme, which included simple majority voting, selection of best performing feature sets from *F(k)* and genetic algorithm optimized weighted voting.

Initially, a simple majority voting system was used for determining the final prediction using all 11 SVM models. The classification accuracy of this scheme had a slight improvement over the highest individual performance of the SVM classifier. However, it was determined that not all classifiers were required equally for improving classification accuracy. Therefore, a new scheme was created whereby a subset of 5 best performing classifiers were selected based on their training accuracies and then simple majority voting was used to determine the final set of predictions, which resulted in further improvement in the classification accuracy.

However, this approach of selecting a subset of classifiers had its shortcomings. It was difficult to determine the optimal number of classifiers that need to be selected to yield the best results. Additionally, by removing certain classifiers from the final prediction calculation, we were effectively removing the input from the entire feature set *F(k)* for that particular value of *k*. Therefore, we pursued a more holistic approach whereby an evolutionary machine learning technique, Genetic Algorithm, was used to assist in consolidating the predictions.

The evolutionary approach to machine learning is based on computational models which includes natural selection and genetics. This is known as evolutionary computation which simulates evolution on a computer and encompasses genetic algorithms, evolutionary strategies and genetic programming. The latter techniques simulate evolution using selection, mutation and reproduction processes. Genetic Algorithm (GA) is basically an optimization algorithm which iteratively improves the quality of a solution until is optimal using a stochastic approach.

In the proposed scheme, GA was used to optimize weights for voting assigned to each classifier that ranged from 0 to 1, with 0 indicating no input for the final prediction and 1 indicating maximum input towards the final prediction. The final prediction was simply based on selecting the output variable with the highest weight. This has been summarized in **Algorithm 1** shown below.

ALGORITHM 1 - WEIGHTED VOTING SYSTEM

```
Loop through each class
    Sum the weighted votes by each classifier for that class
End Loop
Predicted Class := Class with greatest votes
```

To determine the optimal weights for each classifier, real value genetic algorithm was used. The *Optimization Toolbox* from *MATLAB* provided such an implementation of the genetic algorithm for minimization problems. Since the aim was to maximize the classification accuracy, the fitness function had to be modified accordingly to represent the problem as a minimization function. The fitness function (**Equation 3**) and the genetic algorithm parameters are provided in **Table 5**. The genetic algorithm was allowed to terminate after the fitness values remained constant for 100 generations. On an average, the genetic algorithm ran for 150 generations.

TABLE 5 - GENETIC ALGORITHM PARAMETERS AND VALUES

| **Parameter** | **Value** |
|---|---|
| Number of Generations | 1000 |
| Stall Limit | 100 |
| Population Size | 100 |
| Crossover Rate | 0.8 |
| Crossover Function | Two Point Crossover |
| Mutation Function | Adaptive Feasible |
| Initial Population Template | [1 1 1 1 1 1 1 1 1 1 1] |

## IV. RESULTS AND DISCUSSION

The experimentation was performed on the benchmarked DD dataset to evaluate the performance of the classification scheme described previously. As previously stated, the DD dataset, which been widely adopted by many researchers, has divided the samples into two sets, training and testing datasets. The classification model described in this study was evaluated using independent testing using the test dataset and k-fold cross-validation testing. This strategy of performance evaluation is widely employed by researchers in literature. However, the jackknife test was not performed due to high computational demands.

The performance of SVM on each of the individual feature sets *F(k)* with *k=1,2,…,11* was encouraging. The results, shown in **Table 6**, clearly indicate that there is discriminatory information present in k-separated bigrams even when the bigrams are separated by 10 amino acids in between. In order to consolidate the individual classifier predictions, we explored simple majority voting, subset selection with majority voting and weighted voting approaches, the results of which are reported in **Table 7**. As shown by the results, each of these scheme lead to an improvement on the highest individual performance by SVM, however, GA optimized weighted voting scheme leads to the highest improvement in performance. The optimized weights, shown in **Figure 4**, yielded **72.7%** training accuracy. It should be noted that all evaluations up till this stage were carried out by using the training dataset only unless otherwise stated and these results were analyzed to determine the various parameters. Upon finalizing optimal parameters and weights, the model was now ready for evaluations using the standardized test dataset.

The experiment consisted of two parts; in the first part, the parameters for the model were optimized using the training set and the separate test set was employed to determine the classification accuracy of the model. In the second part, the training and testing datasets were combined and n-fold cross validation was used to evaluate the performance of the model. For this method of performance evaluation, the parameter values determined previously in the independent dataset test were used.

FIGURE 3 – GRAPH OF TRAINING ACCURACY OF F(K) FOR K=1,2,…,11

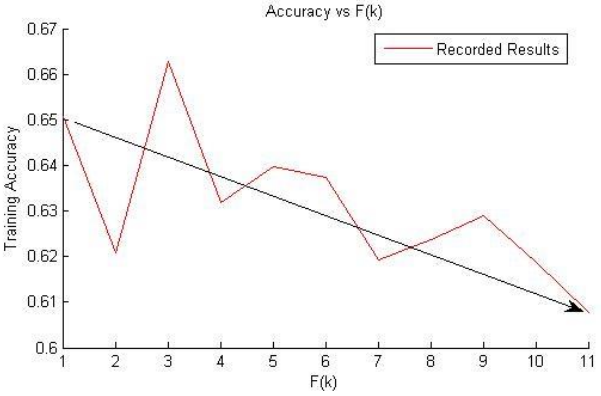


TABLE 6 - CLASSIFICATION ACCURACIES FOR K-SEPARATED BIGRAMS FOR F(K)

| F(k) | Training Acc. | Testing Acc. |
|---|---|---|
| 1 | 65.1 | 68.9 |
| 2 | 62.1 | 68.9 |
| 3 | 66.3 | 68.9 |
| 4 | 63.2 | 65.0 |
| 5 | 64.0 | 66.1 |
| 6 | 63.7 | 67.4 |
| 7 | 61.9 | 65.5 |
| 8 | 62.4 | 65.3 |
| 9 | 62.9 | 65.5 |
| 10 | 61.9 | 66.3 |
| 11 | 60.8 | 65.0 |

FIGURE 4 - SUMMARY OF OPTIMIZED WEIGHTS

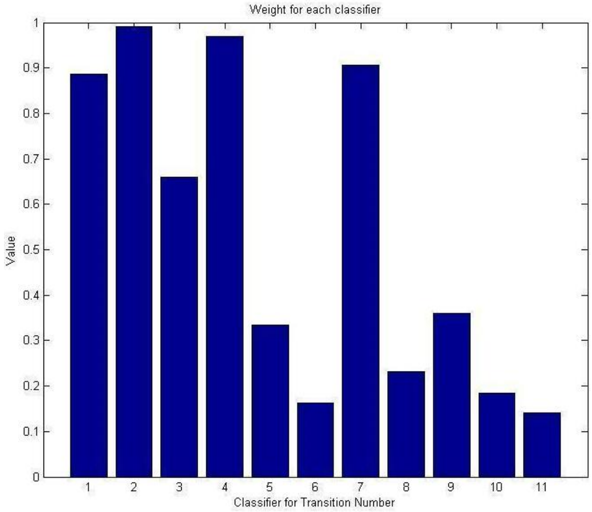


TABLE 7 - SUMMARY OF MODEL PERFORMANCE

| Method | Training Accuracy (%) |
|---|---|
| All 11 classifiers, equal weights | 68.8 |
| 5 top classifiers, equal weights | 70.4 |
| All 11 classifiers, GA optimized weights | 72.7 |

For the first part (using the training set and the independent test set), the performance of the proposed technique showed improvement compared to previous works in literature. It was noted that the highest accuracy achieved was **71.5%** (this paper) which showed a 2% improvement compared to existing feature extraction techniques. The results are summarized in **Table 8**.

TABLE 8 - SUMMARY OF REPORTED TEST ACCURACIES

| Feature set | Accuracy (%) |
|---|---|
| ACC+HXPZV (Ding and Dubchak, 2001) | 56.0 |
| Shamim et al., (2007) | 60.5 |
| Ghanty and Pal (2009) | 59.2 |
| Chmielnicki and Stapor (2012) | 62.8 |
| AHVPZ (Yang et al., 2011) | 44.7 |
| AX (Yang et al., 2011) | 40.3 |
| AHXPZV (Yang et al., 2011) | 49.4 |
| PF (Yang et al., 2011) | 60.8 |
| AHVPZ+PF (Yang et al., 2011) | 51.2 |
| AHXPZV+PF (Yang et al., 2011) | 52.7 |
| Monogram (Sharma et al., 2013) | 62.1 |
| Bigram (Sharma et al., 2013) | 69.5 |
| **This paper** | **71.5** |

For the second part, n-fold cross validation procedure was performed on the merged training and test dataset for *n=5, 6, …, 10*. The same experimental procedure as the first part was adopted in this part which included the ensemble of classifiers with optimal parameters as illustrated in **Table 9**.

The proposed scheme was compared against various other schemes that use information structural and evolutionary information for fold recognition. These techniques included PF1 and PF2 [3], PF [39], Occurrence (O) [40], AAC and AAC+HXPZV [2], which compute feature sets from the original protein sequences. In addition, ACC [18], Bi-gram [19], Tri-gram [20] and Alignment [21] are also included since they compute features directly from the evolutionary information present in PSSM. Moreover, features have been computed from the consensus sequences for PF1, PF2, O, AAC and AAC+HXPZV to obtain additional feature sets for comparison. A prefix of PSSM+ indicates that the features have been computed on the consensus sequence.

The optimal weights that were achieved using GA in the first part were used to combine the classification outputs. The accuracies were compared to past literature works and it was seen that there was an improvement of 2.0%. The highest classification accuracy recorded was **76.7%** (this paper).

Table 9 - Summary of n-fold cross validation procedure

| Feature sets | n=5 | n=6 | n=7 | n=8 | n=9 | n=10 |
|---|---|---|---|---|---|---|
| PF1 | 48.6 | 49.1 | 49.5 | 50.1 | 50.5 | 50.6 |
| PF2 | 46.3 | 47 | 47.5 | 47.7 | 47.9 | 48.2 |
| PF | 51.2 | 52.2 | 52.6 | 52.9 | 53.4 | 53.4 |
| O | 49.7 | 50.4 | 50.8 | 50.8 | 51.1 | 51.0 |
| AAC | 43.6 | 43.9 | 44.2 | 44.8 | 44.6 | 45.1 |
| AAC+HXPZV | 45.1 | 46.2 | 46.5 | 46.8 | 46.9 | 47.2 |
| PSSM+PF1 | 62.5 | 63.2 | 63.7 | 64.2 | 64.5 | 64.6 |
| PSSM+PF2 | 62.7 | 63.3 | 64.1 | 64.2 | 64.6 | 64.7 |
| PSSM+PF | 65.5 | 66.2 | 66.5 | 66.9 | 67.1 | 67.5 |
| PSSM+O | 62.5 | 62.1 | 62.5 | 62.9 | 63.4 | 63.5 |
| PSSM+AAC | 57.5 | 58.1 | 58.4 | 58.7 | 59.1 | 59.2 |
| PSSM+AAC+ HXPZV | 55.9 | 56.9 | 57.1 | 57.7 | 58.0 | 58.2 |
| Bigram | 72.6 | 73.1 | 73.7 | 73.7 | 74.1 | 74.1 |
| Trigram | 72.1 | 72.6 | 73.0 | 73.2 | 73.7 | 73.8 |
| Alignment | 72.6 | 73.5 | 73.8 | 74.2 | 74.7 | 74.7 |
| **This paper** | **73.5** | **75.4** | **74.5** | **76.2** | **76.7** | **75.7** |

## V. Conclusion

In this study, a feature extraction technique has been employed, which is based on sequential evolution probabilities. This technique uses varying distances of amino acid transitions within the sequence to compute the features. A variety of classifiers were used on each transition distance and the predictions of the classifiers were fused using a weighted voted scheme. Genetic Algorithm was further used to optimize the weights so that the best weight distribution can be determined which would give the optimal classification accuracy.

The proposed technique gave promising results, and the accuracy noted was **76.7%** via n-fold cross validation. Since user-friendly and publicly accessible web-servers represent the future direction for developing practically more useful models, simulated methods, or predictors we shall make efforts in our future work to provide a web-server for the method presented in this study.

## References

[1] W. Chmielnicki, “A hybrid discriminative/generative approach to protein fold recognition,” Neurocomputing, vol. 75, no. 1, pp. 194–198, 2012.

[2] C. H. Q. Ding and I. Dubchak, “Multi-class protein fold recognition using support vector machines and neural networks,” Bioinformatics, vol. 17, no. 4, pp. 349–358, 2001.

[3] P. Ghanty and N. R. Pal, “Prediction of protein folds: Extraction of new features, dimensionality reduction, and fusion of heterogeneous classifiers,” NanoBioscience, IEEE Trans., vol. 8, no. 1, pp. 100–110, 2009.

[4] J. Y. Yang and X. Chen, “Improving taxonomy based protein fold recognition by using global and local features,” Proteins Struct. Funct. Bioinforma., vol. 79, no. 7, pp. 2053–2064, 2011.

[5] A. Sharma, K. K. Paliwal, S. Imoto, and S. Miyano, “A feature selection method using improved regularized linear discriminant analysis,” Mach. Vis. Appl., vol. 25, no. 3, pp. 775–786, 2014.

[6] A. Sharma, S. Imoto, S. Miyano, and V. Sharma, “Null space based feature selection method for gene expression data,” Int. J. Mach. Learn. Cybern., vol. 3, no. 4, pp. 269–276, 2012.

[7] A. Sharma, S. Imoto, and S. Miyano, “A between-class overlapping filter-based method for transcriptome data analysis,” J. Bioinform. Comput. Biol., vol. 10, no. 05, 2012.

[8] A. Sharma, S. Imoto, and S. Miyano, “A filter based feature selection algorithm using null space of covariance matrix for DNA microarray gene expression data,” Curr. Bioinform., vol. 7, no. 3, pp. 289–294, 2012.

[9] A. Sharma, S. Imoto, and S. Miyano, “A top-r feature selection algorithm for microarray gene expression data,” IEEE/ACM Trans. Comput. Biol. Bioinforma., vol. 9, no. 3, pp. 754–764, 2012.

[10] A. Sharma, C. H. Koh, S. Imoto, and S. Miyano, “Strategy of finding optimal number of features on gene expression data,” Electron. Lett., vol. 47, no. 8, pp. 480–482, 2011.

[11] A. Sharma, K. K. Paliwal, S. Imoto, and S. Miyano, “Principal component analysis using QR decomposition,” Int. J. Mach. Learn. Cybern., vol. 4, no. 6, pp. 679–683, 2013.

[12] K. K. Paliwal and A. Sharma, “Improved direct LDA and its application to DNA microarray gene expression data,” Pattern Recognit. Lett., vol. 31, no. 16, pp. 2489–2492, 2010.

[13] A. Sharma and K. K. Paliwal, “Rotational linear discriminant analysis technique for dimensionality reduction,” Knowl. Data Eng. IEEE Trans., vol. 20, no. 10, pp. 1336–1347, 2008.

[14] A. Sharma and K. K. Paliwal, “Cancer classification by gradient LDA technique using microarray gene expression data,” Data Knowl. Eng., vol. 66, no. 2, pp. 338–347, 2008.

[15] A. Sharma and K. K. Paliwal, “Fast principal component analysis using fixed-point algorithm,” Pattern Recognit. Lett., vol. 28, no. 10, pp. 1151–1155, 2007.

[16] T. Liu, X. Geng, X. Zheng, R. Li, and J. Wang, “Accurate prediction of protein structural class using auto covariance transformation of PSI-BLAST profiles,” Amino Acids, vol. 42, no. 6, pp. 2243–2249, 2012.

[17] T. Liu and C. Jia, “A high-accuracy protein structural class prediction algorithm using predicted secondary structural information,” J. Theor. Biol., vol. 267, no. 3, pp. 272–275, 2010.

[18] Q. Dong, S. Zhou, and J. Guan, “A new taxonomy-based protein fold recognition approach based on autocross-covariance transformation,” Bioinformatics, vol. 25, no. 20, pp. 2655–2662, 2009.

[19] A. Sharma, J. Lyons, A. Dehzangi, and K. K. Paliwal, “A feature extraction technique using bi-gram probabilities of position specific scoring matrix for protein fold recognition,” J. Theor. Biol., vol. 320, pp. 41–46, 2013.

[20] K. K. Paliwal, A. Sharma, J. Lyons, and A. Dehzangi, “A Tri-Gram Based Feature Extraction Technique Using Linear Probabilities of Position Specific Scoring Matrix for Protein Fold Recognition,” NanoBioscience, IEEE Trans., vol. 13, no. 1, pp. 44–50, 2014.

[21] J. Lyons, N. Biswas, A. Sharma, A. Dehzangi, and K. K. Paliwal, “Protein fold recognition by alignment of amino acid residues using kernelized dynamic time warping,” J. Theor. Biol., vol. 354, pp. 137–145, 2014.

[22] A. Dehzangi and A. Sattar, “Ensemble of diversely trained support vector machines for protein fold recognition,” in Intelligent Information and Database Systems, Springer, 2013, pp. 335–344.

[23] A. Dehzangi, K. Paliwal, J. Lyons, A. Sharma, and A. Sattar, “Enhancing protein fold prediction accuracy using evolutionary and structural features,” in Pattern Recognition in Bioinformatics, Springer, 2013, pp. 196–207.

[24] P. Klein, “Prediction of protein structural class by discriminant analysis,” Biochim. Biophys. Acta (BBA)-Protein Struct. Mol. Enzymol., vol. 874, no. 2, pp. 205–215, 1986.

[25] T.-L. Zhang and Y.-S. Ding, “Using pseudo amino acid composition and binary-tree support vector machines to predict protein structural classes,” Amino Acids, vol. 33, no. 4, pp. 623–629, 2007.

[26] A. Chinnasamy, W.-K. Sung, and A. Mittal, “Protein structure and fold prediction using tree-augmented naive bayesian classifier,” J. Bioinform. Comput. Biol., vol. 3, no. 04, pp. 803–819, 2005.

[27] A. Anand, G. Pugalenthi, and P. N. Suganthan, "Predicting protein structural class by SVM with class-wise optimized features and decision probabilities," J. Theor. Biol., vol. 253, no. 2, pp. 375–380, 2008.

[28] A. Sharma, A. Dehzangi, J. Lyons, S. Imoto, S. Miyano, K. Nakai, and A. Patil, "Evaluation of Sequence Features from Intrinsically Disordered Regions for the Estimation of Protein Function," PLoS One, vol. 9, no. 2, p. e89890, 2014.

[29] A. Sharma, K. K. Paliwal, A. Dehzangi, J. Lyons, S. Imoto, and S. Miyano, "A strategy to select suitable physicochemical attributes of amino acids for protein fold recognition," BMC Bioinformatics, vol. 14, no. 1, p. 233, 2013.

[30] H. Saini, G. Raicar, A. Sharma, S. Lal, A. Dehzangi, A. Rajeshkannan, J. Lyons, N. Biswas, and K. K. Paliwal, "Protein Structural Class Prediction via k-separated bigrams using Position Specific Scoring Matrix," J. Adv. Comput. Intell. Intell. Informatics, vol. 8, no. 4, 2014.

[31] Y.-D. Cai and G.-P. Zhou, "Prediction of protein structural classes by neural network," Biochimie, vol. 82, no. 8, pp. 783–785, 2000.

[32] S. Jahandideh, P. Abdolmaleki, M. Jahandideh, and E. B. Asadabadi, "Novel two-stage hybrid neural discriminant model for predicting proteins structural classes," Biophys. Chem., vol. 128, no. 1, pp. 87–93, 2007.

[33] S. Jahandideh, P. Abdolmaleki, M. Jahandideh, and S. H. S. Hayatshahi, "Novel hybrid method for the evaluation of parameters contributing in determination of protein structural classes," J. Theor. Biol., vol. 244, no. 2, pp. 275–281, 2007.

[34] A. Dehzangi, S. P. Amnuaisuk, K. H. Ng, and E. Mohandesi, "Protein fold prediction problem using ensemble of classifiers," in Neural Information Processing, 2009, pp. 503–511.

[35] K. D. Kedarisetti, L. Kurgan, and S. Dick, "Classifier ensembles for protein structural class prediction with varying homology," Biochem. Biophys. Res. Commun., vol. 348, no. 3, pp. 981–988, 2006.

[36] L. Kurgan, T. Zhang, H. Zhang, S. Shen, and J. Ruan, "Secondary structure-based assignment of the protein structural classes," Amino Acids, vol. 35, no. 3, pp. 551–564, 2008.

[37] L. Kurgan and K. Chen, "Prediction of protein structural class for the twilight zone sequences," Biochem. Biophys. Res. Commun., vol. 357, no. 2, pp. 453–460, 2007.

[38] D. E. Goldberg and J. H. Holland, "Genetic algorithms and machine learning," Mach. Learn., vol. 3, no. 2, pp. 95–99, 1988.

[39] T. Yang, V. Kecman, L. Cao, C. Zhang, and J. Zhexue Huang, "Margin-based ensemble classifier for protein fold recognition," Expert Syst. Appl., vol. 38, no. 10, pp. 12348–12355, 2011.

[40] Y. H. Taguchi and M. M. Gromiha, "Application of amino acid occurrence for discriminating different folding types of globular proteins," BMC Bioinformatics, vol. 8, no. 1, p. 404, 2007.