

**SP06117**



# **IDENTIFICATION AND CLASSIFICATION OF DNA BINDING PROTEINS USING ANN FROM RAW PROTEIN SEQUENCES**

**BY**

**KARISHMA MEHTA-061515**

**NAKSHTRA JAIN-061513**

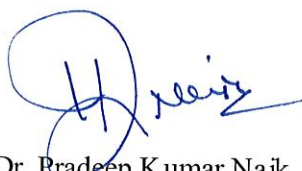


**MAY-2010**

**DEPARTMENT OF BIOTECHNOLOGY & BIOINFORMATICS  
JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY-  
WAGNAGHAT, SOLAN, H.P., INDIA**

## CERTIFICATE

This is to certify that the work entitled, **"Identification and Classification of DNA Binding Proteins Using ANN from Raw Protein Sequences"** submitted by Karishma Mehta (061515) and Nakshatra Jain (061513) in partial fulfillment for the award of degree of Bachelor of Technology in Bioinformatics at Jaypee University of Information Technology has been carried out under my supervision. This work has not been submitted partially or wholly to any other University or Institute for the award of this or any other degree or diploma.



Dr. Pradeep Kumar Naik

(Project Coordinator)

Senior Lecturer

Deptt of Bioinformatics and Biotechnology

Jaypee University of Information Technology

Waknaghat, Distt- Solan

Himachal Pradesh, India

## ACKNOWLEDGEMENT

It has been a great pleasure working under the able guidance of faculty and staff in the Department of Bioinformatics and Biotechnology at Jaypee University of Information Technology, during our study as a B.Tech student. We are very thankful to our project advisor Dr. Pradeep Kumar Naik for his scientific support; otherwise this research work would never have been possible.

Many people have contributed to this project in a variety of ways over the past few months. We express our appreciation to them.



**KARISHMA MEHTA**



**NAKSHTRA JAIN**



## TABLE OF CONTENTS

<b>CERTIFICATE.....</b>	<b>2</b>
<b>ACKNOWLEDGEMENT.....</b>	<b>3</b>
<b>LIST OF FIGURES.....</b>	<b>5</b>
<b>LIST OF TABLES.....</b>	<b>6</b>
<b>LIST OF ABBREVIATIONS.....</b>	<b>7</b>
<b>ABSTRACT.....</b>	<b>8</b>
<b>Chapter 1: INTRODUCTION.....</b>	<b>9-32</b>
• What are DNA Binding Proteins.....	9
• Classification of DNA Binding Proteins.....	11
• Need of Prediction and Classification of DNA Binding Proteins.....	21
• Machine Learning Classification.....	22
• Objective.....	32
<b>Chapter 2: A tool for prediction and classification of DNA Binding Proteins into six major classes using ANN from sequence derived features</b>	<b>33-49</b>
• Abstract.....	34
• Introduction.....	35
• Materials and Methods.....	37
• Discussion.....	48
• Conclusion.....	49
<b>REFERENCES.....</b>	<b>50</b>
<b>APPENDIX 1.....</b>	<b>60</b>
<b>APPENDIX 2.....</b>	<b>61</b>
<b>APPENDIX 3.....</b>	<b>64-75</b>

## **LIST OF FIGURES**

- Figure 1: Showing DNA Binding Protein interaction**
- Figure 2: Showing DNA Binding Protein interaction with Helix-Turn-Helix motif**
- Figure 3: Showing DNA Binding Protein interaction with Helix-Loop-Helix motif**
- Figure 4: Showing DNA Binding Protein interaction with Leucine Zipper motif**
- Figure 5: Showing DNA Binding Protein interaction with Zinc Finger motif**
- Figure 6: Showing DNA Binding Protein interaction with TATA-Box motif**
- Figure 7: Showing DNA Binding Protein interaction with Histone Binding motif**
- Figure 8: Typical Artificial Neural Network Setup**
- Figure 9: Fully connected feed-forward with one hidden layer and one output layer**
- Figure 10: Radial-basic function network**
- Figure 11: Architecture of Layer 1**
- Figure 12: Architecture of Layer 2**



## **LIST OF TABLES**

**Table 1: Binary Classification**

**Table 2: Classification in Six Classes**

**Table 3(a): Percentage of correct prediction in DNA Binding and Non-DNA Binding category for Self-Consistency**

**Table 3(b): Percentage of correct prediction in DNA Binding and Non-DNA Binding category for External Validation**

## LIST OF ABBREVIATIONS

<b>ANN</b>	<b>Artificial Neural Network</b>
<b>NN</b>	<b>Neural Network</b>
<b>AA</b>	<b>Amino Acids</b>
<b>PSEAA</b>	<b>Pseudo Amino Acids</b>
<b>HTH</b>	<b>Helix Turn Helix</b>
<b>HLH</b>	<b>Helix Loop Helix</b>
<b>T-Box</b>	<b>TATA Box</b>
<b>RBF</b>	<b>Radial Basis Function</b>
<b>TBP</b>	<b>TATA Box Binding Protein</b>



## ABSTRACT

The prediction of DNA Binding proteins and their classification into six major classes is one of the most important problems in the biological world. Many previous algorithms have been developed to solve this intricacy. But none of them has been able to provide a reliable method. The previous methods used a much complex method for prediction i.e. from its 3-dimensional structure. It proved out to be a time consuming and a costly method with a lot of limitation. A large number of data are constantly being generated, thanks to several genome-sequencing projects throughout the world. However, the gap between the growth rate of biological sequences and the capability to characterize experimentally the roles and functions associated with these new sequences is constantly increasing. This results in an accumulation of raw data that can lead to an increase in our biological knowledge only if computational characterization tools are developed. The family of DNA-binding proteins is one of the most populated and studied amongst the various genomes of bacteria, archaea and eukaryotes. Understanding the molecular details of protein-DNA interactions is critical for deciphering the mechanisms of gene regulation. We focus here on the annotation of novel protein as DNA/ Non-DNA Binding and if it is a DNA Binding protein, then its classification into six major classes.

## **CHAPTER 1**

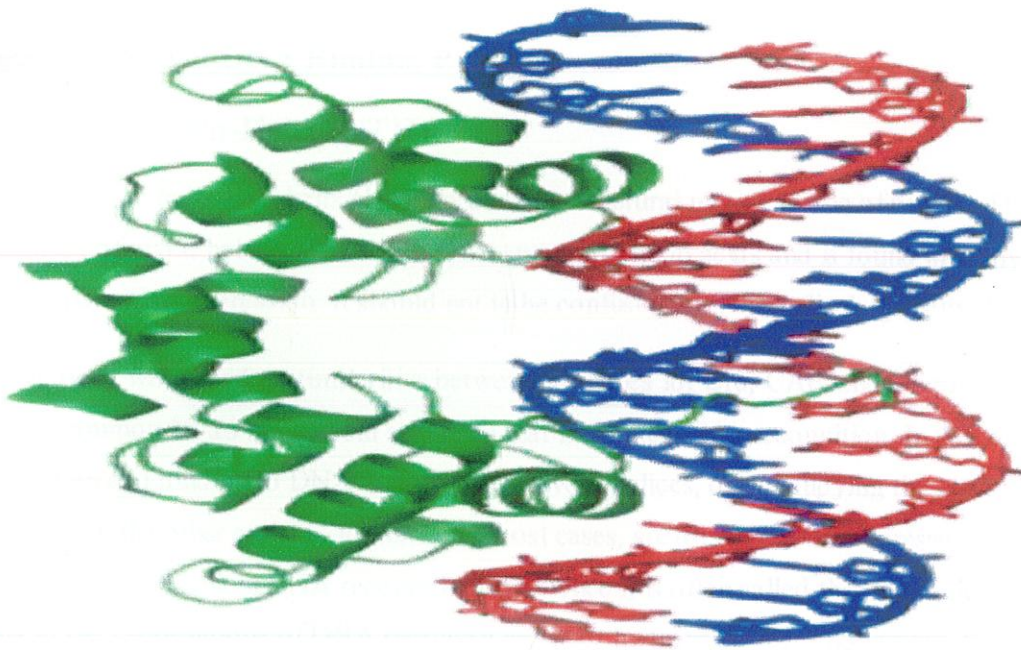
### **INTRODUCTION**

#### **What are DNA Binding Proteins?**

DNA Binding Proteins are those proteins that comprise many DNA Binding domains and thus have a specific or general affinity to DNA. A DNA Binding domain includes any protein motif that binds to double or single stranded DNA with affinity to a specific sequence or set thereof or a general affinity to DNA. DNA Binding domains are included in many proteins involved in regulation of gene expression (including transcription factors), proteins involved in the packaging of DNA within the nucleus (such as histones), nucleic acid dependent-polymerases involved in DNA replication and transcription or any of many accessory proteins which are involved in these processes. Proteins that bind DNA and are involved in replication or transcription do so in a sequence specific way. Transcription factors are dimers when active, i.e., they bind to DNA upon dimerisation and are inactive in the monomeric form. Dimerisation is a regulatory mechanism of controlling transcription factor activity. There are three common features most DNA Binding proteins have in common:

1. The major groove is the binding site of proteins through alpha-helices; the dimension of the major groove is 12Å wide and 8Å deep
2. The minor groove of B-DNA is 5Å wide, 8Å deep and is generally too narrow to fit entire alpha-helices but is recognized by beta-sheet structures of TATA-box binding proteins.
3. Sequence specific DNA Binding proteins generally do not disrupt the base pairs of the DNA, but do distort backbone conformation by bending the double helix.





**Fig 1: Showing DNA Binding protein Interaction**

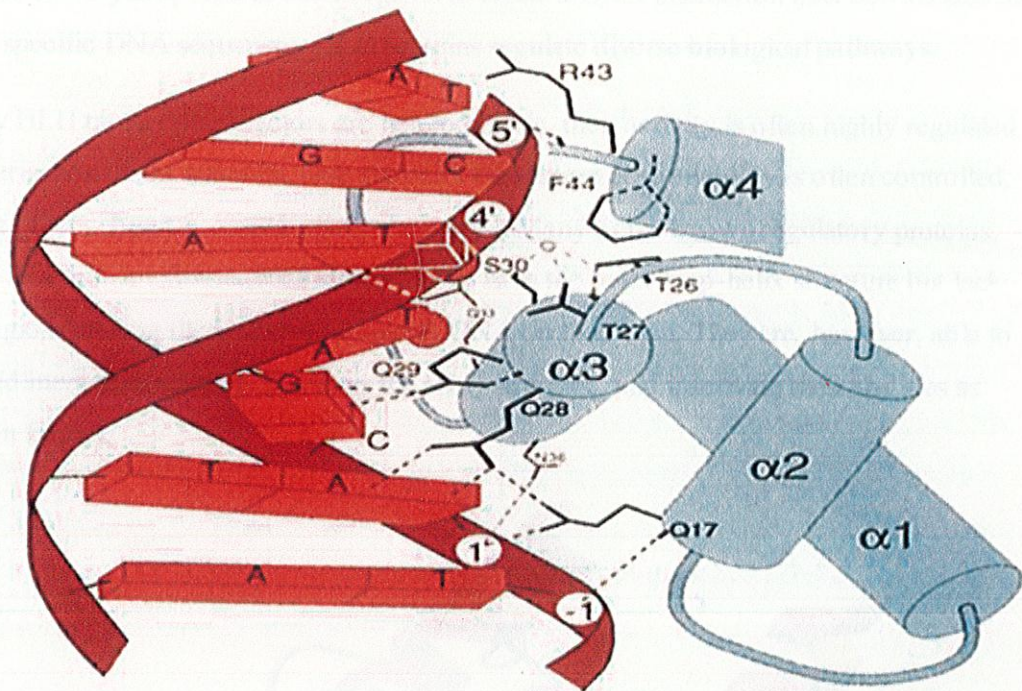
## **Classification of DNA Binding Proteins**

### **1. Helix-Turn-Helix (HTH):**

In proteins, the helix-turn-helix (HTH) is a major structural motif capable of binding DNA. It is composed of two  $\alpha$  helices joined by a short strand of amino acids and is found in many proteins that regulate gene expression. It should not be confused with the helix-loop-helix domain.

Its discovery was based on similarities between the genes for Cro, CAP, and  $\lambda$  repressor, which share a common 20-25 amino acid sequence that facilitates DNA recognition. In particular, recognition and binding to DNA is done by the two  $\alpha$  helices, one occupying the N-terminal end of the motif, the other at the C-terminus. In most cases, such as in the Cro repressor, the second helix contributes most to DNA recognition, and hence it is often called the "recognition helix". It binds to the major groove of DNA through a series of hydrogen bonds and various Van der Waals interactions with exposed bases. The other  $\alpha$  helix stabilizes the interaction between protein and DNA, but does not play a particularly strong role in its recognition. This motif is found in hundreds of DNA Binding proteins including tryptophan repressor, catabolite activator protein (CAP), octamer transcription factor-1 (Oct-1) and heat shock factor (HSF). Products of homeotic genes contain a homeo domain which is a special form of the helix-turn-helix motif. The helix-turn-helix motif is the common DNA recognition motif in prokaryotes. The motif resembles that of an EF-hand described in calmodulin. The F-helix is a recognition helix and the side chains give the specificity of binding. Sometimes more than one protein competes for the same sequence. The protein binds to a 22bp site, which consists of two groups of crucial bases, separated by one group of 10 bases (one turn of a DNA helix). Cyclic AMP (cAMP - Magenta in the structure) binds to the protein, which then binds to DNA, activating transcription.





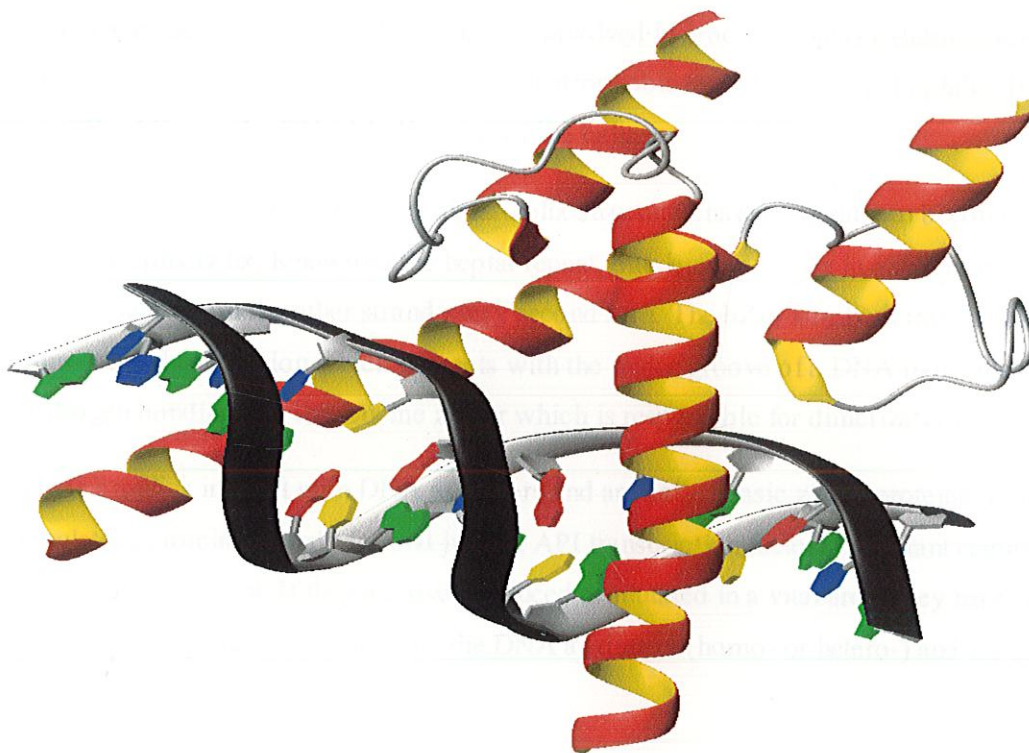
**Fig 2: Helix-Turn-Helix DNA Binding Protein**

## **2. Helix-Loop-Helix(HLH):**

Helix-Loop-Helix is characterized by two alpha helices connected by a loop. Transcription factors including this domain are dimeric, each with one helix containing basic amino acid residues that facilitate DNA binding. In general, one helix is smaller and due to the flexibility of the loop, allows dimerization by folding and packing against another helix. The larger helix typically contains the DNA-binding regions. HLH proteins typically bind to a consensus sequence called an E-box, CANNTG. The canonical E-box is CACGTG (palindromic), however some HLH transcription factors bind to non-palindromic sequences, which are often similar to the E-box. The loop in the HLH motif is flexible enough to permit folding back so that the two helices can pack against each other, that is, the two helices lie in planes that are parallel to each other. The helix-loop-helix (HLH) DNA-binding domain consists of a closed bundle of four helices in a left-handed twist with two crossover connections. The HLH domain directs

dimerisation, and is juxtaposed to basic regions to create a DNA interaction interface surface that recognizes specific DNA sequences. HLH proteins regulate diverse biological pathways.

Since many HLH transcription factors are heterodimeric, their activity is often highly regulated by the dimerization of the subunits. One subunit's expression or availability is often controlled, whereas the other subunit is constitutively expressed. Many of the known regulatory proteins, such as the *Drosophila* extramacrochaetae protein, have the helix-loop-helix structure but lack the basic region, making them unable to bind to DNA on their own. They are, however, able to form heterodimers with proteins that have the HLH structure, and inactivate their abilities as transcription factors.



**Fig 3: Helix Loop Helix DNA Binding Protein**

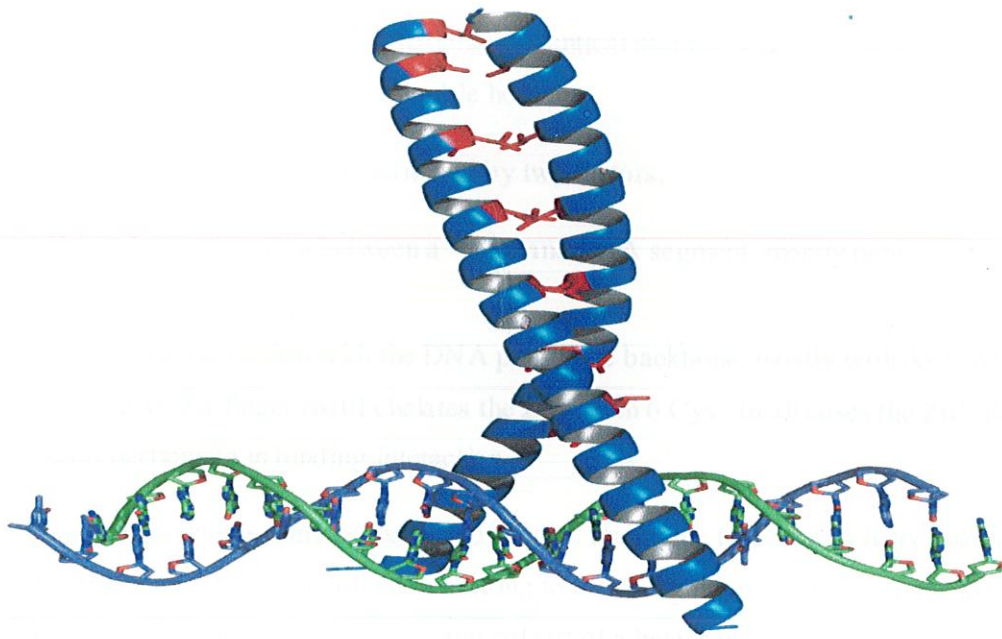
### 3. Leucine Zipper:

A leucine zipper is a super secondary structural motif found in proteins that creates adhesion forces in parallel alpha helices. Leucine Zipper was first identified by sequence alignment of certain transcription factors which identified a common pattern of leucines every seven amino acids. Each half of a leucine zipper consists of a short alpha helix with a leucine residue at every seventh position. In some transcription factors the dimer binding site with the DNA forms a so called leucine zipper. This motif consists of two amphipathic helices, one from each subunit, interacting with each other resulting in a left handed coiled-coil super secondary structure. The leucine zipper is interdigitation of regularly spaced leucine residues in one helix with leucines from the adjacent helix. Mostly the helices involved in leucine zippers exhibit a heptad sequence (abcdefg) with residues 'a' and 'd' being hydrophobic and all others hydrophilic. Leucine zipper motif itself is not the DNA binding part of the helices.

The standard 3.6 residues per turn alpha-helix structure changes slightly to become a 3.5 residue per turn alpha-helix. Known as the heptad repeat, one leucine comes in direct contact with another leucine on the other strand every second turn. The bZip family of transcription factors consist of a basic region which interacts with the major groove of a DNA molecule through hydrogen bonding and the leucine zipper which is responsible for dimerization.

These proteins interact with DNA as dimers and are called basic zipper proteins. Leucine zipper regulatory proteins include fos and jun(the API transcription factor), important regulators of normal development. If they are overproduced or mutated in a vital area, they may generate cancer. These proteins interact with the DNA as dimers (homo- or hetero-) and are also called basic zipper proteins (bZips).





**Fig 4: Leucine Zipper DNA Binding Protein**

#### **4. Zinc Finger:**

Zinc fingers are small protein domains that can coordinate one or more zinc ions to help stabilize their folds. They can be classified into several different structural families and typically function as interaction modules that bind DNA, RNA, proteins or small molecules.

This domain is common in eukaryotic DNA-binding proteins. It was first noticed in the eukaryotic transcription factor, TFIIIA

TFIIIA contains 9 repeated modules, each of which contains two Cysteine and two Histidine residues. These four residues chelate one  $Zn^{++}$  ion. Each finger is bound in the major groove of B-DNA.

Some eukaryotic transcription factors showed a unique motif called a Zn-finger where a  $Zn^{++}$  ion is coordinated by 2 Cys and 2 His residues.

Each Zn-finger interacts in a conformationally identical manner with successive triple base pair segments in the major groove of the double helix.

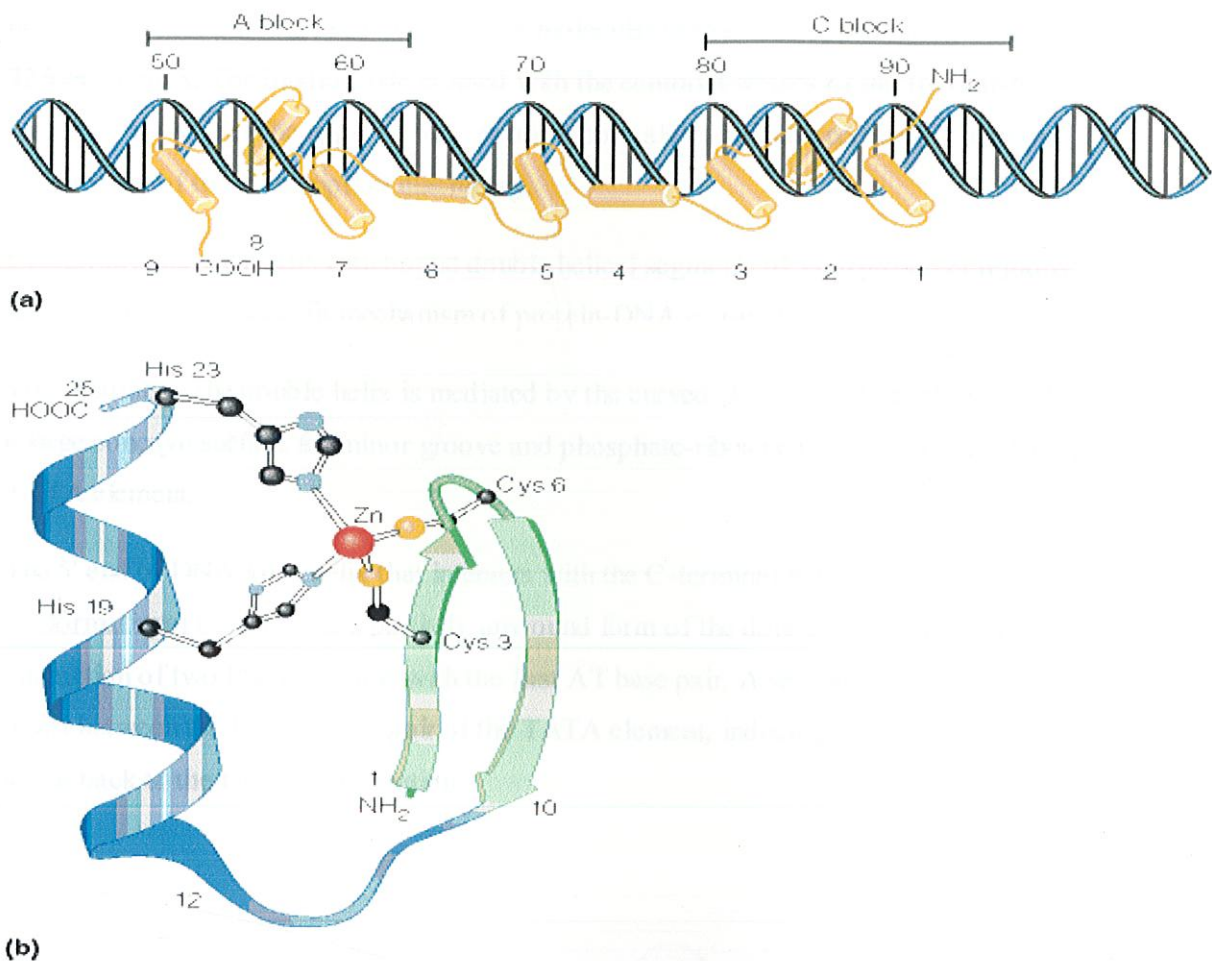
The protein-DNA interaction is determined by two factors:

- H-bonding interaction between a  $\alpha$ -helix and DNA segment, mostly between Arg residues and Guanine bases.
- H-bonding interaction with the DNA phosphate backbone, mostly with Arg and His. An alternative Zn-finger motif chelates the  $Zn^{++}$  with 6 Cys. In all cases the  $Zn^{++}$  does not itself participate in binding interaction.

Zn268 is a protein which contains three zinc fingers. Each zinc finger has a helix associated with it and it is the helix which is involved in binding to the recognition sequence on the DNA. The structure of the Zn Fingers is identical, and consist of a beta-hairpin followed by an alpha-helix. The sequence is guanine rich, although this is not necessary for all Zn fingers. Each finger recognizes three bases on the strand. The helices of the fingers fit into the major groove of the DNA, with the N-terminal ends interacting with the phosphate backbone. Recognition derives from the specific H-bonds from the Arginine residues in the protein to the guanine bases in the DNA.

There is no distortion of the DNA when the protein binds. The fingers simply slot into the major groove. The Zinc fingers are inherently flexible, and they adapt to the DNA structure instead.





**Fig 5: Zinc Finger DNA Binding Protein**

## 5. TATA Box:

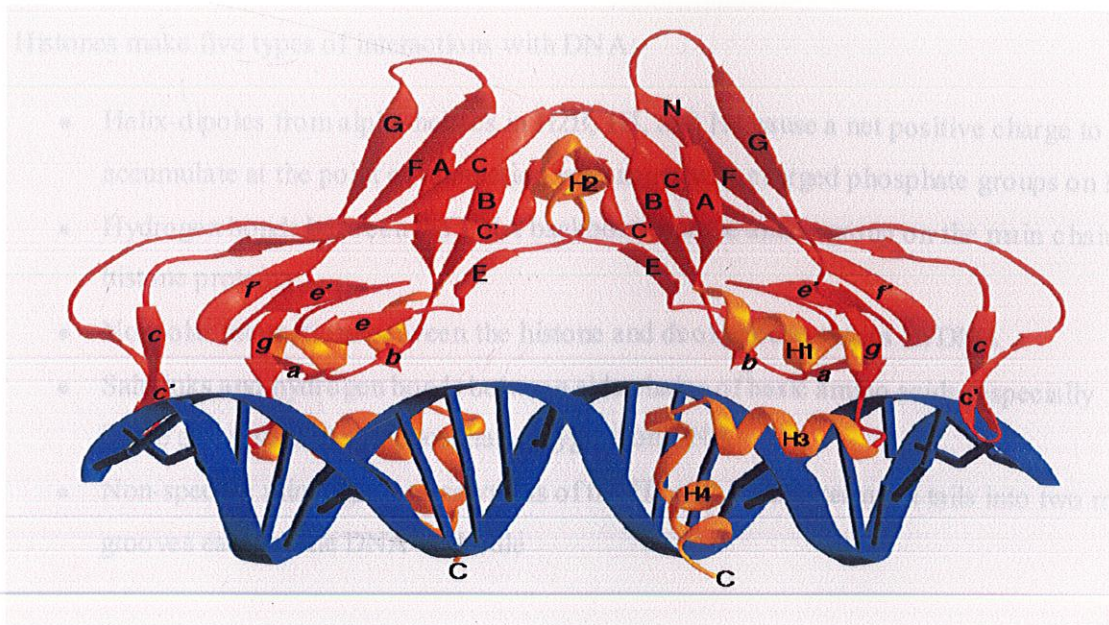
TATA box binding proteins (TBP) were first identified as a component of the class II initiation factor TFIID. These proteins participate in transcription by all three nuclear RNA polymerases acting as subunit in each of them. The structure of TBP was solved at 2.1Å resolution showing two alpha/beta structural domains of 89-90 amino acids. The C-terminal or core region binds with high affinity to the TATA consensus sequence recognizing minor groove determinants and

promoting DNA binding. TBP resembles a molecular saddle with approximate dimensions  $32\text{\AA} \times 45\text{\AA} \times 60\text{\AA}$ . The binding side is lined with the central 8 strands of the 10-stranded anti-parallel beta-sheet. The upper surface contains four alpha-helices and binds to various components of the transcription machinery.

Crystal structures of TBP with bound double helical segments of viral promoter regions demonstrate an induced-fit mechanism of protein-DNA recognition.

The bending of the double helix is mediated by the curved, 8 stranded beta-sheet motif providing a large concave surface for minor groove and phosphate-ribose contacts with the 8 base pair TATA element.

The 5' end of DNA form helix that interacts with the C-terminal portion of TBP producing a conformational transition to a partially unwound form of the double helix, induced by the interaction of two Phenylalanine with the first AT base pair. A second pair of phenylalanine insert between the last TA base pair of the TATA element, inducing a similar bend and the DNA forms back to the DNA conformation.



**Fig 6: TATA Box DNA Binding Protein**

## 6. Histone Binding:

Histones are the chief protein components of chromatin.

There are a total of six classes of histones (H1, H2A, H2B, H3, H4, and H5) organized into two super classes as follows:

- core histones – H2A, H2B, H3 and H4
- linker histones – H1 and H5

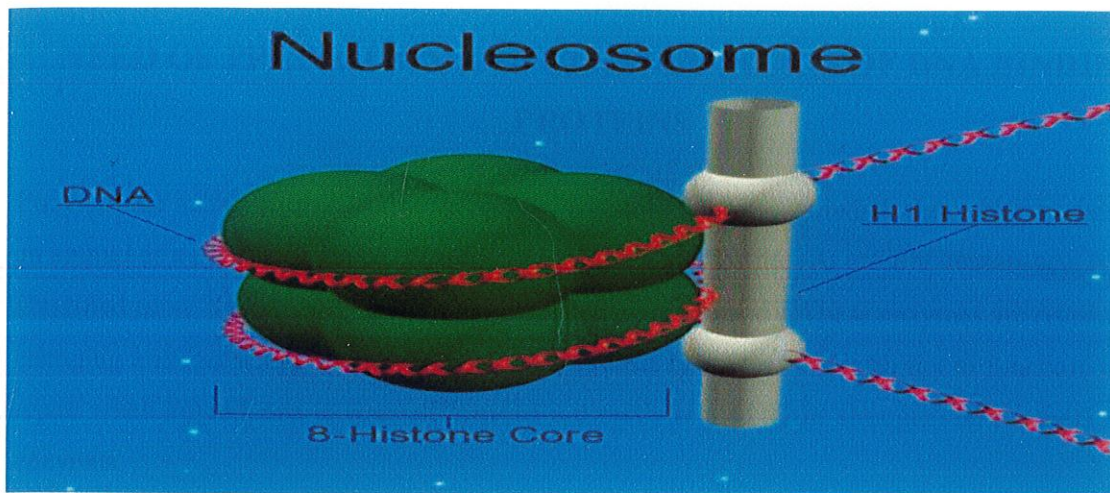
Two of each of the core histones assemble to form one octameric nucleosome core particle by wrapping 146 base pairs of DNA around the protein spool in 1.65 left-handed super-helical turn. The linker histone H1 binds the nucleosome and the entry and exit sites of the DNA, thus locking the DNA into place and allowing the formation of higher order structure.

The nucleosome core is formed of two H2A-H2B dimers and a H3-H4 tetramer, forming two nearly symmetrical halves by tertiary structure. The 4 'core' histones (H2A, H2B, H3 and H4) are relatively similar in structure and are highly conserved through evolution, all featuring a 'helix turn helix' motif.

Histones make five types of interactions with DNA:-

- Helix-dipoles from alpha-helices in H2B, H3, and H4 cause a net positive charge to accumulate at the point of interaction with negatively charged phosphate groups on DNA
- Hydrogen bonds between the DNA backbone and the amide group on the main chain of histone proteins
- Nonpolar interactions between the histone and deoxyribose sugars on DNA
- Salt links and hydrogen bonds between side chains of basic amino acids (especially lysine and arginine) and phosphate oxygens on DNA
- Non-specific minor groove insertions of the H3 and H2B N-terminal tails into two minor grooves each on the DNA molecule





**Fig 7: Histone Binding DNA Binding Protein**

## NEED OF PREDICTION AND CLASSIFICATION OF DNA BINDING PROTEINS

Enzymes are substances that occur naturally in all living things, including the human body. If it's an animal or a plant, it has enzymes. Enzymes are critical for life. At present, researchers have identified more than 3,000 different enzymes in the human body. These enzymes are constantly changing and renewing, sometimes at an unbelievable rate. Our body's ability to function, to repair when injured, and to ward off disease is directly related to the strength and numbers of our enzymes.

Using the protein engineering techniques, new enzymes have been created, ranging from food enzymes to the enzymes used for curing diseases. The large international genome sequence projects are gaining a great amount of public attention and huge sequence data bases are created. It becomes more and more obvious that we are very limited in our ability to access functional data for the gene products- the proteins, in particular for enzymes. It seems quite improbable to experimentally determine function and structure of each candidate protein. So a new method is needed to solve this computation catastrophe. Primary sequence of these proteins are readily available, therefore a method using the sequence derived features will prove a much valuable and a cost effective process of determining and classifying these proteins into six major classes.





## **MACHINE LEARNING CLASSIFICATION**

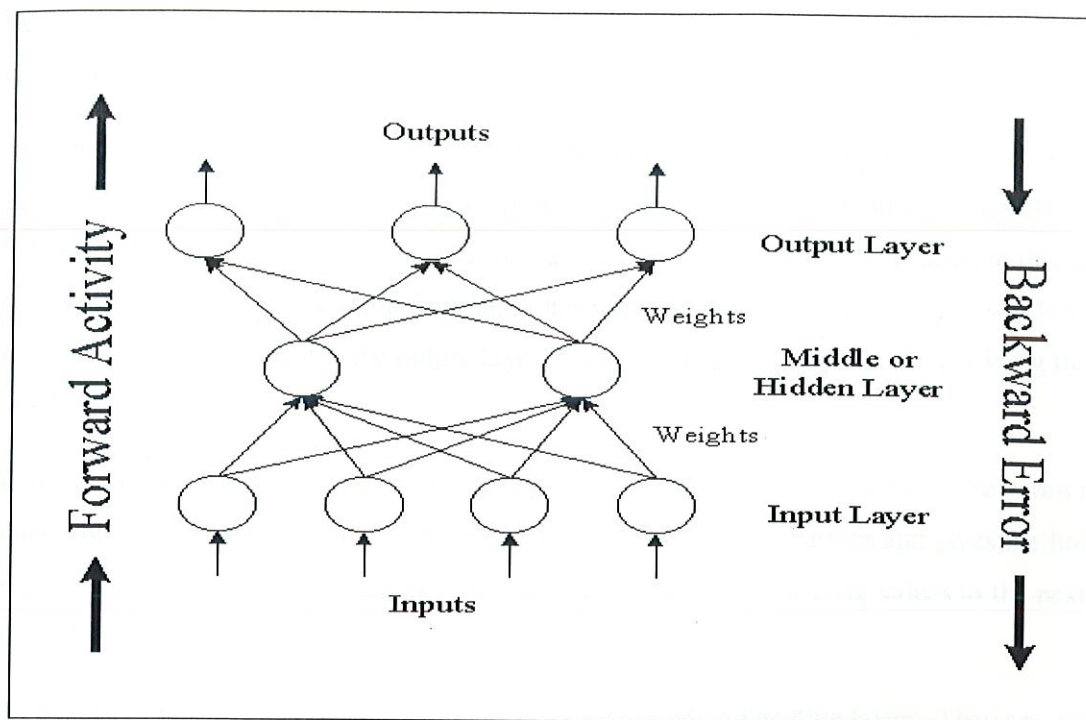
As a broad subfield of artificial intelligence, machine learning is concerned with the design and development of algorithms and techniques that allow computers to 'learn'. At a general level, there are two types of learning: inductive and deductive. Inductive machine learning methods extract rules and patterns out of massive data sets. The major focus of Machine learning research is to extract information from the data automatically by computational and statistical methods. Hence, machine learning is closely related to data mining and statistics but also theoretical computer science. Machine learning has a wide spectrum of applications including natural language processing, syntactic pattern recognition, search engines, medical diagnosis, bioinformatics and chemoinformatics, detecting credit card fraud, stock market analysis, classifying DNA sequences, speech and handwriting recognition, object recognition in computer vision, game playing and robot locomotion.

## **NEURAL NETWORKS**

Neural Network or more appropriately Artificial Neural Network is basically a mathematical model of what goes in our mind (brain). The brain of all the advanced living creatures consist of neurons, a basic cell, which when interconnected produces what we call Neural Network. The sole purpose of a Neuron is to receive electrical signals, accumulate them and see further if they are strong enough to pass forward. The basic functionality lies not in neurons but the complex pattern in which they are interconnected. NNs are just like a game of chess, easy to learn but hard to master. In the same way, a single neuron is useless. Well, practically useless. It is the complex connection between them and values attached with them which makes brains capable of thinking and having a sense of consciousness (much debated).

## **BASIC WORKING OF A NEURON**

A neuron is basically a cell which accumulates electrical signals with different strengths. What it does more is that it compares the accumulated signal with one predefined value unique to every neuron. This value is called bias. Function of a neuron could be explained in the following diagram.



**Fig 8: Typical Artificial Neural Network Setup**

The circles in the image represent neurons. This network or more appropriately this network topology is called feed-forward multi layered neural network. It is the most basic and most widely used network. The network is called multi layered because it consists of more than two layers. The neurons are arranged in a number of layers, generally three. They are input, hidden/middle and output layers. This network is feed –forward, means the values are propagated in one direction only. There are many other topologies in which values can be looped or move in both forward and backward direction. But, this network allows the movement of values only from input layer to output layer. The functions of various layers are explained below:

**Input Layer:** As it says, this layer takes the inputs and forwards it to hidden layer. We can imagine input layer as a group of neurons whose sole task is to pass the numeric inputs to the next level. The larger the number greater is its strength. e.g. 0.51 is stronger than 0.39 but 0.93412 is stronger still. But, the interpretation of this strength depends upon the implementation and the type of problem assigned to NN to solve. For an OCR we connect every pixel with its

input neuron and darker the pixel, higher the signal/input strength. Input layer never processes data, it just hands over it.

**Middle Layer:** This layer is the real thing behind the network. Without this layer, network would not be capable of solving complex problems. There can be any number of middle or hidden layers. But, for most of the tasks, one is sufficient. The number of neurons in this layer is critical. This layer takes the input from input layer, does some calculations and forwards to the next layer, in most cases it is the output layer. There is no specific formula for deciding the number of hidden nodes.

**Output Layer:** This layer consists of neurons which predict the output value of the given input data. This layer takes the value from the previous layer, does calculations and gives the final result. Basically, this layer is just like hidden layer but instead of passing values to the next layer, the values are treated as output.

**Dendrites:** These are straight lines joining two neurons of consecutive layers. They are just a passage (or method) through which values are passed from one layer to the next. There is a value attached with dendrite called weight. The weight associated with dendrites basically determines the importance of incoming value. A weight with larger value determines that the value from that particular neuron is of higher significance. To achieve this we do is multiply the incoming value with weight. So no matter how high the value is, if the weight is low the multiplication yields the final low value.

**Training:** Training is the most important part of a neural network and the one consisting of the most mathematics. It uses Back Propagation method for training the NN. Training a neural network model essentially means selecting one model from the set of allowed models (or, in a Bayesian framework, determining a distribution over the set of allowed models) that minimizes the cost criterion. There are numerous algorithms available for training neural network models; most of them can be viewed as a straightforward application of optimization theory and statistical estimation. Most of the algorithms used in training artificial neural networks are employing some form of gradient descent. This is done by simply taking the derivative of the cost function with respect to the network parameters and then changing those parameters in a gradient-related direction. Evolutionary methods, simulated annealing, and expectation-

maximization and non-parametric methods are among other commonly used methods for training neural networks. The training procedure must be repeated for larger number of samples so that the NN can produce accurate results for untrained input samples.

## LEARNING PARADIGMS

There are three major learning paradigms, each corresponding to a particular abstract learning task. These are supervised learning, unsupervised learning and reinforcement learning. Usually any given type of network architecture can be employed in any of those tasks.

### Supervised learning

This form of learning assumes the availability of a labeled set of training data made up of  $N$  input—output examples:

$$T = \{(x_i, d_i)\}_{i=1}^N$$

where  $x_i$  = input vector of  $i$ th example

$d_i$  = desired (target) response of  $i$ th example, assumed to be scalar for convenience of presentation

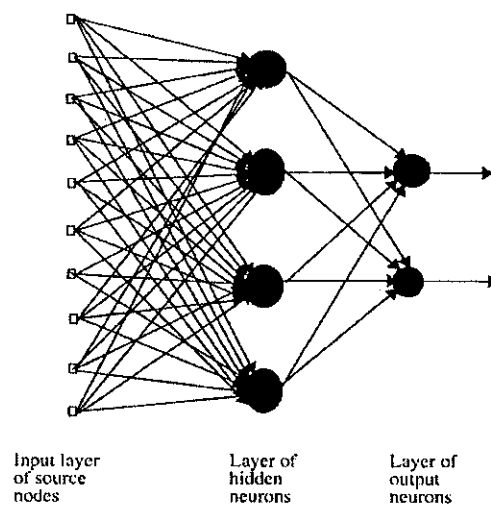
$N$  = sample size

Given the training sample  $T$ , the requirement is to compute the free parameters of the neural network so that the actual output  $y_i$  of the neural network due to  $x_i$  is close enough to  $d_i$  for all  $i$  in a statistical sense. For example, we may use the mean-square error as the index of performance to be minimized.

$$E(n) = \frac{1}{N} \sum_{i=1}^N (d_i - y_i)^2$$

## Multilayer Perceptrons and Back-Propagation Learning

The back-propagation algorithm has emerged as the workhorse for the design of a special class of layered feed-forward networks known as multilayer perceptrons (MLP). As shown in Fig.9, a multilayer perceptron has an input layer of source nodes and an output layer of neurons (i.e., computation nodes); these two layers connect the network to the outside world. In addition to these two layers, the multilayer perceptron usually has one or more layers of hidden neurons, which are so called because these neurons are not directly accessible. The hidden neurons extract important features contained in the input data.



**Fig 9: Fully connected feed-forward with one hidden layer and one output layer**

The training of an MLP is usually accomplished by using backpropagation (*BP*) *algorithm* that involves two phases:



- **Forward Phase.**

During this phase the free parameters of the network are fixed, and the input signal is propagated through the network of Fig.9 layer by layer. The forward phase finishes with the computation of an error signal

$$e_i = d_i - y_i$$

where  $d_i$  is the desired response and  $y_i$  is the actual output produced by the network in response to the input  $x_i$ .

- **Backward Phase.**

During this second phase, the error signal  $e_i$  is propagated through the network of Fig.9 in the backward direction, hence the name of the algorithm. It is during this phase that adjustments are applied to the free parameters of the network so as to minimize the error  $e_i$  in a statistical sense. Back-propagation learning may be implemented in one of two basic ways, as summarized here:

- **Sequential mode** (also referred to as the on-line mode or stochastic mode):

In this mode of BP learning, adjustments are made to the free parameters of the network on an example-by-example basis. The sequential mode is best suited for pattern classification.

- **Batch mode**

In this second mode of BP learning, adjustments are made to the free parameters of the network on an epoch-by-epoch basis, where each epoch consists of the entire set of training examples. The batch mode is best suited for nonlinear regression.

The back-propagation learning algorithm is simple to implement and computationally efficient in that its complexity is linear in the synaptic weights of the network. However, a major limitation of the algorithm is that it does not always converge and can be excruciatingly slow, particularly when we have to deal with a difficult learning task that requires the use of a large network.

We may try to make back-propagation learning perform better by invoking the following list of heuristics:

- Use neurons with anti-symmetric activation functions (e.g., hyperbolic tangent function) in preference to non-symmetric activation functions (e.g., logistic function).
- Shuffle the training examples after the presentation of each epoch; an epoch involves the presentation of the entire set of training examples to the network.
- Follow an easy-to-learn example with a difficult one.
- Preprocess the input data so as to remove the mean and de-correlate the data.
- Arrange for the neurons in the different layers to learn at essentially the same rate. This may be attained by assigning a learning rate parameter to neurons in the last layers that is smaller than those at the front end.
- Incorporate prior information into the network design whenever it is available.

One other heuristic that deserves to be mentioned relates to the size of the training set,  $N$ , for a pattern classification task. Given a multilayer perceptron with a total number of synaptic weights including bias levels, denoted by  $W$ , a rule of thumb for selecting  $N$  is

$$N = O\left(\frac{W}{\epsilon}\right)$$

where,  $O$  denotes “the order of,” and  $\epsilon$  denotes the fraction of classification errors permitted on test data. For example, with an error of 10% the number of training examples needed should be about 10 times the number of synaptic weights in the network.

## Radial-Basis Function Networks

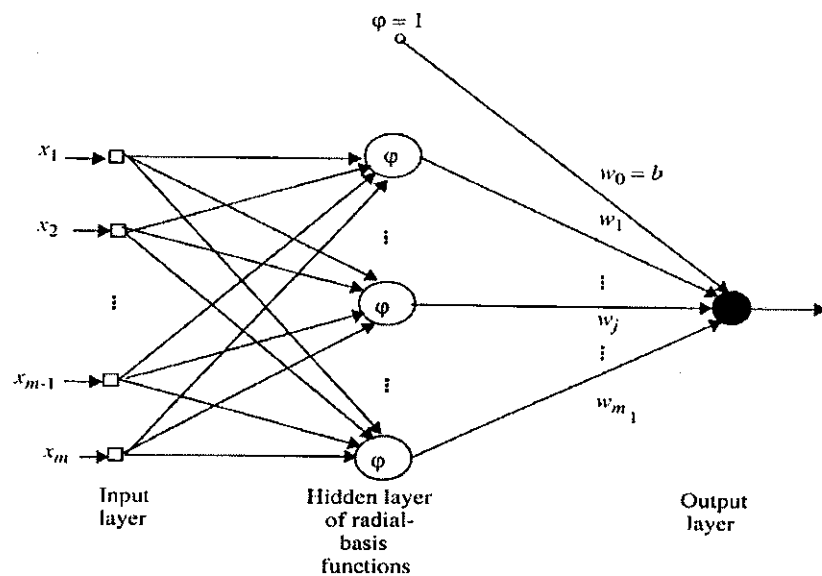
Another popular layered feed-forward network is the radial-basis function (RBF) network which has important universal approximation properties (Park and Sandberg 1993), and whose structure is shown in Fig.10. RBF networks use memory-based learning for their design. Specifically, learning is viewed as a curve-fitting problem in high-dimensional space

1. Learning is equivalent to finding a surface in a multidimensional space that provides a best fit to the training data.

2. Generalization (i.e., response of the network to input data not seen before) is equivalent to the use of this multidimensional surface to interpolate the test data.

RBF networks differ from multilayer perceptrons in some fundamental respects:

- RBF networks are local approximators, whereas multilayer perceptrons are global approximators.
- RBF networks have a single hidden layer, whereas multilayer perceptrons can have any number of hidden layers.
- The output layer of a RBF network is always linear, whereas in a multilayer perceptron it can be linear or nonlinear.



**Fig 10: Radial-basis function network.**

- The activation function of the hidden layer in an RBF network computes the Euclidean distance between the input signal vector and parameter vector of the network, whereas the activation function of a multilayer perceptron computes the inner product between the input signal vector and the pertinent synaptic weight vector.

The use of a linear output layer in an RBF network may be justified in light of *Cover's theorem* on the separability of patterns.

According to this theorem, provided that the transformation from the input space to the feature (hidden) space is nonlinear and the dimensionality of the feature space is high compared to that of the input (data) space, then there is a high likelihood that non-separable pattern classification task in the input space is transformed into a linearly separable one in the feature space.

## **Learning algorithms**

Training a neural network model essentially means selecting one model from the set of allowed models (or, in a Bayesian framework, determining a distribution over the set of allowed models) that minimizes the cost criterion. There are numerous algorithms available for training neural network models; most of them can be viewed as a straightforward application of optimization theory and statistical estimation.

Most of the algorithms used in training artificial neural networks employ some form of gradient descent. This is done by simply taking the derivative of the cost function with respect to the network parameters and then changing those parameters in a gradient-related direction.

Evolutionary methods, simulated annealing, expectation-maximization and non-parametric methods are some commonly used methods for training neural networks.

## **Applications:**

The utility of artificial neural network models lies in the fact that they can be used to infer a function from observations. This is particularly useful in applications where the complexity of the data or task makes the design of such a function by hand impractical.

## **Real life applications:**

The tasks to which artificial neural networks are applied tend to fall within the following broad categories:

- Function approximation, or regression analysis, including time series prediction and modeling.
- Classification, including pattern and sequence recognition, novelty detection and sequential decision making.
- Data processing, including filtering, clustering, blind source separation and compression.
- Application areas include system identification and control (vehicle control, process control), game-playing and decision making (backgammon, chess, racing), pattern recognition (radar systems, face identification, object recognition and more), sequence recognition (gesture, speech, handwritten text recognition), medical diagnosis, financial applications, data mining (or knowledge discovery in databases, "KDD"), visualization and e-mail spam filtering.



## OBJECTIVE

Rational classification of proteins encoded in sequenced genome is critical for making the genome sequences maximally useful for functional and evolutionary studies. The family of DNA-binding proteins is one of the most populated and studied amongst the various genomes of bacteria, archaea and eukaryotes and the method presented here is an approach to their classification. Sequence similarity metrics are a useful approach to provide functional annotation, but its use is sometimes limited, prompting the development and use of machine learning methods (MLMs). MLMs also have a certain degree of flexibility regarding data inputs, allowing them to expand progressively to meet the requirements of rapidly accumulating mountain of data generated from genomics research.

Hence, in this study an attempt has been taken to develop an automated tool using machine learning technique for annotation of protein sequences with following objectives:

1. To extract sequence derived features and selection of important features from protein sequences to be used for prediction and classification of DNA binding proteins.
2. To develop and optimize the 1st layer ANN for classifying the user input protein sequences into DNA or Non-DNA binding based on sequence derived features.
3. To develop and optimize the 2nd layer ANN for classifying the predicted DNA binding proteins into 6 major classes based on sequence derived features.

## **CHAPTER 2**

### **A TOOL FOR PREDICTION AND CLASSIFICATION OF DNA BINDING PROTEINS INTO SIX MAJOR CLASSES USING ANN FROM SEQUENCE DERIVED FEATURES**

## ABSTRACT

The problem of predicting the DNA binding and Non-DNA binding from protein sequence information is still an open problem in bioinformatics. It is further becoming more important as the number of sequenced information grows exponentially over time. The large amount of proteomic data is available for a variety of organisms which allow researchers to efficiently identify novel proteins in distantly related organisms and annotating them. A faster means of annotation would be to match them with the already annotated sequences. Therefore arises the need for development of a prediction tool which can take raw protein sequence as input and can predict the output as whether the input protein sequence is DNA binding or not and if it is then which class it belongs to. This will further give us the predictive insight on molecular function and pathways in which a novel protein may be involved prompting the development and use of machine learning methods.

The tool would be consisting of 2 levels of Classification. First layer classification includes whether protein is DNA Binding or Non-DNA Binding. If the protein is DNA Binding the program enters into the second level of hierarchical classification system which then groups them into one of the following categories – Helix Loop Helix, Helix Turn Helix, Leucine Zipper, Histone Binding, T-Box and Zinc Finger.

## INTRODUCTION

The prediction of protein structure from amino acid sequence has become the Holy Grail of computational molecular biology. Since Anfinsen first noted that the information necessary for protein folding resides completely within the primary structure, molecular biologists have been fascinated with the possibility of obtaining a complete three-dimensional picture of a protein by simply applying the proper algorithm to a known amino acid sequence. The development of rapid methods of DNA sequencing coupled with the straightforward translation of the genetic code into protein sequences has amplified the urgent need for automated methods of interpreting these one-dimensional, linear sequences in terms of three-dimensional structure and function. Advanced and specialized databases are needed to facilitate the retrieval of relevant information from the deluge of sequence data and to provide insight into the protein structure and function. Further, it is clear that rational classification of proteins encoded in sequenced genomes is critical for making the genome sequences maximally useful for functional and evolutionary studies.

The family of DNA binding proteins is one of the most populated and studied amongst the various genomes of bacteria, archaea and eukaryotes. Most of these proteins, such as the eukaryotic and prokaryotic transcription factors, contain independently folded units (domains) in order to accomplish their recognition with the contours of DNA. It is now clear that the majority of these DNA-binding scaffolds which are in general relatively small, less than 100 amino acid residues, belong to a large number of structural families with characteristic sequences and three-dimensional designs or conformations. Computational biology applying fast and sensitive algorithms strives to extract the maximum possible information from these sequences by classifying them according to their homologous relationships, predicting their likely biochemical activities and/or cellular functions, three-dimensional structures and evolutionary origin. There have been studies to detect, design and predict them using a probabilistic recognition code. There have also been works towards analyzing protein-DNA recognition mechanism and binding site discovery. DNA binding proteins represent a broad category of proteins, known to be highly diverse in sequence and structure. Structurally, they have been divided into 54 protein-structural families. With such a high degree of variance, using conventional annotation methods rooted in



database searching for sequence similarity, profile or motif similarity and phylogenetic profiles may not lead to reliable annotations.

Previously, there have been a few bioinformatics methods developed towards automated identification and prediction of DNA binding proteins. Cai and Lin used pseudo-amino acid composition to identify proteins that bind to RNA, rRNA and DNA. Ahmad integrated structural information with a neural network approach for the prediction of DNA binding proteins. Stawiski and Jones characterized electrostatic features of proteins for an automated approach to DNA binding protein and DNA binding site prediction. Ahmad and Sarai showed that overall charge and electric moment can be used to identify DNA binding proteins.

Strategically, we have used a neural network, two-layer, fully automated computational method capable of recognizing DNA binding proteins first, and then classifying them into six different classes based on their sequences derived features.

## MATERIALS AND METHODS

These neural networks cluster takes the sequences one by one for the prediction. In the study along with usage of machine learning approach like ANN automated as wells as customized, we have also used three types of parameters like physicochemical properties, amino acid composition and pseudo amino acid composition. Using this combination of we have seven neural network models- ANN<sub>pepstat</sub>, ANN<sub>AA comp</sub> and ANN<sub>PseAA</sub> as the individual ANNs and rest all are their combinations.

For building up the neural network models we have used *STATISTICA* v.9.1 by Statsoft. SANN is *STATISTICA* Enterprise-Wide Data Mining System (Data Miner) that offers a comprehensive selection of Neural Network solutions. By joining these models we have developed 7 neural network clusters.

**Following are the steps which are performed for the development of the tool**

- Data Collection
- Data Reduction
- Descriptor Calculation
- Neural Network Model Building
- Neural Cluster Formation
- Server Development

All the above steps are described in detail:

## 1. Data Collection and Data Reduction:

- **Dataset for prediction of DNA binding/non-DNA binding**

A dataset of 766 DNA binding protein sequences extracted from UniProt (<http://www.uniprot.org/>) was used as a model class after removing the redundant sequences. A non-redundant treatment was applied to eliminate the sequences which share a high degree of similarity (>90%) with others in order to avoid overtraining. The treatment was carried out using the program BLASTCLUST (<http://www.ncbi.nlm.nih.gov/BLAST/>), which used the BLAST algorithm to systematically cluster protein sequences on the basis of pair-wise matches. The default values were used for all BLAST parameters: matrix BLOSUM62, gap opening cost of 11, gap extension cost of 1, E-value threshold of  $1e^{-6}$ . These sequences were used as positive examples for prediction as DNA binding proteins. The sequences data on negative examples were obtained from the IMTECH (<http://imtech.res.in/>). DNA binding proteins were removed from the original dataset. A non-redundant treatment was applied (same as for positive datasets) such that no sequence had similarity higher than 25% to any others. Thus, 983 non-DNA binding sequences were optimized as negative examples.

- **Dataset for classification of DNA binding proteins into six major classes**

The above mentioned 766 protein sequences of DNA binding proteins were then grouped into six major classes: Helix-Loop-Helix(219), Helix-Turn-Helix(109), Leucine Zipper(102), Histone Binding(170), T-Box(74) and Zinc Finger(92). They were used for construction of neural networks, training and validating the model for classification of predicted DNA binding proteins into six classes.

## 2. Descriptor Calculation:

Molecular descriptors play a fundamental role in chemistry, pharmaceutical sciences, environmental protection policy, and health researches, as well as in quality control, being the way molecules, thought of as real bodies, are transformed into numbers, allowing some mathematical treatment of the chemical information contained in the molecule. This was defined by Todeschini and Consonni as:



*"The molecular descriptor is the final result of a logic and mathematical procedure which transforms chemical information encoded within a symbolic representation of a molecule into a useful number or the result of some standardized experiment."*

The descriptors used are:

- **Amino Acid composition**: This descriptor consist of 20 factors each representing composition of 20 standard amino acids in the protein sequences that include A, C, D, E, F, G, H, I, K, L, M, P, Q, R, S, T, V, W, X, and Y. The formula to calculate this composition is:

$$AA\ comp(i) = \frac{Freq.\ of\ AA(i)}{\sum Freq.\ of\ AA\ in\ seq.}$$

- **Physicochemical Properties**: This descriptor consists of 12 properties calculated using EMBOSS (EBI) package. The parameters include Molecular weight, Charge, Isoelectric point., Mole percentages of Tiny, Small, Aliphatic, Aromatic, Non-polar, Polar, Charged, Acidic, Basic amino acids. The different categories include different sets of amino acids like Tiny (A+C+G+S+T), Small (A+B+C+D+G+N+P+S+T+V), Aliphatic (I+L+V), Aromatic (F+H+W+Y), Non-polar (A+C+F+G+I+L+M+P+V+W+Y), Polar (D+E+H+K+N+Q+R+S+T+Z), Charged (B+D+E+H+K+R+Z), Basic (H+K+R) and Acidic (B+D+E+Z).
- **Pseudo AA composition**: This descriptor is a collection of 37 factors, 20 of which are simple amino acid compositions and rest 17 are correlation factors calculated among amino acids of the given sequences. It was introduced by Kuo-Chen Chou in 2001 to represent protein samples for statistical prediction.

The simplest discrete model is using the AA composition to represent protein samples, as formulated as follows. Given a protein sequence P with  $L$  amino acid residues, i.e.,

$$P = R_1 R_2 R_3 R_4 R_5 R_6 R_7 \cdots R_L \quad (1)$$



where  $R_1$  represents the 1st residue of the protein P,  $R_2$  the 2nd residue, and so forth, according to the AA composition model, the protein P of Eq.1 can be expressed by

$$\mathbf{P} = [f_1 \ f_2 \ \cdots \ f_{20}]^T \quad (2)$$

where  $f_u$  ( $u = 1, 2, \dots, 20$ ) are the normalized occurrence frequencies of the 20 native amino acids in P, and T the transposing operator. The additional factors are a series of rank-different correlation factors along a protein chain, but they can also be any combinations of other factors so long as they can reflect some sorts of sequence-order effects one way or the other. The algorithm for this is as follows:

According to the Pseudo AA composition model, the protein P of Eq.1 can be formulated as

$$\mathbf{P} = [p_1, p_2, \dots, p_{20}, p_{20+1}, \dots, p_{20+\lambda}]^T, \quad (\lambda < L) \quad (3)$$

where  $20 + \lambda$  the components are given by

$$p_u = \begin{cases} \frac{f_u}{\sum_{i=1}^{20} f_i + w \sum_{k=1}^{\lambda} \tau_k}, & (1 \leq u \leq 20) \\ \frac{w \tau_{u-20}}{\sum_{i=1}^{20} f_i + w \sum_{k=1}^{\lambda} \tau_k}, & (20 + 1 \leq u \leq 20 + \lambda) \end{cases} \quad (4)$$

Where  $w$  is the weight factor, and  $\tau_k$  the  $k$ -th tier correlation factor that reflects the sequence order correlation between all the  $k$ -th most contiguous residues as formulated by

$$\tau_k = \frac{1}{L-k} \sum_{i=1}^{L-k} J_{i,i+k}, \quad (k < L) \quad (5)$$

with

$$J_{i,i+k} = \frac{1}{\Gamma} \sum_{g=1}^{\Gamma} [\Phi_g(R_{i+k}) - \Phi_g(R_i)]^2 \quad (6)$$

Where  $\Phi_g(R_i)$  is the  $g$ -th function of the amino acid  $R_i$ , and  $\Gamma$  the total number of the functions considered. For example, in the original paper by Chou,  $\Phi_1(R_i)$ ,  $\Psi_2(R_i)$  and  $\Psi_3(R_i)$  are respectively the hydrophobicity value, hydrophilicity value, and side chain mass of amino acid  $R_i$ ; while  $\Phi_1(R_{i+1})$ ,  $\Phi_2(R_{i+1})$  and  $\Phi_3(R_{i+1})$  the corresponding values for the amino acid  $R_{i+1}$ . Therefore, the total number of functions considered there is  $\Gamma = 3$ . It can be seen from Eq.3 that the first 20 components, i.e.  $p_1, p_2, \dots, p_{20}$  are associated with the conventional AA composition of protein, while the remaining components  $p_{20+1}, \dots, p_{20+\lambda}$  are the correlation factors that reflect the 1st tier, 2nd tier, ..., and the  $\lambda$ -th tier sequence order correlation patterns. It is through these additional  $\lambda$  factors that some important sequence-order effects are incorporated.

### 3. Neural Network Model Building:

The implementation of ANN was realized using the software package STATISTICA v.9.1 by Statsoft SANN. We have used two feed-forward back-propagation neural networks with a single hidden layer. First layer of neural network is used for prediction of DNA binding/non-DNA binding proteins from the protein sequence, whereas, the second layer is used for classifying the predicted DNA binding protein into six major classes.

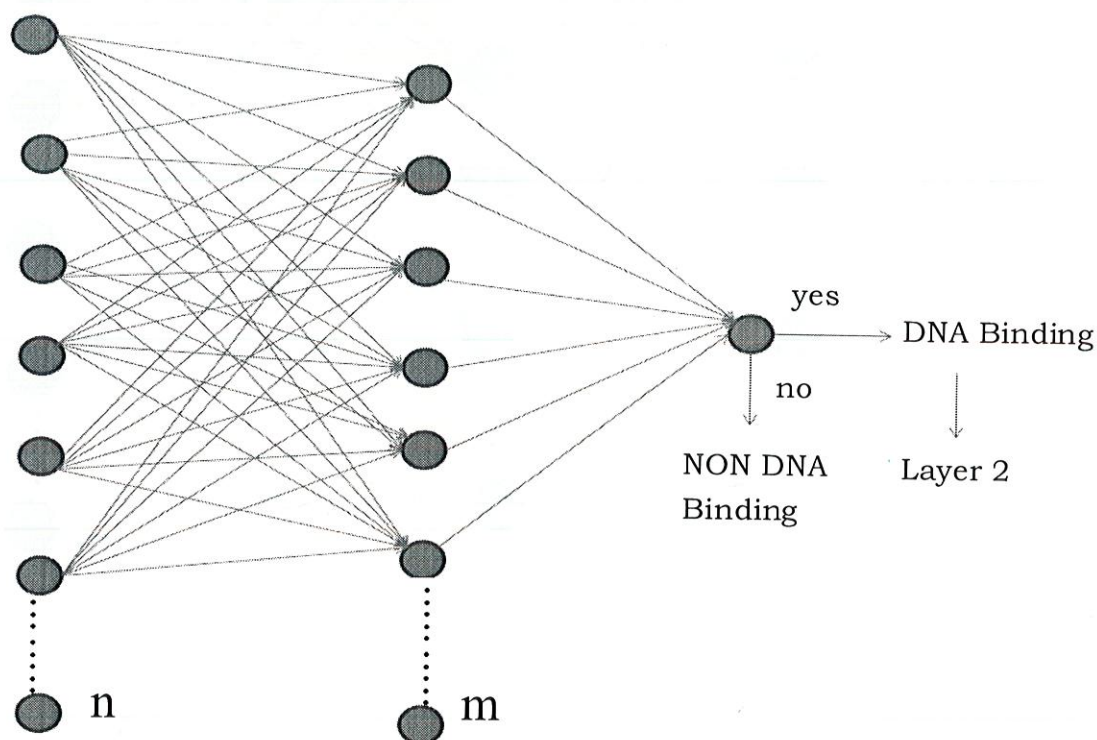


Fig 11: Architecture of Layer 1



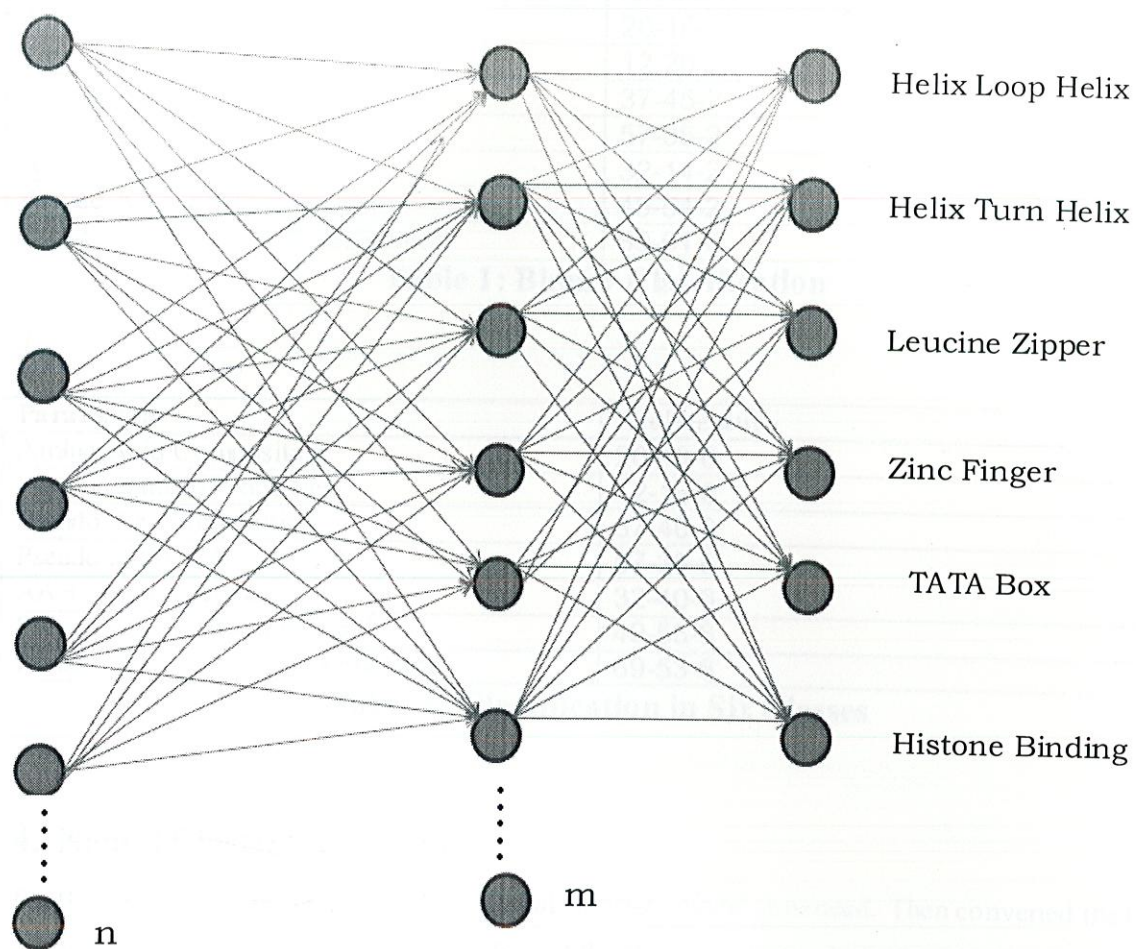


Fig 12: Architecture for Layer 2



Parameters	Architecture
Amino Acid Composition	20-16-2
Physiochemical Properties	12-20-2
Pseudo AA Composition	37-45-2
Pseudo AA + AA	57-35-2
AA + Physiochemical	32-14-2
Pseudo AA + Physiochemical	49-54-2
Pseudo AA + AA + Physiochemical	69-61-2

**Table 1: Binary Classification**

Parameters	Architecture
Amino Acid Composition	20-18-6
Physiochemical Properties	12-18-6
Pseudo AA Composition	37-40-6
Pseudo AA + AA	57-39-6
AA + Physiochemical	32-40-6
Pseudo AA + Physiochemical	49-55-6
Pseudo AA + AA + Physiochemical	69-53-6

**Table 2: Classification in Six Classes**

#### 4. Neural Cluster Formation:

Firstly, we saved the C codes for each Neural Network model generated. Then converted the C codes to C library references as Header files. After that we generated a main parser code, which can take in the descriptors in the form of a file and can send them to the particular network models in their corresponding header files and retrieving the output of the model based on output, taking the decision to go which way or to which particular model to feed the descriptor and retrieving the output.

#### 5. Server Development:

We have generated a Perl parser which can take the constraints of the descriptor from the user and take in the sequences from the file, then it can pipe in the sequences to the other codes for calculation of the desired descriptor and thereby the prediction cluster is fired, which has been developed in the previous case and this retrieve the output and presents it to the user.

## 6. Validation:

The Validation is the way to confirm the validity of data, information, or processes of a model.

- **Self consistency**

Consistent with one's self or with itself; not deviation from the ordinary standard by which the conduct is guided; logically consistent throughout; having each part consistent with the rest.

In our case we have taken the entire dataset that we have used for creation of these models, and we have used this data set for validating our prediction tool. We took 766 protein sequences in DNA Binding category and 983 protein sequences in Non-DNA Binding category.

**Table 3(a) Percentage of correct prediction in DNA Binding and Non- DNA Binding category for Self Consistency**

Parameters	DNA Binding	Non-DNA Binding
Amino Acid Composition	98.96	98.58
Physiochemical Properties	98.43	96.85
Pseudo AA Composition	93.86	98.98
Pseudo AA + AA	99.22	98.58
AA + Physiochemical	99.22	98.78
Pseudo AA + Physiochemical	99.09	99.09
Pseudo AA + AA + Physiochemical	99.09	99.80



Parameters	HLH	HTH	Leucine Zipper	Zinc Finger	Histone like	T-Box
Amino Acid Composition	88.58	87.16	85.29	84.26	80.29	75.68
Physiochemical Properties	78.54	61.47	70.59	75.58	72.41	65.37
Pseudo AA Composition	93.61	86.24	91.18	88.74	84.68	95.95
AA + physiochemical	96.81	83.49	92.16	85.63	90.28	92.23
Pseudo AA + AA	94.98	82.57	91.18	86.31	88.25	91.90
Pseudo AA + Physiochemical	97.26	88.07	94.12	89.37	91.24	95.81
Pseudo AA + AA + Physiochemical	98.17	87.16	95.10	88.74	91.38	96.27

- **External Validation**

Validation of the software was done using the dataset that was not used for training or we can say the dataset that we have removed in the process of data cleaning and data scaling. We took 172 protein sequences in DNA Binding category and 184 protein sequences in Non-DNA Binding category.

**Table 3(b) Percentage of correct prediction in DNA Binding and Non-DNA Binding category for External Validation**

Parameters	DNA Binding	Non-DNA Binding
Amino Acid Composition	87.42	89.67
Physiochemical Properties	92.81	92.39
Pseudo AA Composition	81.43	86.41
Pseudo AA + AA	86.22	89.13
AA + Physiochemical	82.63	85.32
Pseudo AA + Physiochemical	96.40	95.10
Pseudo AA + AA + Physiochemical	97.00	89.67

Parameters	HLH	HTH	Leucine Zipper	Zinc Finger	Histone like	T-Box
Amino Acid Composition	34.29	30.85	62.5	56.67	28.0	12.27
Physiochemical Properties	45.71	80.80	81.25	43.33	40.0	33.37
Pseudo AA Composition	57.14	26.23	46.88	36.67	12.0	24.31
Pseudo AA + AA	68.58	11.54	81.25	46.67	28.0	32.67
AA + Physiochemical	42.86	29.17	65.63	43.33	48.0	49.97
Pseudo AA + Physiochemical	71.43	53.84	81.25	70.0	56.0	56.55
Pseudo AA + AA + Physiochemical	82.86	76.92	84.38	80.0	80.0	62.39



## DISCUSSION

The DNA Binding Prediction Tool has been developed in this study using two layered neural network based on sequence derived features. The results demonstrate that the developed ANN based model for binary prediction of DNA Binding Proteins/Non-DNA Binding Proteins and classification of proteins into six major classes is adequate and can be considered an effective tool for 'in silico' screening. The results also demonstrate that the sequence derived parameters readily accessible from the protein sequences only, can produce a variety of useful information to be used 'in silico'; clearly demonstrates an adequacy and good predictive power of the developed ANN model. There is strong evidence, that the introduced sequence features do adequately reflect the structural properties of proteins. The structure of a protein is an important determinant for the detailed molecular function of proteins, and would consequently also be useful for prediction of Binding Proteins/Non-DNA Binding Proteins and for their classification. This agrees well with our result that sequence derived features can be used for predicting DNA Binding Proteins.

Presumably, accuracy of the approach operating by the sequence derived features can be improved even further by expanding the parameters or by applying more powerful classification techniques such as Support Vector Machines or Bayesian Neural Networks. Use of merely statistical techniques in conjunction with the sequence parameters would also be beneficial, as they will allow interpreting individual parameter contributions into Binding Proteins/Non-DNA Binding Proteins.

The results of the present work demonstrate that the sequence derived features with ANN appear to be a very fast protein classification mechanism providing good results, comparable to some of the current efforts in the literature. The developed ANN-based model for Binding Proteins/Non-DNA Binding Proteins prediction and their classification into different classes can be used as a powerful tool for filtering through the collections of genome sequences to discover novel proteins.

## CONCLUSION

From a practical point of view the most important aspect of a prediction method is its ability to make correct predictions. Till date most of the available methods use the 3-d structure of the protein to predict and classify DNA binding proteins. This is a very tedious job and requires much costlier endeavors. The sequence of a protein is an important determinant for the detailed molecular function of proteins, and would consequently also be useful for prediction of DNA binding proteins and in turn into various classes. Therefore, we have developed a tool which predicts the DNA binding proteins and their subsequent classes based on both strategies.

This thesis contains detailed work on DNA binding proteins prediction and classification. We achieved an accuracy of 70% based on dataset of 1749 proteins using the ANN technique. The neural network architecture used for the prediction was optimized for maximum accuracy. This was achieved by gradually testing networks with variable hidden nodes and retaining the one with highest true predictions. This is at par with best prediction tools available till date, but to the contrary, uses a much simpler and efficient prediction method based on sequence features only. This application not only gives optimum result with the dataset used, but also predicts DNA binding proteins from complex genomes to a very high satisfactory level. A much elaborate analysis has been done, which is evident from the extracted data, figures and tables compiled.



## REFERENCES:

- National Cancer Institute: Vaginal Cancer (public domain)
- [Stenchever: Comprehensive Gynecology, 4th ed., Copyright © 2001 Mosby, Inc.]
- The Image Processing Handbook by John C. Russ, ISBN 0849372542 (2006)  
Fundamentals of Image Processing by Ian T. Young, Jan J. Gerbrands, Lucas J. VanVliet, Paperback, ISBN 90-75691-01-7 (1995)
- Image Analysis and Mathematical Morphology by Jean Serra, ISBN 0126372403 (1982)
- Front-End Vision and Multi-Scale Image Analysis by Bart M. ter Haar  
Romeny, Christopher M. Bishop (2007) Pattern Recognition and Machine Learning, Springer ISBN 0-387-31073-8.
- Neural Computing and Applications, Springer-Verlag. (address: Sweetapple Ho, Catteshall Rd., Godalming, GU7 3DJ)
- Bhagat, P.M. (2005) Pattern Recognition in Industry, Elsevier. ISBN 0-08-044538
- Bishop, C.M. (1995) Neural Networks for Pattern Recognition, Oxford: Oxford University Press. ISBN 0-19-853849-9 (hardback) or ISBN 0-19-853864-2 (paperback)
- Duda, R.O., Hart, P.E., Stork, D.G. (2001) Pattern classification (2nd edition), Wiley, ISBN 0-471-05669-3
- Gurney, K. (1997) An Introduction to Neural Networks London: Routledge.
- ISBN 1-85728-673-1 (hardback) or ISBN 1-85728-503-4 (paperback)

- Haykin, S. (1999) *Neural Networks: A Comprehensive Foundation*, Prentice Hall, ISBN 0-13-273350-1
- Altschul, S., Madden, T., Schaffer, A., Zhang, L., Zhang, Z., Miller, W., and Lipman, D. 1997. Gapped BLAST and PSI-BLAST: A new generation of protein database search programs. *Nucleic Acids Res.* 25, 3389-3402.
- Anfinsen, C.B. 1973 Principles that govern the folding of protein chains. *Science* 181, 223-230
- Baker, E.N., Arcus, V.L., and Lott, I.S. 2003. Protein structure prediction and analysis as a tool for functional genomics. *Applied Bioinformatics* 2, (3 Suppl), s3-s10.
- Benson, D., Karsch-Mizrachi, I., Lipman, D., Ostell, L., Rapp, B., and Wheeler, D. 2002. *GenBank*. *Nucleic Acids Res.* 30, 17-20.
- Boberg, J., Salakoski, T. & Vihinen, M. 1995. Accurate prediction of protein secondary structural class with fuzzy structural vectors. *Protein Eng.* 8, 505-512.
- Bork, P., and Koonin, E.V. 1998. Predicting functions from protein sequences - where are the bottlenecks? *Nat Genet.* 18, 313-318.
- Bu, W.S., Feng, Z.P., Zhang, Z.D. and Zhang, C.T. 1999. Prediction of protein (domain) structural classes based on amino-acid index. *Eur. J Biochem.* 266, 1043-1049.
- Cai, C.Z., Wang, W.L., Sun, L.Z., and Chen, Y.Z. 2003. Protein function classification via support vector machine approach. *Math Biosci.* 185(2), 111-122.
- Chou, P.Y. 1989. Prediction of protein structural classes from amino acid composition. In *Prediction of Protein Structures and the Principles of Protein Conformation* (Fasman, G.D., ed.), pp. 549-586. Plenum Press, New York.



- Chou, K.C. and Zhang, C.T. 1995. Prediction of protein structural classes. *Crit. Rev. Biochem. Mol. Biol.* 30, 275-349.
- Chou, K.C., Liu, W.M., Maggiora, G.M. and Zhang, C.T. 1998. Prediction and Classification of domain structural classes. *Proteins: Struct. Funct. Genet.* 31, 97-103.
- DesJardins, M., Karp, P.D., Krummenacker, M., Lee, T.J., and Ouzonis, C.A. 1997. Prediction of enzyme classification from protein sequence without the use of sequence similarity. *Proceedings of the Fifth International Conference on Intelligent Systems for Molecular Biology*, Halkidiki, Greece.
- Dobson, P.D., and Doig, A.J. 2003. Distinguishing Enzyme Structures from Non-Enzymes Without Alignments. *J Mol. Biol.* 330, 771-783.
- Eisenberg, D., Marcotte, C.A., Xenarios, I., and Yeates, T.O. 2000. Protein function in the post-genomic era. *Nature* 405, 823 - 826.
- Jensen, L.J., Skovgaard, M., and Brunak, S. 2002. Prediction of novel archaeal enzymes from sequence-derived features. *Protein Sci.* 11, 2894-2898.
- King, R.D., Paul, H., and Clare, A. 2004. Confirmation of data mining based predictions of protein function. *Bioinformatics* 20, 1110-1118.
- King, R.D., Karwath, A., Clare, A., and Dehaspe, L. 2000. Accurate prediction of protein functional class from sequence in the *M. tuberculosis* and *E. coli* genomes using data mining. *Yeast-Comparative and Functional Genomics* 17(4), 283 - 293.
- Klein, P. 1986. Prediction of protein structural class by discriminant analysis. *Biochem. Biophys. Acta.* 874, 205-215.

- Nagl, S. 2003. Function prediction from protein sequence. In Orengo, c.A., Jones, D.T. Thompson, J.M. (eds). *Bioinformatics - Genes, proteins and computers*. BIOS Scientific publishers. Oxford. 298 pp.
- Nishikawa, .and Ooi, T. 1982. Correlation of amino acid composition of a protein to its structural and biological characters. *J Biochem.* 91, 1821-1824.
- Nishikawa, K., Kubota, Y. and Ooi, T. 1983. Classification of proteins into groups based on amino acid composition and other characters. I. Angular distribution. *J Biochem.* 94,
- Nakashima, K., Kubota, Y. and Ooi, T. 1983. Classification of the proteins into groups based on amino acid composition and other characters. II, Grouping into four types. *J Biochem.* 94, 997-1007.
- Nakashima H., Nishikawa, K. and Ooi, T. 1986. The folding type of a protein is relevant to the Amino acid composition. *J Biochem.* 99, 152-162.
- Pasquier C., Promponas, V., and Hamodrakas, S.J. 2001. PRED-CLASS: Cascading Neural networks for generalized protein classification and genome wide applications . *Proteins* 44, 361-369.
- Pellegrini, M., Marcotte, E., Thompson, M., Eisenberg, D., and Yeates, T. 1999. Assigning protein functions by comparative genome analysis: Protein phylogenetic profiles. *Proc. Natl. Acad. Sci.* 38, 667-677
- Rice, P., Longden, I., and Bleasby, A. 2000. EMBOSS: The European Molecular Biology Open Software Suite. *Trends in Genetics* 16, (6) pp276-277.
- Schomburg, J., Chang, A., Ebeling, c., Gremse, M., Heldt, c., Huhn, G., and Schomburg, D. 2004. BRENDA, the enzyme database: updates and major new developments. *Nucleic*



Acids Res. Jan1,32(Database issue):D43, 1-3.

- Tian, W., and Skolnick, I. 2003. How well is enzyme function conserved as a function of Pairwise sequence identity? *J Mol. Biol.* 333, 863-882.
- Wu, C., Berry, M., Shivakumar, S., and McLarty, J. 1995. Neural Networks for Full-Scale protein Sequence Classification: Sequence Encoding with singular Value Decomposition. *J Mach. Learn.* 21 N(1-2), 177-193
- Zell, A., and Mamier, G. 1997. Stuttgart Neural Network Simulator version 4.2. University of Stuttgart, Stuttgart, Germany.
- Janssen P, Audit B, Cases I, Darzentas N, Goldovsky L, Kunin V, Lopez-Bigas N, Peregrin-Alvarez JM, Pereira-Leal JB, Tsoka S, Ouzounis CA: Beyond 100 genomes. *Genome Biol* 2003, 4:402.
- Andrade MA, Sander C: Bioinformatics: from genome data to biological knowledge. *Curr Opin Biotechnol* 1997, 8:675-683.
- Karp PD: What we do not know about sequence analysis and sequence databases. *Bioinformatics* 1998, 14:753-754. 4. Pearson WR: Rapid and sensitive sequence comparison with FASTP and FASTA. *Methods Enzymol* 1990, 183:63-98.
- Shah I, Hunter L: Predicting enzyme function from sequence: a systematic appraisal. *Proc Int Conf Intell Syst Mol Biol* 1997, 5:276-283.
- Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ: Basic local alignment search tool. *J Mol Biol* 1990, 215:403-410.

- Altschul SF, Madden TL, Schaffer AA, Zhang J, Zhang Z, Miller W, Lipman DJ: GappedBLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res* 1997,25:3389-3402.
- Krogh A, Brown M, Mian IS, Sjolander K, Haussler D: Hidden Markov models in computational biology. Applications to protein modeling. *J Mol Biol* 1994, 235:1501-1531.
- Vinga S, Almeida J: Alignment-free sequence comparison-a review. *Bioinformatics* 2003,19:513-523.
- Vries JK, Munshi R, Tobi D, Klein-Seetharaman J, Benos PV, Bahar I: A sequence alignment-independent method for protein classification. *Appl Bioinformatics* 2004,3:137-148.
- Abascal F, Valencia A: Automatic annotation of protein function based on family identification. *Proteins* 2003,53:683-692.
- Krebs WG, Bourne PE: Statistically rigorous automated protein annotation. *Bioinformatics* 2004, 20:1066-1073.
- Leontovich AM, Brodsky LI, Drachev VA, Nikolaev VK: Adaptive algorithm of automated annotation. *Bioinformatics* 2002, 18:838-844.
- Tatusov RL, Galperin MY, Natale DA, Koonin EV: The COG database: a tool for genome-scale analysis of protein functions and evolution. *Nucleic Acids Res* 2000, 28:33-36.



- Andrade MA, Brown NP, Leroy C, Hoersch S, de Daruvar A, Reich C, Franchini A, Tamames J, Valencia A, Ouzounis C, Sander C: Auto-mated genome sequence analysis and annotation. *Bioinformatics* 1999, 15:391-412.
- Wilson CA, Kreychman J, Gerstein M: Assessing annotation transfer for genomics: quantifying the relations between protein sequence, structure and function through traditional and probabilistic scores. *J Mol Biol* 2000, 297:233-249.
- Kyrpides NC, Ouzounis CA: Whole-genome sequence annotation: 'Going wrong with confidence'. *Mol Microbiol* 1999, 32:886-887.
- Bork P, Koonin EV: Predicting functions from protein sequences--where are the bottlenecks? *Nat Genet* 1998, 18:313-318.
- Devos D, Valencia A: Intrinsic errors in genome annotation. *Trends Genet* 2001, 17:429-431.
- Gerlt JA, Babbitt PC: Can sequence determine function? *Genome Biol* 2000, 1:REVIEWS0005
- Gilks WR, Audit B, De Angelis D, Tsoka S, Ouzounis CA: Modeling the percolation of annotation errors in a database of protein sequences. *Bioinformatics* 2002, 18: 1641-
- Cheng BY, Carbonell JG, Klein-Seethararnan J: Protein classification based on text document classification techniques. *Proteins* 2005, 58:955-970.
- Jardins M, Karp PD, Krummenacker M, Lee TJ, Ouzounis CA: Prediction of Enzyme classification from protein sequence without the use of sequence similarity. *Proc Int Conf Intell Syst Mol Biol* 1997, 5:92-99.

- Karchin R, Karplus K, Haussler D: Classifying G-protein coupled receptors with support vector machines. *Bioinformatics* 2002, 18: 147-159.
- Fillinger S, Boschi-Muller S, Azza S, Dervyn E, Branlant G, Aymerich S: Two glyceraldehyde-3-phosphate dehydrogenases with opposite physiological roles in a non photosynthetic bacterium. *J BioI Chem* 2000, 275:14031-14037.
- Sigrist CJ, Cerutti L, Hulo N, Gattiker A, Falquet L, Pagni M, Bairoch A, Bucher P: PROSITE: a documented database using patterns and profiles as motif descriptors. *Brief Bioinform* 2002, 3:265-274.
- Wen Z, Morrison M: The NAD(P)H-dependent glutamate dehydrogenase activities of *Prevotella ruminicola* B(1)4 can be attributed to one enzyme (GdhA), and *gdhA* expression is regulated in response to the nitrogen source available for growth. *Appl Environ Microbiol* 1996, 62:3826-3833.
- Itkor P, Tsukagoshi N, Udaoka S: Nucleotide sequence of the raw starch- digesting amylase gene from *Bacillus* sp. B10 18 and its strong homology to the cyclodextrin glucanotransferase genes. *Biochem Biophys Res Commun* 1990, 166:630-636.
- Shah I, Hunter L: Predicting enzyme function from sequence: a systematic appraisal. *Proc Int Conf Intell Syst Mol Biol* 1997, 5:276-283.
- Devos D, Valencia A: Practical limits of function prediction. *Proteins* 2000, 41 :98-107.
- Ashburner M, Ball CA, Blake JA, Botstein D, Butler H, Cherry JM, Davis AP, Dolinski K, Dwight SS, Eppig JT, Harris MA, Hill DP, Issel-Tarver L, Kasarskis A, Lewis S, Matese JC, Richardson JE, Ringwald M, Rubin GM, Sherlock G: Gene ontology: Tool for the unification of biology. *Nat Genet* 2000, 25:25-29.



- Gattiker A, Michoud K, Rivoire C, Auchincloss AH, Coudert E, Lima T, Kersey P, Pagni M, Sigrist CJ, Lachaize C, Veuthey AL, Gasteiger E, Bairoch A: Automated annotation of microbial proteomes in SWISS-PROT. *Comput Biol Chem* 2003, 27:49-58.
- Wieser D, Kretschmann E, Apweiler R: Filtering erroneous protein annotation. *Bioinformatics* 2004, 20 Suppl 1:1342-1347.
- Murzin AG, Brenner SE, Hubbard T, Chothia C: SCOP: a structural classification of proteins database for the investigation of sequences and structures. *J Mol Biol* 1995, 247:536-540.
- Holm L, Sander C: Mapping the protein universe. *Science* 1996, 273:595-603.
- Jaakkola T, Diekhans M, Haussler D: A discriminative framework for detecting remote protein homologies. *J Comput Biol* 2000, 7:95-114.
- Bairoch A: The ENZYME database in 2000. *Nucleic Acids Res* 2000, 28:304-305.
- Boeckmann B, Bairoch A, Apweiler R, Blatter MC, Estreicher A, Gasteiger E, Martin MJ, Michoud K, O'Donovan C, Phan I, Pilbout S, Schneider M: The SWISS-PROT protein knowledgebase and its supplement TrEMBL in 2003. *Nucleic Acids Res* 2003, 31:365-370.
- Promponas VJ, Enright AJ, Tsoka S, Kreil DP, Leroy C, Hamodrakas S, Sander C, Ouzounis CA: CAST: an iterative algorithm for the complexity analysis of sequence tracts. Complexity analysis of sequence tracts. *Bioinformatics* 2000, 16:915-922.
- N. M. Laird A. P. Dempster and D. B. Rubin. Maximum likelihood estimation from incomplete data. *Journal of the Royal Statistics Society*, 39:1-38, 1977.
- L. E. Baum. An inequality and associated technique occurring in the

- statistical analysis of probabilistic functions of markov chains. *Inequalities*, 3:1-8, 1972.
- R. Bellman. *Dynamic Programming*. Princeton University Press, Boston, 1957.
- R. Durbin, S. Eddy, A. Krough, and G. Mitchison. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, 1998.
- Krogh, M. Brown, S. Mian, M. Sj"olander, and D. Haussler. Hidden markov models in computational biology. applications to protein modeling. *Journal of Molecular Biology*, 235(5):1501-1531, 4 February 1994.
- Anders Krogh, I. Saira Mian, and David Haussler. A hidden markov model that finds genes in E. coli DNA. *Nucleic Acids Research*, 22:4768-4778, 1994.
- D. Kulp, D. Haussler, M.G. Reese, and F.H. Eeckman. A generalized hidden Markov model for the recognition of human genes in DNA. In D. J. States, P. Agarwal, T. Gaasterland, L. Hunter, and R. Smith, editors, *Proc. Conf. On Intelligent Systems in Molecular Biology '96*, pages 134-142. AAAI/MIT Press, 1996. St. Louis, Mo.
- Charles E. Lawrence, Stephen F. Altschul, Mark S. Boguski, Jun S. Liu, Andrew F. Neuwald, and John C. Wootton. Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignment. *Science*, 262:208-214, 8 October 1993.
- L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257-286, February 1989.



## APPENDIX 1: WEBPAGE OF THE SITE

The screenshot shows a web browser window displaying the 'DNA BINDING PREDICTION TOOL' website. The website has a blue header with a logo on the left and the title 'DNA BINDING PREDICTION TOOL' in the center. Below the header is a navigation bar with links: Home, About, Help, and Contact Us. The main content area is light blue and contains three checkboxes: 'Amino Acid Composition' (checked), 'Pseudo Amino Acid' (unchecked), and 'Pepstat' (unchecked). Below these checkboxes is a large, empty text input box with a vertical scrollbar on the right. At the bottom of the input box is a 'Submit' button. The browser's status bar at the bottom shows 'Internet' and '100%' zoom.

**DNA BINDING PREDICTION TOOL**

Home About Help Contact Us

☒ Amino Acid Composition  
☐ Pseudo Amino Acid  
☐ Pepstat

Submit

Internet 100%



## APPENDIX 2:

1. Pseudo Amino Acid Composition						
Class	Train			Test		
	Total	Correct	Correct %	Total	Correct	Correct %
Binding	615	615	100	151	139	92.37
Non Binding	785	785	100	198	194	97.84
2. Amino Acid Composition						
Class	Train			Test		
	Total	Correct	Correct %	Total	Correct	Correct %
Binding	615	615	100	151	143	94.11
Non Binding	785	785	100	198	184	92.39
3. Physicochemical Properties						
Class	Train			Test		
	Total	Correct	Correct %	Total	Correct	Correct %
Binding	615	583	94	151	136	90.21
Non Binding	785	770	98	198	182	91.95
4. Pseudo Amino Acid Composition + Amino Acid Composition						
Class	Train			Test		
	Total	Correct	Correct %	Total	Correct	Correct %
Binding	615	615	100	151	146	96.78
Non Binding	785	785	100	198	188	94.97
5. Pseudo Amino Acid Composition + Physicochemical Properties						
Class	Train			Test		
	Total	Correct	Correct %	Total	Correct	Correct %
Binding	615	615	100	151	142	93.96
Non Binding	785	785	100	198	191	96.28
6. Amino Acid Composition + Physicochemical Properties						
Class	Train			Test		
	Total	Correct	Correct %	Total	Correct	Correct %
Binding	615	615	100	151	142	94.26
Non Binding	785	785	100	198	181	91.47
7. Pseudo Amino Acid Composition + Amino Acid Composition + Physicochemical Properties						
Class	Train			Test		
	Total	Correct	Correct %	Total	Correct	Correct %
Binding	615	615	100	151	149	98.87
Non Binding	785	785	100	198	193	97.29
TABLE 3: MODEL SUMMARY OF LAYER 1						



1. Pseudo Amino Acid Composition						
Class	Train			Test		
	Total	Correct	Correct %	Total	Correct	Correct %
Helix-Loop-Helix	177	177	100	44	33	74.26
Helix-Turn-Helix	86	81	94	21	15	73.75
Zinc Finger	75	75	100	19	14	72.17
Leucine Zipper	82	79	96	20	14	69.48
Histone Binding	137	136	99	34	25	73.27
T-Box	56	56	100	14	11	75.31
2. Amino Acid Composition						
Class	Train			Test		
	Total	Correct	Correct %	Total	Correct	Correct %
Helix-Loop-Helix	177	162	91	44	31	69.93
Helix-Turn-Helix	86	76	88	21	14	67.83
Zinc Finger	75	72	96	19	14	72.26
Leucine Zipper	82	73	89	20	14	70.18
Histone Binding	137	115	83	34	23	68.31
T-Box	56	46	82	14	10	74.95
3. Physicochemical Properties						
Class	Train			Test		
	Total	Correct	Correct %	Total	Correct	Correct %
Helix-Loop-Helix	177	151	85	44	31	70.15
Helix-Turn-Helix	86	68	79	21	14	67.32
Zinc Finger	75	62	82	19	13	70.13
Leucine Zipper	82	53	64	20	14	68.25
Histone Binding	137	105	76	34	22	65.33
T-Box	56	44	78	14	11	75.32
4. Pseudo Amino Acid Composition + Amino Acid Composition						
Class	Train			Test		
	Total	Correct	Correct %	Total	Correct	Correct %
Helix-Loop-Helix	177	177	100	44	31	69.57
Helix-Turn-Helix	86	81	94	21	15	71.24
Zinc Finger	75	75	100	19	13	69.43
Leucine Zipper	82	80	97	20	15	73.64
Histone Binding	137	136	99	34	24	70.16
T-Box	56	56	100	14	11	76.34

5. Pseudo Amino Acid Composition + Physicochemical Properties						
Class	Train			Test		
	Total	Correct	Correct %	Total	Correct	Correct %
Helix-Loop-Helix	177	176	99	44	38	85.62
Helix-Turn-Helix	86	82	95	21	17	81.04
Zinc Finger	75	75	100	19	15	80.95
Leucine Zipper	82	79	96	20	16	79.38
Histone Binding	137	137	100	34	28	83.21
T-Box	56	56	100	14	11	75.46
6. Amino Acid Composition + Physicochemical Properties						
Class	Train			Test		
	Total	Correct	Correct %	Total	Correct	Correct %
Helix-Loop-Helix	177	177	100	44	33	75.81
Helix-Turn-Helix	86	80	93	21	15	73.26
Zinc Finger	75	73	97	19	14	74.31
Leucine Zipper	82	78	95	20	16	79.22
Histone Binding	137	135	98	34	23	67.94
T-Box	56	55	98	14	10	71.47
7. Pseudo Amino Acid Composition + Amino Acid Composition + Physicochemical Properties						
Class	Train			Test		
	Total	Correct	Correct %	Total	Correct	Correct %
Helix-Loop-Helix	177	176	99	44	36	82.61
Helix-Turn-Helix	86	83	96	21	17	81.70
Zinc Finger	75	74	98	19	16	85.39
Leucine Zipper	82	79	96	20	16	80.05
Histone Binding	137	137	100	34	27	79.37
T-Box	56	56	100	14	11	77.82

**TABLE 4: MODEL SUMMARY FOR SECOND LAYER**



### APPENDIX 3: C Codes

#### 7\_formatter\_and\_prediction\_execution.pl

```
#formatting input sequence given by user so that it can be used by Prediction cluster
#opening sequence file
open(INP, "par.txt");
while(<INP>)
{
    if($_ !~ />/)
    {
        chomp($_);
        $seq=$seq.$_;
    }
}
close(INP);

#updating sequence file
$seq =~ tr/a-z/A-Z/;
open(OUT, ">par.txt");
print OUT ">Query|PDBID|CHAIN|SEQUENCE\n$seq";
close (OUT);

#prediction execution
system "modified pseaa_desc_calc.exe";
`7_modified_aa_composition_desc_calc.pl`;
system "pepstats.exe par.txt pepstat.xls";
`7_modified_pepstats_parsing.pl`;
system "7_pseaa_aa_pepstat_ANN.exe";
```

```
`del par.xls`;
`del pepstat.xls`;
`del par.txt`;
```

---

## 7\_modified\_aa\_composition\_desc\_calc.pl

```
#opening pseaa descriptor file
open(PSEAA,"par.xls");
while(<PSEAA>)
{
    chomp($_);
    @pseaa=split(/\s+/, $_);
}
close(PSEAA);

#Amino Acid Composition Based Descriptors
#inputting file
#print "\nInput filename (.txt):\t";
#$filename=<>;
open (file,"par.txt")
    or print "cannot open sequence file";

#reading file into array
$i=0;
while(<file>)
{
    if(/^>/)
```

```

        {
            $i++;
            $name[$i]=$_;
        }
    else
        {
            chomp($_);
            $seq[$i]=$seq[$i].$_;
        }
    }
close(file);

#Reference array
$ref=(ACDEFGHIKLMNPQRSTUVWY);
@ref=split("",$ref);

#output file open
#print "\nenter output filename: ";
#$out=<>;
open (desc,"+>par.xls");

#printing PSEAA descriptors
foreach $y(@pseaa)
    {
        print desc "$y\t";
    }

```



#opening sequence and calculating frequency of amino acids

```
for($i=1;$i<$#name+1;$i++)
{
    @pro=();
    @pro=split(",",$seq[$i]);
    for($y=0;$y<$#ref+1;$y++)
    {
        $freq[$y]=0;
    }
    foreach $aa(@pro)
    {
        for($j=0;$j<$#ref+1;$j++)
        {
            if ($aa eq $ref[$j])
            {
                $freq[$j]+=1;
            }
        }
    }

    $prname=(split /[ ]/, $name[$i])[0];
    print "protein: $prname\t@freq\n";
    #print desc "$prname\t";
    for($k=0;$k<$#ref+1;$k++)
    {
        $probab=$freq[$k]/($#pro+1);
        if($freq[$k] eq 0)
        {
```

```

        $probab=0;
    }
    print desc "$probab\t";
}
print desc "\n";
}
close(desc);

```

---

### 7\_modified\_pepstats\_parsing.pl

```

#opening PSEAA+AA descriptors
open (PSEAA_AA,"par.xls")
    or print "cannot open sequence file";
$c=0;
while(<PSEAA_AA>)
{
    chomp($_);
    $pseaa_aa[$c]=$_;
    $c++;
}
close(PSEAA_AA);

#PEPSTATS Parsing
#inputting file
#print "\nInput filename (.xls):\t";
#$filename=<>;
open (INP,"pepstat.xls")

```

```

        or print "cannot open sequence file";
$c=0;
while(<INP>)
{
    chomp($_);
    $file[$c]=$_;
    $c++;
}
close(INP);

#output file open
#print "\nenter output filename: ";
#$out=<>;
open (OUT, ">par.xls");

#parsing PSEAA+AA descriptors
foreach $y(@pseaa_aa)
{
    print OUT "$y\t";
}

#parsing output of pepstat
foreach $y(@file)
{
    @line = ();
    @line = split (/s+/, $y);
    if ($line[0] eq "Molecular")

```



```

        {
            print OUT "$line[3]\t";
        }
    elif ($line[0] eq "Average")
    {
        print OUT "$line[7]\t";
    }
    elif ($line[0] eq "Isoelectric")
    {
        print OUT "$line[3]\t";
    }
    elif ($line[0] eq "Tiny")
    {
        print OUT "$line[3]\t";
    }
    elif ($line[0] eq "Small")
    {
        print OUT "$line[3]\t";
    }
    elif ($line[0] eq "Aliphatic")
    {
        print OUT "$line[3]\t";
    }
    elif ($line[0] eq "Aromatic")
    {
        print OUT "$line[3]\t";
    }

```

```

elseif($line[0] eq "Non-polar")
{
    print OUT "$line[3]\t";
}

elseif($line[0] eq "Polar")
{
    print OUT "$line[3]\t";
}

elseif($line[0] eq "Charged")
{
    print OUT "$line[3]\t";
}

elseif($line[0] eq "Basic")
{
    print OUT "$line[3]\t";
}

elseif($line[0] eq "Acidic")
{
    print OUT "$line[3]\t";
}

elseif($line[0] eq "PEPSTATS")
{
    #print OUT "\n";
}

}

print "DONE!!";

```

```
close(OUT);
```

---

```
binding_standalone.pl
```

```
#!C:/Perl/bin/perl.exe
```

```
##standalone prediction program
```

```
print "\n\t\t...SEQUENCE PREDICTION...\n\t\t(Standalone Version)";
```

```
#reading in the sequence
```

```
print "\n\nEnter the file with Sequences(in fasta format): ";
```

```
$file=<>;
```

```
chomp($file);
```

```
open(SEQ, "$file") or print "\nError!!\nCannot open sequence file";
```

```
while(<SEQ>)
```

```
{
```

```
  if(/>/)
```

```
    {
```

```
      $c=-1;
```

```
      last;
```

```
    }
```

```
  else
```

```
    {
```

```
      $c=0;
```

```
    }
```

```
}
```

```
close(SEQ);
```

```
open(SEQ, "$file") or print "\nError!!\nCannot open sequence file";
```



```

while(<SEQ>)
{
    if(/>/)
    {
        $c++;
        chomp($_);
        $name[$c]=$_;
    }
    else
    {
        chomp($_);
        $seq[$c]=$seq[$c].$_;
    }
}

#taking in the choice of parameters
print "\nPrediction through Pseudo Amino Acid Composition(37 factors)(Y/N): ";
$pseaa=<>;
chomp($pseaa);
print "\nPrediction through Amino Acid Composition(20 factors)(Y/N): ";
$aac=<>;
chomp($aac);
print "\nPrediction through PhysicoChemical Properties(12 factors)(Y/N): ";
$pep=<>;
chomp($pep);

#opening Output file
print "\nEnter the Output Filename: ";

```

```

$file=<>;
chomp($file);
open(OUT, ">$file");

#Prediction Segment
for($i=0;$i<=$#seq;$i++)
{
#preparing input sequence file
open(PAR, ">par.txt");
print PAR $seq[$i];
close(PAR);

#STARTING PREDICTION based on the choice of parameters(user given)
#firing predictor executers accordingly
if(($pseaa =~ /y/i) && ($aa !~ /y/i) && ($pep !~ /y/i))
{
    `1_formatter_and_prediction_execution.pl`;
}
elseif(($pseaa !~ /y/i) && ($aa =~ /y/i) && ($pep !~ /y/i))
{
    `2_formatter_and_prediction_execution.pl`;
}
elseif(($pseaa !~ /y/i) && ($aa !~ /y/i) && ($pep =~ /y/i))
{
    `3_formatter_and_prediction_execution.pl`;
}
elseif(($pseaa =~ /y/i) && ($aa =~ /y/i) && ($pep !~ /y/i))
{

```

```

        `4_formatter_and_prediction_execution.pl`;
    }
elseif(($pseaa =~ /y/i) && ($aa !~ /y/i) && ($pep =~ /y/i))
    {
        `5_formatter_and_prediction_execution.pl`;
    }
elseif(($pseaa !~ /y/i) && ($aa =~ /y/i) && ($pep =~ /y/i))
    {
        `6_formatter_and_prediction_execution.pl`;
    }
elseif(($pseaa =~ /y/i) && ($aa =~ /y/i) && ($pep =~ /y/i))
    {
        `7_formatter_and_prediction_execution.pl`;
    }

#printing the output
open(INP,"binding_out.txt");
@output=<INP>;
close(INP);
`del binding_out.txt`;
print OUT "binding for protein: $name[$i]\n@output\n";
print "\n binding for protein: $name[$i]\n\n@output";
}
close(OUT);
print "\n\nPrediction Complete...\n\nPress enter to terminate...";
<>

```