

# Protein Superfamily Classification Using Kernel Principal Component Analysis And Probabilistic Neural Networks

Swati Vipsita  
Dept. of Computer Science & Engg.  
N.I.T. Rourkela, India  
Email: vipsita.nitrkl@gmail.com

Bithin Kanti Shee  
Dept. of Computer Science & Engg.  
N.I.T. Rourkela, India  
Email: bithinkanti.shee@gmail.com

Santanu Kumar Rath  
Dept. of Computer Science & Engg.  
N.I.T. Rourkela, India  
Email: skrath@nitrkl.ac.in

**Abstract**—This paper intends to implement Probabilistic Neural Network(PNN) for protein superfamily classification problem. The classification task organizes proteins into their superfamilies and helps in correct prediction of structure and function of newly discovered proteins. The two main steps for any pattern classification problem are feature selection and feature extraction. The bi-gram hashing function is used which extracts and counts the occurrences of bi-gram patterns from long strings of amino acid sequences. The bi-gram method maps sequences of different length into input vectors of same length, but the major drawback of this method is that, the size of the input feature vector tends to be very large. Selection of optimal number of features remains a critical issue for any pattern classification problem. Principal Component Analysis(PCA), a very powerful statistical technique, is used to reduce the dimension of the large input vector without much loss of information and thereby identifying pattern in data of high dimension. Traditional PCA makes a linear transformation whereas Kernel PCA(KPCA) is used when data are distributed nonlinearly. Numerical simulations have shown that for protein data distributed non-linearly, KPCA outperforms PCA in terms of accuracy, sensitivity and specificity.

**Keywords**— *Feature extraction, Feature selection, Dimensionality Reduction, Precision, Sensitivity, Specificity, Smoothing parameter, Gaussian Kernel.*

## I. INTRODUCTION

Protein superfamily classification focuses on organizing proteins into their super family and also predicting the family of newly discovered proteins. Correct classification helps in correct prediction of structure and function of new proteins which is one of the primary objective of computational biology and proteomics. Correct prediction of newly discovered proteins mainly concerns the Biologists or researchers for prediction of molecular function, drug discovery, medical diagnosis etc.[1]. Protein classification can be done by classifying a new protein to a given family with previously known characteristics. The aim of classification is to predict target classes for given input protein. There are many approaches available for classification tasks, such as statistical techniques, decision trees and the neural networks. Neural networks have been chosen as technical tools for the protein super family classification task because:

- The extracted features of the protein sequences are

distributed in a high dimensional space with complex characteristics which are difficult to satisfactorily model using some parametrized approaches.

- The rules produced by decision tree techniques are complex and difficult to understand because the features are extracted from long character strings [2].

Classification based on feature extraction from long sequences can be divided into three broad categories. The first category is *feature based classification* in which a sequence is transformed into a feature vector. The second category is *sequence distance based classification* in which the distance function measures the similarity between the sequences. The third category is *model based classification* such as Hidden Markov Model(HMM)[3]. For any protein classification problem, two types of feature such as one related to global structure and other related to local structure are considered. In our work, the n-gram hashing function used is the *bi-gram measure* which assesses the frequency of occurrence of any two amino acid residues consecutively and also of the consecutive occurrence of two amino acids belonging to different substitution groups.

The bi-gram measure gives large feature vector and the overall training matrix tends to be sparse. To reduce the dimension of input matrix, various dimension reduction techniques such as Principal component Analysis(PCA), Singular value decomposition(SVD), Independent Component Analysis(ICA), Non-negative Matrix Factorization(NMF), Eigen decomposition, Random projection, Factor analysis(FA) can be used. PCA is a powerful statistical technique which reduces the dimension of the input matrix without much loss of information and thus identifies pattern in data of high dimension. It always makes a linear transformation but if the distribution structure of the data is *nonlinear* then the points twists and curves its way through p-dimensional space. In such a case, Kernel PCA performs better than linear PCA by extracting significant features.

In our present work, we intend to develop an efficient protein classifier using Probabilistic Neural Network (PNN) and use some dimension reduction techniques such as PCA and KPCA for feature extraction. A PNN is an efficient classifier

as it can map any number of input patterns to their respective classes. PNN learns quickly than any other neural networks and have found success in wide variety of applications such as, pattern recognition problem, human face recognition, speaker verification, solving signal processing problem, classification of soil texture, cloud classification etc.[4]. It basically estimates the probability of class membership by which the network actually learns to estimate the probability density function (p.d.f). Thus, it follows a probabilistic approach based on Bayes decision theory.

## II. RELATED WORK

In previous work, the popular BLAST tool (Altschul et al.,1990) represents the simplest nearest neighbour approach and exploits pairwise local alignments to measure sequence similarity. Another type of direct modeling method is based on Hidden Markov Model (HMM)(Durbin et al.,1998;Karplus et al.,1998). After constructing an HMM for each family, protein queries can be easily scored against all established HMMs by calculating the log likelihood of each model for the unknown sequence and then selecting the class label of the most likely model. The Motif Alignment and Search Tool (MAST) (Bailey and Gribskov, 1998) is based on the combination of multiple motif-based statistical score values. According to this scheme, groups of probabilistic motifs discovered by the MEME algorithm (Bailey and Elkan, 1994), are used to construct protein profiles for the families of interest. The Principal Component null space classifier for protein superfamily classification was implemented in [5]. PCA with Genetic Algorithm(GA) for subset feature selection along with Support vector machine(SVM) was implemented in [6]. In [7],[8] Singular value decomposition (SVD) was used to reduce the dimension of long n-gram feature vectors and then neural networks was implemented using backpropagation training algorithm.

Protein superfamily classification using probabilistic neural network using direct feature extraction was implemented in [9]. Protein Sequences Classification Using Modular RBF Neural Networks which aims to enhance the performance of single neural classifiers based on a centralized information structure in terms of recognition rate, generalization and reliability is implemented in [10]. Multi-class protein classification based on neural networks that maps a protein sequence into a numerical feature space using the matching scores of the sequence to groups of conserved patterns (called motifs) into protein families is implemented in[11]. Protein Classification Using Artificial Neural Networks with Different Protein Encoding Methods is shown in [12]. In [2], Extreme Learning Machine(ELM) was applied for protein sequence classification which has shown significant performance by showing four orders of magnitude less training time compared to BP Network. Standard RBFN and modular RBFN was used to classify protein sequences to multiple classes using n-gram feature extraction in [13].

## III. N-GRAM FEATURE EXTRACTION AND DIMENSION REDUCTION

### A. Feature extraction

Proteins (also known as **polypeptides**) are organic compounds made of amino acids arranged in a linear chain or folded into a globular form. The amino acids are joined together by the peptide bonds between the carboxyl and amino groups of adjacent amino acid residues. In general, the genetic code specifies 20 standard amino acids such as

$$\Sigma = (A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y)$$

. For protein feature selection, the two gram features such as

$$(AA, AC, \dots AY), (CA, CC, \dots CY), \dots (YA, YC, \dots YY)$$

are selected. The total number of possible bi-grams from a set of 20 amino acids is  $20^2$ , that is, 400. The two gram features represent the majority of the protein features. Two grams have the advantages of being length invariant, insertion/deletion invariant, not requiring motif finding and allowing classification based on local similarity [7], [8]. Apart from this, bi-grams reflecting the pattern of substitution of amino acids are also extracted. For this purpose, equivalence classes of amino acids that substitute for one another are derived from the percent accepted mutation matrix (PAM) [14]. Exchange grams are similar but are based on a many to one translation of the amino acid alphabet into a six letter alphabet that represents six groups of amino acids, which represent high evolutionary similarity. Generally the exchange groups used are :

$$e1 = \{H, R, K\}, e2 = \{D, E, N, Q\}, e3 = \{C\},$$

$$e4 = \{S, T, P, A, G\}, e5 = \{M, I, L, V\}, \text{ and } e6 = \{F, Y, W\}$$

. The exchange groups statistically describes the probability of one amino acid replacing another over time. The total number of possible bi-grams on these six substitution groups is  $6^2$ , that is 36. Thus the bigram measure in computation of 436 values, 400 corresponding to the consecutive pairs of amino acids and 36 corresponding to the consecutive pairs of substitution groups. Besides that, the amino acid distribution (20) and exchange group distribution(6) are also taken into account.

Consider the amino acid sequence as MN-NPQMNPQRS. The extracted two-gram features are (MN,2),(NN,1),(NP,2)(PQ,2)(QM,1)(QR,1)(RS,1). The above sequence can be denoted in terms of 6-letter exchange group as

$$e_5 e_2 e_2 e_4 e_2 e_5 e_2 e_4 e_2 e_1 e_4$$

The two gram features of exchange group can be denoted as

$$\{(e_5 e_2, 2), (e_2 e_2, 1), (e_2 e_4, 2), (e_4 e_2, 2), (e_2 e_5, 1), (e_2 e_1, 1), (e_1 e_4, 1)\}$$

Besides the bigram measure and exchange group residues some other features are also extracted such as:

$X^{(1)}, X^{(2)}, \dots, X^{(5)}$  = atomic composition  
 $X^{(6)}$  = molecular weight  
 $X^{(7)}$  = isoelectric point  
 $X^{(8)}$  = average mass of protein sequence  
 $X^{(9)}, X^{(8)}, \dots, X^{(28)}$  = amino acid distribution  
 $X^{(29)}, X^{(30)}, \dots, X^{(428)}$  = two gram distribution  
 $X^{(429)}, X^{(430)} \dots X^{(434)}$  = exchange group distribution  
 $X^{(435)}, X^{(434)} \dots X^{(470)}$  = two gram exchange group distribution.

#### 1) Other Features Extracted ::

- **Atomic composition:** Counts the Carbon, Hydrogen, Nitrogen, Oxygen and Sulphur atoms in the amino acid sequence.
- **Molecular weight:** Molecular weight is calculated by summing up the atomic weights of the atoms making up the protein's molecular formula.
- **Isoelectric point:** The isoelectric point (pI) is the pH at which a particular molecule or surface carries no net electrical charge. At a pH below their pI, proteins carry a net positive charge; above their pI they carry a net negative charge. Proteins can thus be separated according to their isoelectric point (overall charge).
- **Amino acid distribution:** There are twenty standard amino acid bases for a protein sequence. The occurrence of individual frequency of amino acid base gives the distribution of amino acid for a protein sequence.
- **Average Mass of Protein Sequence:** The average mass of a molecule is obtained by summing the average atomic masses of the constituent elements. For example, the average mass of natural water with formula  $H_2O$  is  $1.00794 + 1.00794 + 15.9994 = 18.01528$ .

Therefore, for every amino acid sequence, the **470 features** were processed to build the feature vector as follows:

Thus, a vector of fixed dimension is built for every amino acid sequence. Now, if we assume the length of a amino acid sequence ( $N=50$ ), then the number of bigram feature is  $(N - 1) = 49$ . So, out of 400 feature value ( $20^2$ ),  $400 - 49 = 351$  have zero entries, thereby showing sparseness of the data set. For trigrams of amino acid, it requires  $20^3 = 8000$  input units, which makes the data more sparse. To solve this problem of reducing the size of n-gram vectors, PCA and KPCA are used.

#### B. Principal Component Analysis(PCA)

The concept of PCA was developed by Karl Pearson in 1901. Principal component analysis (PCA) is a statistical technique used to transform a data space of high dimension into a feature space of lower dimension having the most *significant* features. Features (or inputs) that have little variance are thereby removed. PCA is an orthogonal transformation of the coordinate system in which the data are represented. The new transformed coordinate values by which data are represented are called principal components. A small number of principal components(PCs) are sufficient to represent most of the patterns in the data. These are also referred as factors or latent variables of the data. PCA rigidly rotates the axes

of the p-dimensional space to new positions (*principal axes*) such that principal axis 1 has the highest variance, axis 2 has the next highest variance and so on. The covariance among each pair of the principal axes is zero so the principal axes are uncorrelated.

We first compute the covariance matrix  $S$  and find its eigenvalues. The eigenvalues are then sorted in a decreasing order and let they are denoted as  $\lambda_1 \geq \lambda_2 \geq \dots \lambda_M$ . Let the corresponding eigen vectors be denoted as  $a_1 a_2, \dots a_M$ . We choose the first  $d$  eigen vectors from  $M$  vectors such as  $d \ll M$ . Finally, we project the data set into lower dimension as given by:

$$G \leftarrow [a_1 a_2, \dots a_d] \text{ where } d \ll M.$$

if  $x$  is a test point

$$x \in R^M \rightarrow G^T x \in R^d \quad (1)$$

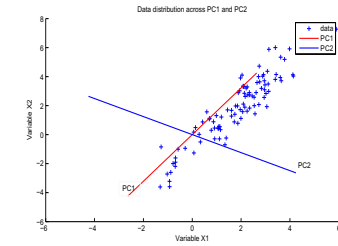


Fig. 1: 2D feature distribution using the first two PC's

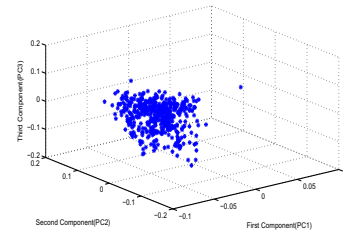


Fig. 2: 3D feature distribution using the first three PC's

#### C. Kernel Principal Component Analysis(KPCA)

Traditional PCA applies linear transformation which may not be effective when data are distributed non-linearly. In such a case, nonlinear PCA is used which is also referred as Kernel PCA. We apply nonlinear transformation to potentially very high-dimensional space by the use of integral operator kernel functions.

$$\phi : x \rightarrow \phi(x)$$

PCA is written here in the form of dot product and any one of the kernel trick is then applied. We decide kernel function a priori which can be either of the following types:

- Polynomial Kernel:  $k(x, y) = (x \cdot y)^d$
- RBF kernel :  $k(x, y) = \exp(-\frac{\|x-y\|^2}{2\sigma^2})$
- Sigmoid :  $k(x, y) = \tanh(k(x, y)) + \Theta$

In our experiment, RBF kernel is used and the kernel matrix using the dot product is computed as :

$$K_{ij} = (\phi(x_i) \cdot \phi(x_j)) \quad (2)$$

Here  $K$  is the symmetric kernel matrix. The non-negative eigenvalues of the  $K$  matrix are found and arranged in decreasing order such as  $\lambda_1 \geq \lambda_2 \geq \dots \lambda_M$  represent the eigenvalues. Let the corresponding set of eigenvectors with respect to their eigenvalues be denoted as  $\alpha^1, \dots \alpha^M$ . The eigenvectors are then normalized in feature space  $F$  such that:

$$(V^k \cdot V^k) = 1$$

$$\text{where } V = \sum_{i=1}^l \alpha_i \phi(x_i) \quad (3)$$

Finally, we compute a low dimension projection for test data by deriving the nonlinear principal components from the eigenvectors  $V^k$  in  $F$  according to :

$$(V^k \cdot \phi(x)) = \sum_{i=1}^M \alpha_i^k (\phi(x_i) \cdot \phi(x)) \quad (4)$$

where  $x$  is the test data with a feature value of  $\phi(x)$ .

#### IV. BRIEF OVERVIEW OF PNN

The concept of PNN was developed by Donald Specht in 1990. The concept of PNN relies on Parzen Window classifier. The Parzen Window is a non-parametric procedure that synthesizes an estimate of probability density function (p.d.f) by superposition of number of windows. The Parzen Window classifier takes a classification decision after calculation of probability density function of each class using the given training samples. The multi-category classifier decision is expressed as follows:

$$P_k f_k > P_j f_j \text{ for all } j \neq k \quad (5)$$

where  $P_k$  is the prior probability of occurrence of examples from class  $k$ .

$f_k$  is the estimated pdf of class  $k$ .

PNN works on the concept of Parzen window and also it is a direct continuation of the work on Bayes Classifier. In original Specht's implementation, the basis function used as window is Gaussian Kernel which is given by:

$$g(x) = \frac{1}{n\sigma} \sum_{k=1}^n \exp \frac{-(x-x_k)^2}{\sigma^2} \quad (6)$$

##### A. PNN architecture and theory of operation

The PNN is a multilayer feedforward network having four layers namely: input layer, hidden or pattern layer, summation layer, output or decision layer. The pattern layer has one pattern node for each training sample. The summation node or unit receives the outputs from the pattern nodes associated with a given class. It simply sums the outputs from the pattern nodes that correspond to the category from which the training pattern was selected. Thus, the number of nodes

in the summation layer is same as the number of classes in multi class classification problem. The output node takes the decision of classifying the unknown sample to its respective class.

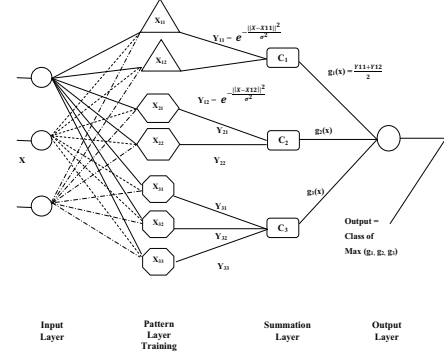


Fig. 3: Brief overview of PNN

If it is a two class classification problem, and the p.d.f of each of the population is known, then an unknown  $X$  belongs to class  $i$  if,

$$f_i > f_j, \quad \text{for all } j \neq i.$$

where  $f_i$  and  $f_j$  are the p.d.f of class  $i$  and  $j$  respectively. In case of three class classification problem, the unknown sample belongs to a class by evaluating the  $\max(f_i, f_j, f_k)$ .

The p.d.f for univariate input can be evaluated as :

$$g_i(x) = \frac{1}{n\sigma} \sum_{k=1}^{n_i} \exp \frac{-(x-x_k)^2}{\sigma^2} \quad (7)$$

where  $n_i$  is the number of training samples belonging to the  $i$ -th class.

For multivariate inputs the p.d.f can be calculated as :

- p.d.f for a single sample in a population:

$$k_i(x) = \frac{1}{(2\pi)^{p/2}(\sigma^p)} \exp \frac{-\|(x-x_k)^2\|}{2\sigma^2} \quad (8)$$

where  $x = \text{unknown input}$

$x_k = k^{\text{th}} \text{ sample}$

$\sigma = \text{smoothing parameter}$

$p = \text{length of vector}$

- p.d.f for a single population

$$h_i(x) = \frac{1}{(2\pi)^{p/2}(\sigma^p)(n_i)} \sum_{k=1}^{n_i} \exp \frac{-\|(x-x_k)^2\|}{2\sigma^2} \quad (9)$$

The smoothing parameter value  $\sigma$  can be guessed based on the knowledge of the data or the value which can be estimated using some heuristic technique.

TABLE I: Details of Training and Test Sequence

Protein Superfamily	No. of training sequences	No. of test sequences
Globin (840)	588	252
Kinase (542)	380	162
Ligase (678)	475	203

## V. MEASURES USED TO ESTIMATE THE PERFORMANCE OF NEURAL NETWORK AS CLASSIFIER :

The measure used to evaluate the performance of the NN classifier:

- 1) Precision =  $(TP+TN) / (TP+FP+TN+FN)$
- 2) Sensitivity =  $(TP/TP+FN)*100\%$
- 3) Specificity =  $(TN/ TN+FP) * 100\%$  Where TP=true positive TN=true negative FP=false positive FN=false negative.

## VI. EXPERIMENT DETAILS

The three superfamilies considered in our experiment are Globin(840), Kinase(542), Ligase(678) from Protein Information Resource (PIR) database. From each family, 70% of total data set formed the training set and the remaining 30% formed the test set. The total number of training sequences counts to 1443 and the total number of test sequences counts to 617. After the input matrix was formed, PCA and KPCA were applied to reduce the dimension of sparse matrix and thus relevant and significant patterns were retrieved from the data set.

TABLE II: Eigenvalues  $\lambda_i$  and variance percentages of first five PCs.

Axis	Eig.val	% of Var	% Cum. Var.
1	4.91	21.98	21.98
2	2.38	10.65	32.63
3	1.18	5.28	37.91
4	0.07	4.79	42.71
5	0.96	4.29	46.99

Each input vector had 470 attributes which includes the bigram distribution, two-gram exchange group distribution and some general features. The dimension reduction techniques transformed a set of correlated input space variables into a lower dimension set of new uncorrelated features. The accuracy were observed in both cases of PCA and KPCA

TABLE III: Eigenvalues  $\lambda_i$  and variance percentages of first five KPCs.

Axis	Eig.val	% of Var	% Cum. Var.
1	6.68	26.72	26.72
2	4.45	17.81	44.53
3	3.33	13.32	57.85
4	2.05	8.2	66.05
5	0.98	3.92	69.97

by projecting the data to lower dimensions by using the first ten principal components. The smoothing parameter  $\sigma$  is the controlling parameter in case of PNN. For all our experiment,  $\sigma$  was assumed to be 0.5. Graphs were plotted showing the variation of accuracy with respect to principal components. The sensitivity vs.  $\sigma$  graphs were plotted for the principal component which has shown the maximum accuracy.

## VII. RESULTS AND DISCUSSION

From the graphs, it is observed that, the maximum accuracy value reached upto 98.67% using KPCA and PNN. As the percentage of variance in the principal components is more in KPCA in comparison to PCA, so many significant patterns were retrieved using KPCA, which increases the classification accuracy of the classifier. Considering the next higher principal component also gave good accuracy but slowly when the PCs were increased, the performance of the classifier degraded as irrelevant and noisy data were included in the training set. It is clear from the graphs, that accuracy falls to 84.67% in PCA and 87.67% in KPCA when the PCs were increased. The value of the smoothing parameter ( $\sigma$ ) has an impact on the performance of the classifier.  $\sigma$  was varied within a range of 0.1-0.9 for the principal components which has shown the maximum accuracy. It was observed in both cases of PCA and KPCA,  $\sigma$  lied within a range of 0.3-0.5, where maximum accuracy values were obtained. The sensitivity value remained constant for most of the value of  $\sigma$  in case of KPCA and PNN but the value has decreased in Ligase and Globin in case of PCA and PNN.

TABLE IV: Clasification accuracy in % across three principal components

	PCA	KPCA
PC1	92	97.33
PC2	95.33	98.67
PC3	94.67	95.67

## VIII. CONCLUSION AND SCOPE OF FUTURE WORK

From the above numerical simulations, it can be concluded that KPCA outperforms PCA when the distribution of data are non-linear. The trade-off between speed and accuracy was also solved using PNN as a classifier. Thus, KPCA with PNN is successfully implemented for protein superfamily classification problem. The two main limitations of PNN are, it requires large amounts of data storge when extensive training sets are available. Secondly, an intrinsic smoothing parameter need to be estimated for better classification performance. Estimation of smoothing parameter value becomes very difficult while implementing for a particular pattern classification. The work can be further extended by implementing some other dimension reduction techniques and the performance of the classifier can be increased by overcoming these two limitations.

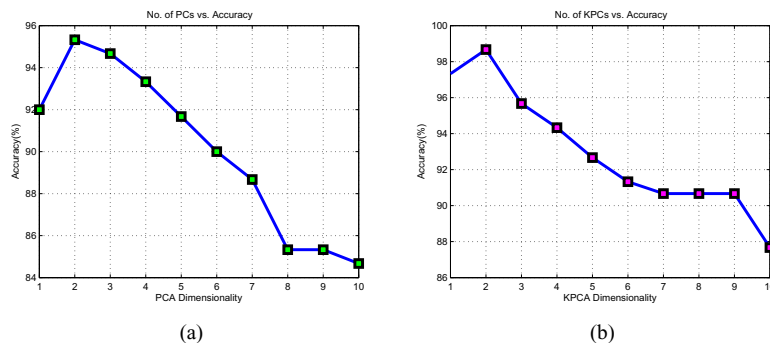


Fig. 4: Plot showing Accuracy vs. No. of PCs using PCA and KPCA respectively.

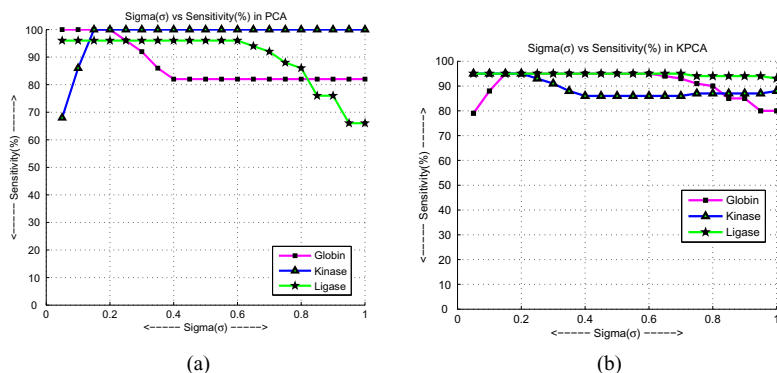


Fig. 5: Plot showing Sensitivity vs. Sigma in PCA and KPCA respectively.

## REFERENCES

- [1] J. H. F. Juliana S Bernardes and A. T. Vasconcelos, "Structural descriptor database: a new tool for sequence based functional site prediction," *BMC Bioinformatics*, vol. 9, no. 1, 2008.
- [2] D. Wang and G. B. Huang, "Protein sequence classification using extreme learning machine," in *Proceedings of International Joint Conference on Neural Networks(IJCNN,2005)*, Montreal, Canada, 2005.
- [3] Z. Xing, J. Pei, and E. Keogh, "A brief survey on sequence classification," *ACM SIGKDD Explorations Newsletter*, vol. 12, no. 1, pp. 40–48, June 2010.
- [4] I.M.M.EIEMary and S.Ramakrishnan, "On the application of various probabilistic neural networks in solving different pattern classification problems," *World applied sciences*, vol. 4, no. 6, pp. 772–780, 2008.
- [5] L. French, A. Ngom, and L. Rueda, "Fast protein superfamily classification using principal component null space analysis," pp. 158–169, 2005.
- [6] Y.-M. C. Xing-Ming Zhaoa and D.-S. Huang, "A novel approach to extracting features from motif content and protein composition for protein sequence classification," *Neural Networks,Elsevier*, vol. 18, pp. 1019–1028, 2005.
- [7] C. Wu and et.al., "Neural networks for molecular sequence classification," *Journal Mathematics and Computers in Simulation,Elsevier*, vol. 40, no. 1-2, p. 10191028, Dec 1995.
- [8] C. Wu, M. Berry, S. Shivakumar, and J. McLarty, "Neural networks for full-scale protein sequence classification: Sequence encoding with singular value decomposition," *Mach. Learn.*, vol. 21, pp. 177–193, October 1995. [Online]. Available: <http://portal.acm.org/citation.cfm?id=218261.218281>
- [9] P. V. N. Rao, T. U. Devi, G. R. Sridhar, and A. A. Rao, "A probabilistic neural network approach for protein superfamilt classification," *Journal of Theoretical and Applied Information Technology*, pp. 101–105, 2005.
- [10] D. Wang, N. K. Lee, T. S. Dillon, and N. J. Hoogenraad, "Protein sequences classification using modular rbf neural networks," in *AI '02: Proceedings of the 15th Australian Joint Conference on Artificial Intelligence*. London, UK: Springer-Verlag, 2002, pp. 477–486.
- [11] K. Blekas, D. I. Fotiadis, and A. Likas, "Motif-based protein sequence classification using neural networks," *Journal of Computational Biology*, vol. 12, no. 1, pp. 64–82, 2005.
- [12] A. Luis and D. Rossi, "Protein classification using artificial neural networks with different protein encoding methods," in *Proceeding ISDA '07 Proceedings of the Seventh International Conference on Intelligent Systems Design and Applications*. IEEE Computer Society, 2007, pp. 169–174.
- [13] Z. Zainuddin and M.Kumar, "Radial basis neural network for protein sequence classification," *Malaysian Journal of Mathematical Sciences*, vol. 2, no. 2, pp. 195–204, 2008.
- [14] S. S. T. Sobha Rani and et.al., "Application of neural networks for protein sequence classification," in *Proceedings of the Conference of International Conference on Intelligent Sensors and Infromation Processing*. IEEE Press, 2004, pp. 325–328.
- [15] I. Jonassen, "Methods for finding motifs in sets of related biosequences," Ph.D. dissertation, Department of Informatics, University of Bergen, 1996.
- [16] D. F. Specht, "Probabilistic neural networks," *Neural Networks*, vol. 3, no. 1, pp. 109–118, 1990.
- [17] F. Gorunescu, "Benchmarking probabilistic neural network algorithms," 2000.
- [18] J. Shlens, "A tutorial on principal component analysis," in *Systems Neurobiology Laboratory, Salk Institute for Biological Studies*, 2005.
- [19] B. Scholkopf, S. Mika, A. Smola, G. Rtsch, and K.-R. Mller, "Kernel pca pattern reconstruction via approximate pre-images," 1998.
- [20] B. Schoolkopf and A. S. K. R. Muller, "Nonlinear component analysis as a kernel eigenvalue problem," Max-Planck-Institute for biologische Kybernetik, Spemannstrae 38,72076 Tübingen,Germany, Tech. Rep. 44, December 1996.
- [21] S. Bandyopadhyay, "An efficient technique for superfamily classification of amino acid sequences: feature extraction, fuzzy clustering and prototype selection," *Fuzzy Sets Syst.*, vol. 152, no. 1, pp. 5–16, 2005.