METALLOPRED: AN ONLINE TOOL FOR HIERARCHICAL PREDICTION OF METAL ION BINDING PROTEINS USING CLUSTER OF NEURAL NETWORKS AND SEQUENCE DERIVED FEATURES

By

Pooja Kesari (061509)

Piyush Ranjan (061517)

THESIS SUBMITTED IN FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

Bachelor of Technology
IN
BIOINFORMATICS

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY
WAKNAGHAT, SOLAN - 173215, HIMACHAL PRADESH,

INDIA

May 2010

Dr. Pradeep Kumar Naik
Assistant Professor
Biotechnology & Bioinformatics

Jaypee University of
Information Technology
Waknaghat-173215
Solan, Himachal Pradesh
Phone No.: 91-1792-239227
Fax No.: 91-1792-245362

CERTIFICATE

This is to certify that the thesis entitled "MetalloPred: an online tool for hierarchical prediction of metal ion binding proteins using cluster of neural networks and sequence derived features" submitted by **Piyush Ranjan and Pooja Kesari** to the Jaypee University of Information Technology, Waknaghat in partial fulfillment of the requirement for the award of the degree of **Bachelor of Technology in Bioinformatics (Science)** is a record of bona fide research work carried out by both of them under my supervision and guidance and no part of this work has been submitted for any other degree or diploma.

(Dr. P. K. Naik)

DECLARATION

I hereby declare that the work presented in this thesis has been carried out by both of us under the supervision of Dr. Pradeep Kumar Naik, Department of Biotechnology & Bioinformatics, Jaypee University of Information Technology, Waknaghat, Solan-173215, Himachal Pradesh, and has not been submitted for a degree or diploma of any other university. All assistance and help received during the course of the investigation has been duly acknowledged.

Poojakesari

Piyush Ranjan Pooja Kesari

ACKNOWLEDGEMENT

"To speak gratitude is courteous and pleasant, to enact gratitude is generous and noble, but to live gratitude is to touch Heaven"

We would like to express our heartfelt gratitude to all those who have contributed directly or indirectly towards obtaining our bachelor degree. We are highly indebted to our esteemed supervisor, Dr. Pradeep Kumar Naik, who has guided us through thick and thin. Many people have contributed to this project in a variety of ways over the past few months. To the individuals who have helped us, we again express our appreciation. We express our special thanks to Ms. Somlata for helping us in downloading the dataset and we also give our indebt thanks to Mr. Shrikrishna Chanumolu for guiding us in development of the server edition of the tool. We also acknowledge the many helpful comments Received from our Teachers of the bioinformatics department. We are indebted to all those who provided reviews & suggestion for improving the result and the topics covered in our project, and we extended our apologies to anyone we may have failed to mention.

PoojaKesari

Piyush Ranjan

Pooja Kesari

CONTENTS

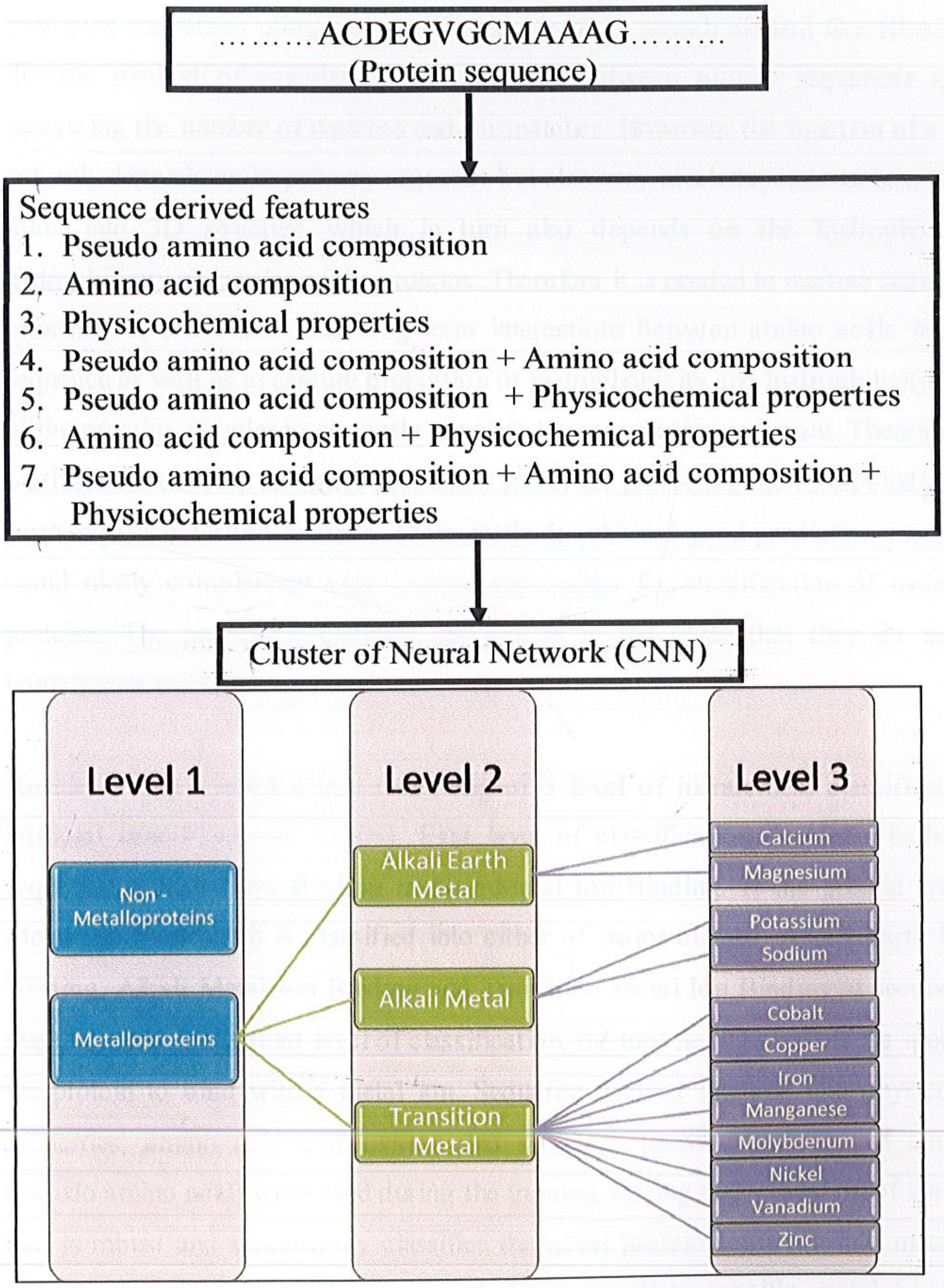
List of Figures:

List of Tables:

List of abbreviations

ANN	Artificial Neural Network
BFGS Algorithm	Broyden-Fletcher-Goldfarb-Shanno Algorithm
pI	Iso-electric Point
pepstat	Protein Statistics (program used for calculation of physicochemical properties)
SANN	Statistica Automated Neural Network
ANS	Automated Network Search
AA comp	Amino Acid Composition
PseAA	Pseudo Amino Acid Composition

MetalloPred : A tool for hierarchical prediction of Metal Ion Binding Proteins



………ACDEGVGCMAAAG………
(Protein sequence)

Sequence derived features
1. Pseudo amino acid composition
2. Amino acid composition
3. Physicochemical properties
4. Pseudo amino acid composition + Amino acid composition
5. Pseudo amino acid composition + Physicochemical properties
6. Amino acid composition + Physicochemical properties
7. Pseudo amino acid composition + Amino acid composition + Physicochemical properties

Cluster of Neural Network (CNN)

ABSTRACT

Motivation: The large amount of proteomic data is available for a variety of organisms, 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 using sequence based similarity search method like BLAST. It is a discrete method of calculating the similarity between protein sequences simply by measuring the number of matches and mismatches. However, the function of a protein is not only depends on its primary sequence but also very much depends on how the protein folds into 3D structure which in turn also depends on the hydrophobicity and hydrophilicity properties of the proteins. Therefore it is needed to capture sequence order information, short term and long term interactions between amino acids in a protein sequence as well as to capture proportion of hydrophobicity and hydrophilicity properties of the proteins in order to correctly annotate the raw protein sequence. Therefore, we are motivated to develop an online prediction server for predicting the metal binding proteins from sequence derived features. These methods achieved good prediction accuracies and could nicely complement experimental approaches for identification of metal binding proteins. The prediction methods are unique in the sense that they do not require homologous protein sequences.

Result: We developed a tool consisting of 3 level of hierarchical classification using artificial neural network (ANN). First layer of classification decides whether protein sequence is Metal Ion Binding or Non-Metal Ion Binding. If the protein sequence is Metal Ion Binding, it is classified into either of major classes, Alkali Earth Metal Ion Binding, Alkali Metal Ion Binding and Transition Metal Ion Binding at second level of classification. In the third level of classification, the tool finally predicts the specificity of the protein to bind with a metal ion. Sequence derived features like physicochemical properties; amino acid composition and sequence based correlation of amino acids (pseudo amino acid) were used during the training, testing and validation of the tool. Our tool is robust and successfully classifies the novel protein sequence into metal binding protein, then into its major class and finally predicts specific metal binding. The performance accuracy of our tool at 1st layer is 76.05%, at 2nd layer is 77.48% and at 3rd layer is 84.44%. Using Jackknifing validation technique the performance accuracy of our tool is 66.46% at 1st layer, 68.55% at 2nd layer and 73.73% at the 3rd layer.

Availability: The Metallopred tool is available for free use to non commercial users and can be downloaded to be used in-house as a standalone from following link. **http://www.juit.ac.in**

CHAPTER 1

METALLOPROTEINS

INTRODUCTION

The 'metalloproteins' have captivated chemists and biochemists, particularly since the 1950s, when the first X-ray crystal structure of a protein, sperm whale myoglobin, indicated the presence of an iron atom. They account for nearly half of all proteins in the nature. Protein metal-binding sites are responsible for catalyzing important biological processes, such as photosynthesis, respiration, water oxidation, molecular oxygen reduction and nitrogen fixation. This reaction involves H_2, N_2, CO, CO_2 and CH_4 which are likely to have been central to the origin of life. This is indicated by the active-site structures of the enzymes involved, which are often reminiscent of minerals. Although the reactions are based on metal centres, the protein matrix regulates reactivity. Much effort has been devoted to understanding the structure and function of these proteins. With the automation in genome sequencing projects huge amount of data were generated each day. However, it is now more important to annotate these sequencing data with the help of computational methods. Therefore the requirement is to develop automated computational methods for the annotation. With this challenge here in this study we have developed an online automated tool for the annotation of metal binding proteins.

Metalloprotein is a generic term for a protein that contains a metal ion cofactor. Metalloproteins have many different functions in cells as enzymes, transport and storage proteins, signal transduction proteins, etc.. The metal ion is usually coordinated by nitrogen, oxygen or sulfur atoms belonging to amino acids in the polypeptide chain and/or a macro cyclic ligand incorporated into the protein. The presence of the metal ion allows metalloenzymes to perform functions such as redox reactions that cannot easily be performed by the limited set of functional groups found in amino acids.

1.1 How metalloproteins evolve in nature?

The most popular autotrophic theory of the origin of life postulates that primordial metabolisms developed on mineral iron–sulphur surfaces under reducing conditions. During this period of the Earth's evolution, between 4.6 and 3.5 billion years ago, the atmosphere was probably rich in gases such as H_2, CO and CO_2, and its hot oceans contained relatively high concentrations of transition-metal ions such as Fe^{2+} and Ni^{2+}.

The physical and chemical properties of a selected metal satisfied a protein's need to form structure, as for zinc-fingers, or to drive catalysis. Proteins evolved to use those metals that at least once were, most accessible. A tenet of the cell biology of metals is that some metals tend to bind organic molecules more avidly than others. The natural order of stability for divalent metals, often called the Irving–Williams series, sets out a resulting trend with copper and zinc forming the tightest complexes, then nickel and cobalt, followed by ferrous iron and manganese and finally, forming the weakest complexes, calcium and magnesium.

1.2 Why are metalloproteins useful?

Metal ions present in the proteins help in its structure, function and stability. The study of these cofactors falls under the area of bioinorganic chemistry. In nutrition, the list of essential trace elements reflects their role as cofactors. In humans this list commonly includes iron, manganese, cobalt, copper, zinc, and molybdenum. Iodine is also an essential trace element, but this element is used as part of the structure of thyroid hormones rather than as an enzyme cofactor. Calcium is another special case, in that it is required as a component of the human diet, and it is needed for the full activity of many enzymes: such as nitric oxide synthase, protein phosphatases or adenylate kinase, but calcium activates these enzymes in allosteric regulation, often binding to these enzymes in a complex with Calmodulin. Calcium is therefore a cell signaling molecule, and not usually considered as a cofactor of the enzymes it regulates.

Other organisms require additional metals as enzyme cofactors, such as vanadium in the nitrogenase of the nitrogen-fixing bacteria of the genus *Azotobacter*, tungsten in the aldehyde ferredoxin oxidoreductase of the thermophilic archaean *Pyrococcus furiosus*, and even cadmium in the carbonic anhydrase from the marine diatom *Thalassiosira weissflogii* .In many cases, the cofactor includes both an inorganic and organic component. One diverse set of examples are the haem proteins, which consists of a porphyrin ring coordinated to iron.

1.2.1 Storage and transport metalloproteins

1.2.1.1 Oxygen carriers proteins

Hemoglobin, which is the principal oxygen carrier in humans, has four sub-units in which the iron (II) ion is coordinated by the planar, macrocyclic ligand protoporphyrin IX (PIX) and the imidazole nitrogen atom of a histidine residue. The sixth coordination site contains a water molecule or a dioxygen molecule. Myoglobin has only one such unit. The active site is located in a hydrophobic pocket. In both hemoglobin and Myoglobin it is known that the diamagnetic nature of these species is due to the fact that the iron (II) is in the low-spin state. Hemerythrin is another iron-containing oxygen carrier. The oxygen binding site is a binuclear iron center. The iron atoms are coordinated to the protein through the carboxylate side chains of a glutamate and aspartate and five histidine residues. Hemocyanins carry oxygen in the blood of most mollusks, and some arthropods. They are second only to hemoglobin in biological popularity of use in oxygen transport. On oxygenation the two copper (I) atoms at the active site are oxidised to copper (II) and the dioxygen molecules is reduced to peroxide, O_2^{2-}.

1.2.1.2 Cytochrome

Cytochromes function as electron-transfer vectors. The iron atom in most cytochromes is contained in a heme group. Figure 1.1 represents iron atom interacting with the four sulphur atoms, one of each cysteine amino acid present in the cytochrome. The differences between the cytochromes lie in the different side-chains. For instance Cytochrome-a has a heme-a prosthetic group and Cytochrome-b has a heme-b prosthetic group. These differences result in different Fe^{2+}/Fe^{3+} redox potentials such that various cytochromes are involved in the mitochondrial electron transport chain. Cytochromes P450 enzymes perform the function of inserting an oxygen atom into a C—H bond, an oxidation reaction.

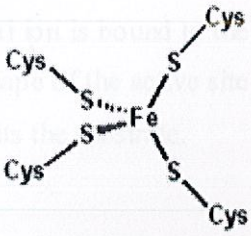


Figure 1.1 Fe ions interacting with cystein amino acid present in the Cytochrome

1.2.1.3 Rubredoxin

Rubredoxin is an electron-carrier found in sulfur-metabolizing bacteria and archaea. The active site contains an iron ion which is coordinated by the sulphur atoms of four cysteine residues forming an almost regular tetrahedron. Rubredoxin performs one-electron transfer processes. The oxidation state of the iron atom changes between the +2 and +3 states. In both oxidation states the metal is high spin, which helps to minimize structural changes.

1.2.1.4 Iron storage and transfer

Iron is stored as iron (III) in ferritin. The exact nature of the binding site has not yet been determined. The iron appears to be present as a hydrolysis product such as FeO (OH). Iron is transported by transferring whose binding site consists of two tyrosines, one aspartic acid and one histidine. The human body has no mechanism for iron excretion. This can lead to iron-overload problems in patients treated with blood transfusions, as, for instance, with β-thallasemia.

1.2.2 Metalloenzymes

Metalloenzymes have one feature in common, namely, that the metal ion is bound to the protein with one labile coordination site. As with all enzymes, the shape of the active site is crucial. The metal ion is usually located in a pocket whose shape fits the substrate.

1.2.2.1 Vitamin B12-dependent enzymes

Vitamin B12 catalyzes the transfer of methyl ($-CH_3$) groups between two molecules, which involves the breaking of C-C bonds, a process that is energetically expensive in organic reactions. The metal ion lowers the activation energy for the process by forming a transient $Co-CH_3$ bond. It consists of a cobalt (II) ion coordinated by four nitrogen atoms of a corrin rings and a fifth Nitrogen atom from an imidazole group. In the resting state there is a Co—C σ bond with the 5' carbon atom of adenosine. This is a naturally occurring organ metallic compound, which explains its function in trans-methylation reactions, such as the reaction carried out by methionine synthase.

1.2.2.2 Nitrogenase (nitrogen fixation)

The fixation of atmospheric nitrogen is a very energy-intensive process, as it involves breaking the very stable triple bond between the nitrogen atoms. The enzyme nitrogenase is one of the few enzymes that can catalyze the process. The enzyme occurs in certain bacteria. There are three components to its action: a molybdenum atom at the active site, Iron-sulfur clusters which are involved in transporting the electrons needed to reduce the nitrogen and an abundant energy source. The energy is provided by a symbiotic relationship between the bacteria and a host plant, often a legume. The relationship is symbiotic because the plant supplies the energy by photosynthesis and benefits by obtaining the fixed nitrogen. The reaction may be written symbolically as

$$\mathbf{N_2 + 16\ MgADP + 8e^- \rightarrow 2NH_3 + 16\ MgADP\ 16\ Pi + H_2}$$

where P_i stands for inorganic phosphate. The precise structure of the active site has been difficult to determine. It appears to contain a $MoFe_7S_8$ cluster which is able to bind the dinitrogen molecule and, presumably, enable the reduction process to begin. The electrons are transported by the associated "P" cluster, which contains two cubical Fe_4S_4 clusters joined by sulphur bridges.

1.2.2.3 Chlorophyll-containing Proteins

Chlorophyll plays a crucial role in photosynthesis. It contains magnesium enclosed in a chlorin ring (Figure 1.2). However, the magnesium ion is not directly involved in the photosynthetic function and can be replaced by other divalent ions with little loss of activity. Rather, the photon is absorbed by the chlorin ring, whose electronic structure is well-adapted for its purpose.

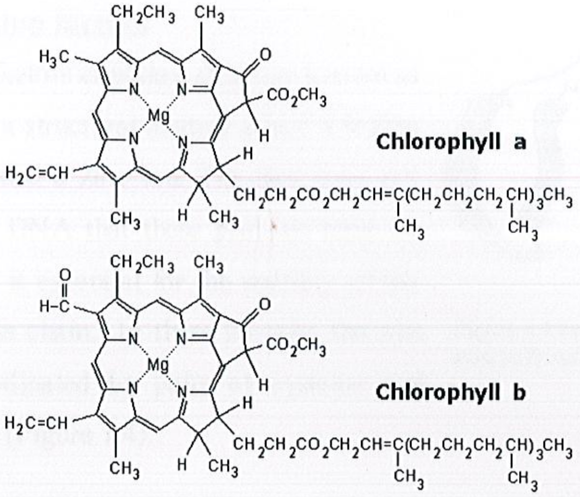


Figure 1.2 Mg ion interacting in the chlorin ring of chlorophyll molecule

1.2.3 Signal-transduction metalloproteins

1.2.3.1 Calmodulin

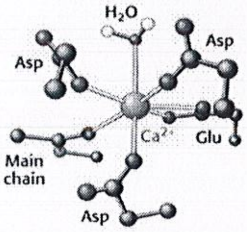


Figure 1.3 Ca ion is coordinated in a pentagonal bipyramidal configuration in the EF hand

Calmodulin is an example of a signal-transduction protein. It is a small protein which contains four EF-hand motifs, each of which can bind a Ca^{2+} ion. In an EF-hand loop the calcium ion is coordinated in a pentagonal bipyramidal configuration. Six Glutamic acid and Aspartic acid residues involved in the binding are in positions 1, 3, 5, 7, 9 of the polypeptide chain (Figure 1.3). At position 12 there is a glutamate or aspartate ligand which behaves as a (bidentate ligand), providing two oxygen atoms. The ninth residue in the loop is necessarily glycine due to the conformational requirements of the backbone. The coordination sphere of the calcium ion contains only carboxylate oxygen atoms and no nitrogen atoms. This is consistent with the hard nature of the calcium ion.

1.2.3.2 Transcription factors

Many transcription factors contain a structure known as a zinc finger; this is a structural module where a region of protein folds around a zinc ion. The zinc does not directly contact the DNA that these proteins bind to; instead the cofactor is essential for the stability of the tightly-folded protein chain. In these proteins the zinc ion is usually coordinated by pairs of cysteine and histidine side chains (Figure 1.4).

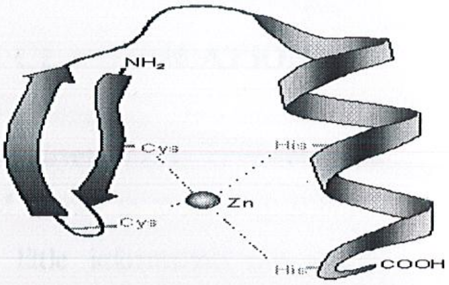


Figure 1.4 Zn ion interacting with histidine and cystein forming the Zn finger

CHAPTER 2

NEURAL NETWORKS FOR PROTEIN CLASSIFICATION

Molecular biology is a field that has experienced dramatic developments in recent years. A large number of data are constantly being generated thanks to several genomes – sequencing projects throughout the world. However, little information can readily extracted from these data and, therefore, data analysis has becomes a central issue in molecular biology. The analysis includes methods and algorithms for preprocessing visualization, knowledge discovery and data-mining of genomic and proteomic data. A vertiginous increase in the rate at which new protein structures are discovered has taken place as a by-product of ongoing sequencing projects. The functional annotation of membrane proteins in genomic sequences is an important problem in bioinformatics and computational biology.

2.1 What is an Artificial Neural Network?

An artificial neural network is a system based on the operation of biological neural networks, in other words, is an emulation of biological neural system. Why would be necessary the implementation of artificial neural networks? Although computing these days is truly advanced, there are certain tasks that a program made for a common microprocessor is unable to perform; even so a software implementation of a neural network can be made with their advantages and disadvantages.

Advantages:

- A neural network has the ability to learn in non-linear and random fashion and thus, can perform tasks that a linear regression, multiple linear regression or even nth order non-linear system cannot.
- When an element of the neural network fails, it can continue without much problem due to their parallel learning nature.
- A neural network learns by input/output mapping and thus, does not need to be reprogrammed.

- It can be implemented in applications where critical information e.g. sequence pattern information, for classification/prediction is not known and thus, user doesn't need to specify the same.
- It is an easy to use computational algorithm which, after learning once, can be implemented again and again.

Disadvantages:

- The neural network needs initial training/learning to operate, which is computationally complex and needs good resources.
- Biological data is usually diverse and imbalanced, having unequal number of data points in different classes. Neural Networks are not perfectly capable of learning from imbalanced dataset and usually over train for classes having higher number of data points.

Another aspect of the artificial neural networks is that there are different architectures, which consequently requires different types of algorithms, but despite to be an apparently complex system, a neural network is relatively simple.

The field of Artificial Neural Network is highly interdisciplinary, but our approach will restrict the view only to the research perspective. In research, neural networks serve two important functions: as pattern classifiers and as nonlinear adaptive filters. In terms of definition and style of computation, an Artificial Neural Network is an adaptive, most often nonlinear system that learns to perform a function (an input/output map) from data. Adaptive means that the system parameters are changed during operation, normally called the training phase. After the training phase the Artificial Neural Network parameters are fixed and the system is deployed to solve the problem at hand (the testing phase). The Artificial Neural Network is built with a systematic step-by-step procedure to optimize a performance criterion or to follow some implicit internal constraint, which is commonly referred to as the learning rule. The input/output training data are fundamental in neural network technology, because they convey the necessary information to "discover" the optimal operating point. The nonlinear nature of the neural network processing elements (PEs) provides the system with lots of flexibility to achieve practically any desired input/output map, i.e., some Artificial Neural Networks are universal mappers.

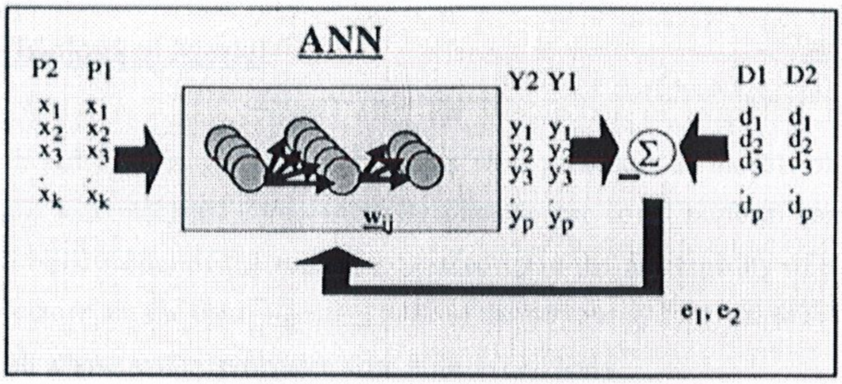


Figure 2.1 Architecture of Neural Network

An input is presented to the neural network and a corresponding desired or target response set at the output (when this is the case the training is called supervised). An error is composed from the difference between the desired response and the system output. This error information is fed back to the system and adjusts the system parameters in a systematic fashion (the learning rule). The process is repeated until the performance is acceptable (Figure 2.1). It is clear from this description that the performance hinges heavily on the data. If one does not have data that cover a significant portion of the operating conditions or if they are noisy, then neural network technology is probably not the right solution. On the other hand, if there is plenty of data and the problem is poorly understood to derive an approximate model, then neural network technology is a good choice. In artificial neural networks, the designer chooses the network topology, the performance function, the learning rule, and the criterion to stop the training phase, but the system automatically adjusts the parameters. So, it is difficult to bring a priori information into the design, and when the system does not work properly it is also hard to incrementally refine the solution. But in many difficult problems artificial neural networks provide performance that is difficult to match with other technologies. Denker 10 years ago said that "*artificial neural networks are the second best way to implement a solution*" motivated by the simplicity of their design and because of their universality, only shadowed by the traditional design obtained by studying the physics of the problem.

At present, artificial neural networks are emerging as the technology of choice for many applications, such as pattern recognition, prediction, system identification, and control.

2.2 The Biological Model

Artificial neural networks emerged after the introduction of simplified neurons by McCulloch and Pitts in 1943. These neurons were presented as models of biological neurons and as conceptual components for circuits that could perform computational tasks. The basic model of the neuron is founded upon the functionality of a biological neuron. Neurons are the basic signaling units of the nervous system and each neuron is a discrete cell whose several processes arise from its cell body.

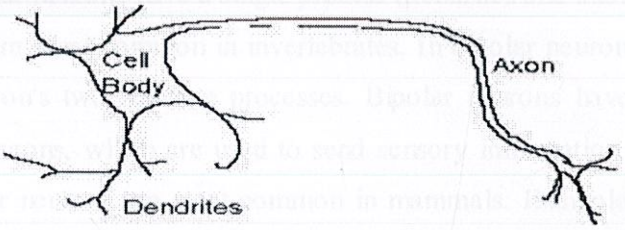


Figure 2.2 Biological neuron

The neuron has four main regions to its structure. The cell body, or soma, has two offshoots from it, the dendrites, and the axon, which end in presynaptic terminals (Figure 2.2). The cell body is the heart of the cell, containing the nucleus and maintaining protein synthesis. A neuron may have many dendrites, which branch out in a treelike structure, and receive signals from other neurons. A neuron usually only has one axon which grows out from a part of the cell body called the axon hillock. The axon conducts electric signals generated at the axon hillock down its length. These electric signals are called action potentials. The other end of the axon may split into several branches, which end in a presynaptic terminal. Action potentials are the electric signals that neurons use to convey information to the brain. All these signals are identical. Therefore, the brain determines what type of information is being received based on the path that the signal took. The brain analyzes the patterns of signals being sent and from that information it can interpret the type of information being received. Myelin is the fatty tissue that surrounds and insulates the axon. Often short axons do not need this insulation. There are uninsulated parts of the axon. These areas are called Nodes of Ranvier. At these nodes, the signal traveling down the axon is regenerated. This ensures that the signal traveling down the axon travels fast and remains constant (i.e. very short propagation delay and no weakening of the signal). The synapse is the area of contact between two neurons. The neurons do not actually physically touch. They are separated by the synaptic cleft, and electric signals are sent through thirteen chemical interactions. The neuron sending the signal is called the presynaptic cell and the neuron receiving the signal is called the postsynaptic cell. The signals are generated by the membrane potential, which is based on the differences in concentration of sodium and potassium ions inside and outside the cell membrane. Neurons can be classified by their number of processes (or appendages), or by their function. If they are classified by the number of processes, they fall into three categories. Unipolar neurons have a single process (dendrites and axon are located on the same stem), and are most common in invertebrates. In bipolar neurons, the dendrite and axon are the neuron's two separate processes. Bipolar neurons have a subclass called pseudo-bipolar neurons, which are used to send sensory information to the spinal cord. Finally, multipolar neurons are most common in mammals. Examples of these neurons are spinal motor neurons, pyramidal cells and Purkinje cells (in the cerebellum).

2.3 <u>The Mathematical Model</u>

When creating a functional model of the biological neuron, there are three basic components of importance. First, the synapses of the neuron are modeled as weights. The strength of the connection between an input and a neuron is noted by the value of the weight. Negative weight values reflect inhibitory connections, while positive values designate excitatory connections (Figure 2.3).

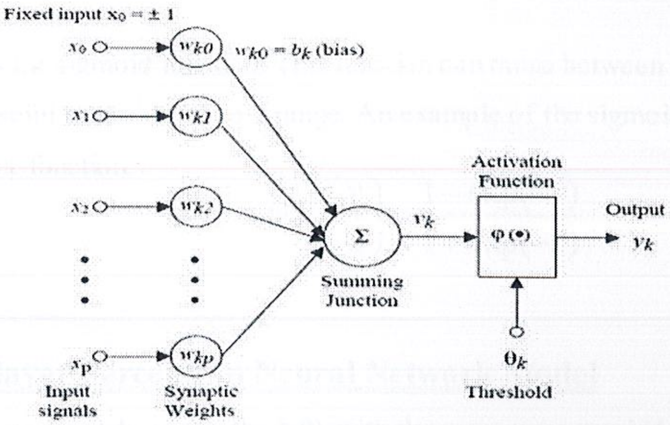


Figure 2.3 Mathematical description of Neural Network

The next two components model the actual activity within the neuron cell. An adder sums up all the inputs modified by their respective weights. This activity is referred to as linear combination. Finally, an activation function controls the amplitude of the output of the neuron. An acceptable range of output is usually between 0 and 1, or -1 and 1.

From this model the interval activity of the neuron can be shown to be:

$$v_k = \sum_{j=1}^{p} w_{kj} x_j$$

where w_{kj} represents weight and x_j represents the input coming from pth presynaptic node. The output of the neuron, v_k, would therefore be the outcome of some activation function on the value of v_k.

2.4 Activation functions

As mentioned previously, the activation function acts as a squashing function, such that the output of a neuron in a neural network is between certain values (usually 0 and 1, or -1 and 1). In general, there are three types of activation functions, denoted by Φ(.) . First, there is the Threshold Function which takes on a value of 0 if the summed input is less than a certain threshold value (v), and the value 1 if the summed input is greater than or equal to the threshold value.

$$\varphi(v) = \begin{cases} 1 & \text{if } v \geq 0 \\ 0 & \text{if } v < 0 \end{cases}$$

Secondly, there is the sigmoid function. This function can range between 0 and 1, but it is also sometimes useful to use the -1 to 1 range. An example of the sigmoid function is the hyperbolic tangent function.

$$\varphi(v) = \tanh\left(\frac{v}{2}\right) = \frac{1-\exp(-v)}{1+\exp(-v)}$$

2.5 The Multilayer Perceptron Neural Network Model

This network has an input layer (on the left) with three neurons, one hidden layer (in the middle) with three neurons and an output layer (on the right) with three neurons. There is one neuron in the input layer fur each predictor variable. In the case of categorical variables-*l* neurons are used to represent the *N* categories of the variable. The following diagram illustrates a perceptron network with three layers:

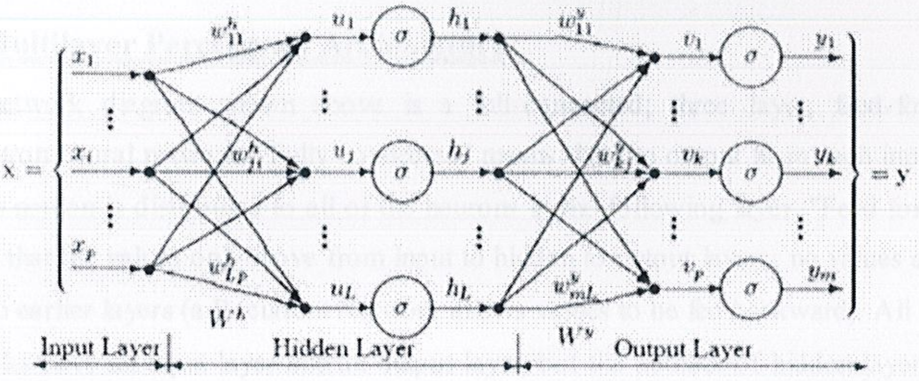


Figure 2.4 Multilayer perceptron neural network model

Input Layer - A vector of predictor variable values *(x, ...x$_p$)* is presented to the input layer. The input layer (or processing before the input layer) standardizes these values so that the range of each variable is -1 to 1. The input layer distributes the values to each of the neurons in the hidden layer. In addition to the predictor variables, there is a constant input of 1.0, called the *bias* that is fed to each of the hidden layers; the bias is multiplied by a weight and added to the sum going into the neuron.

Hidden Layer - The arriving at a neuron in the hidden layer, the value from each input neuron is multiplied by a weight *(W_{ji})*, and the resulting weighted values are added together producing a combined value u_i. The weighted sum *(u_j)* is fed into a transfer function, σ, which outputs a value h_j. The outputs from the hidden layer are distributed to the output layer.

Output Layer - Arriving at a neuron in the output layer, the value from each hidden layer neuron is multiplied by a weight *(W_{kj})*, and the resulting weighted values are added together producing a combined value *vi* The weighted sum *(v)* is fed into a transfer function, σ, which outputs a value Y_k. The *y* values are the outputs of the network.

If a regression analysis is being performed with a continuous target variable, then there is a single neuron in the output layer, and it generates a single *y* value.

2.6 Multilayer Perceptron Architecture

The network diagram shown above is a full-connected; three layer, feed-forward, perceptron neural network. "Fully connected" means that the output from each input and hidden neuron is distributed to all of the neurons in the following layer. "Feed forward" means that the values only move from input to hidden to output layers; no values are fed back to earlier layers (a Recurrent Network allows values to be fed backward). All neural networks have an input layer and an output layer, but the number of hidden layers may vary. When there is more than one hidden layer, the output from one hidden layer is fed into the next hidden layer and separate weights are applied to the sum going into each layer.

2.7 Training Multilayer Perceptron Networks

The goal of the training process is to find the set of weight values that will cause the output from the neural network to match the actual target values as closely as possible. There are several issues involved in designing and training a multi layer perceptron network:

- Selecting how many hidden layers to use in the network.

- Deciding how many neurons to use in each hidden layer.
- Finding a globally optimal solution that avoids local minima.
- Converging to an optimal solution in a reasonable period of time.
- Validating the neural network to test for over fitting.

2.8 Selecting the Number of Hidden Layers

For nearly all problems, one hidden layer is sufficient. Two hidden layers are required for modeling data with discontinuities such as a saw tooth wave pattern. Using two hidden layers rarely improves the model, and it may introduce a greater risk of converging to a local minima. There is no theoretical reason for using more than two hidden layers. SANN can build models with one or two hidden layers. Three layer models with one hidden layer are recommended.

2.9 Deciding how many neurons to use in the hidden layers

One of the most important characteristics of a perceptron network is the number of neurons in the hidden layer(s). If an inadequate number of neurons are used, the network will be unable to model complex data, and the resulting fit will be poor.

SANN includes an automated feature to find the optimal number of neurons in the hidden layer. We specify the minimum and maximum number of neurons we want it to test, and it will build models using varying numbers of neurons and measure the quality using either cross validation or hold-out data not used for training. This is a highly effective method for finding the optimal number of neurons, but it is computationally expensive, because many models must be built, and each model has to be validated.

The automated network search (ANS) for the optimal number of neurons only searches the first hidden layer.

2.10 Finding a globally optimal solution

A typical neural network might have a couple of hundred weighs whose values must be found to produce an optimal solution If neural networks were linear models like linear regression, it would be a breeze to find the optimal set of weights. But the output of a neural network as a function of the inputs is often highly nonlinear; this makes the optimization process complex.

2.11 Converging to the Optimal Solution - BFGS

Given a set of randomly-selected starting weight values, SANN Statistica uses the BFGS algorithm to optimize the weight values. Most training algorithms follow this cycle to refine the weight values: (1) run a set of predictor variable values through the network using a tentative set of weights, (2) compute the difference between the predicted target value and the actual target value for this case, (3) average the error information over the entire set of training cases, (4) propagate the error backward through the network and compute the gradient (vector of derivatives) of the change in error with respect to changes in weight values, (5) make adjustments to the weights to reduce the error. Each cycle is called an *epoch.*

OBJECTIVE:

With the explosion of protein sequences entering into databanks, it is highly desirable to explore the feasibility of selectively classifying newly found protein sequences into their respective metal ion binding classes by means of an automated method. This is indeed important because knowing which protein potentially binds to metal ion may help to deduce its catalytic mechanism and specificity, giving clues to the relevant biological function. With the availability of huge amount of genome sequencing data generated each day and for their functional annotation, sequence derived features are useful approaches.

Here in this study, an attempt has been taken for distinguishing protein sequences into metal ion binding and its classes using ANN for annotation of protein sequence with following objectives:

1. To extract sequence derived features and selection of important features from protein sequence to be used for prediction and classification of metal ion binding.
2. To develop and optimize the 1st layer for classifying the user input protein sequence into metal ion binding or non-metal ion binding based on sequence derived features.
3. To develop and optimize the 2nd layer for classifying the predicted metal ion binding protein sequences into three major classes based on sequence derived features.
4. To develop a 3rd layer for classifying the predicted class of metal ion binding protein sequences into their corresponding sub-classes and thus their specific binding metal ion.

Chapter 3

Materials and Methods

Overview of the work

In this study we have developed a cluster of neural networks consisting of three layers with usage of machine learning approach like ANN (Figure 3.1). This three layer classification system predicts the metal ion to which a protein sequence can potentially bind to. The sub-classes of Metal Ion binding class are constructed on the basis of chemical properties of the metal ions (Figure 3.1).

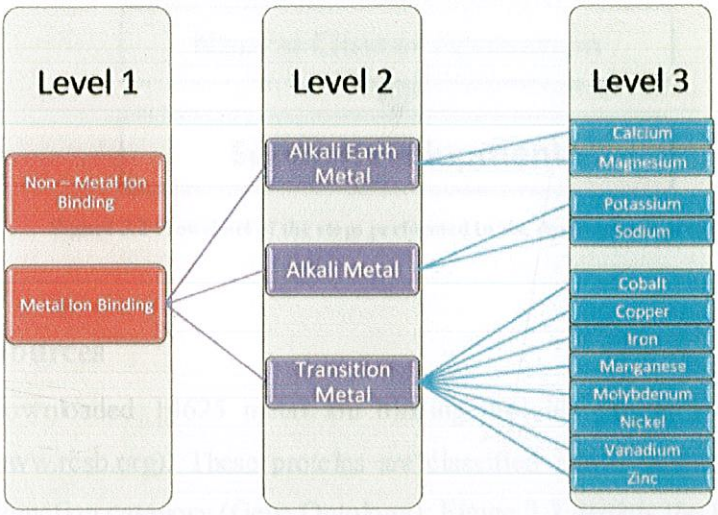


Figure 3.1 Neural Network consisting of three layers. The first layer classifies protein sequence as the metal ion binding and non-metal ion binding. The second layer shows the major classes and third layer shows the sub-classes.

The sequence derived features that were used are physicochemical properties, amino acid composition and pseudo amino acid composition. Using these parameters and their combination we have developed in total seven neural network clusters- $ANN_{pepstat}$, $ANN_{AA\ comp}$, ANN_{PseAA}, $ANN_{PseAA\ +\ pepstat}$, $ANN_{AA\ comp\ +\ pepstat}$, $ANN_{PseAA\ +\ AA\ comp}$ and $ANN_{PseAA\ +\ AA\ comp\ +\ pepstat}$. The overall protocol used in this study is described in Figure 3.2.

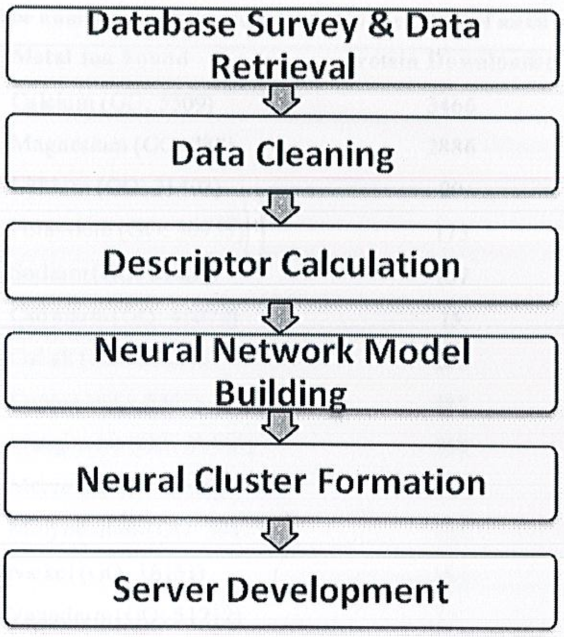


Figure 3.2 Flowchart of the steps performed in the development of the Metallopred

3.1 Data sources

We have downloaded 14625 metal ion binding proteins (Table 3.1) from the PDB database (www.rcsb.org). These proteins are classified into different classes based on Molecular Function category (Gene Ontology). Figure 3.3 depicts the hierarchy of Metal ion binding proteins in Gene Ontology.

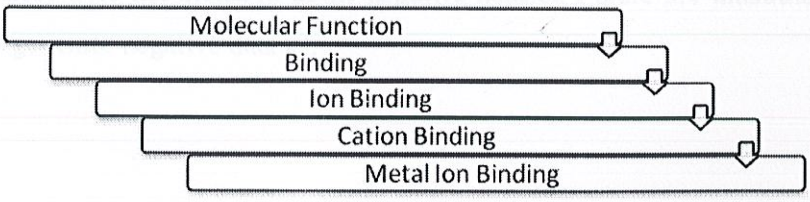


Figure 3.3 Steps followed in retrieving the data

Table 3.1 The number of proteins downloaded in each class of metal ion binding.

Metal Ion Found	Protein Downloaded
Calcium (GO: 5509)	3466
Magnesium (GO: 287)	2886
Lithium (GO: 31403)	20
Potassium (GO: 30955)	173
Sodium (GO: 31402)	157
Cadmium (GO: 46870)	15
Cobalt (GO: 50897)	200
Copper (GO: 5507)	887
Manganese (GO: 30145)	968
Mercury (GO: 45340)	11
Molybdenum (GO: 30151)	134
Nickel (GO: 16151)	147
Vanadium (GO: 51212)	11
Zinc (GO: 8270)	4861
Iron (GO: 5506)	328
TOTAL	**14264**

Similarly we have also taken a negative dataset (non-metal ion binding) consisting of 5738 proteins from the PDB database. We have checked the non metal binding proteins based on following two assumptions: 1. the proteins were not functionally annotated as metal binding based on gene ontology and 2. the PDB IDs which were used in the positive dataset were not included in the negative dataset. Figure 3.4 illustrates the steps followed to generate negative dataset.

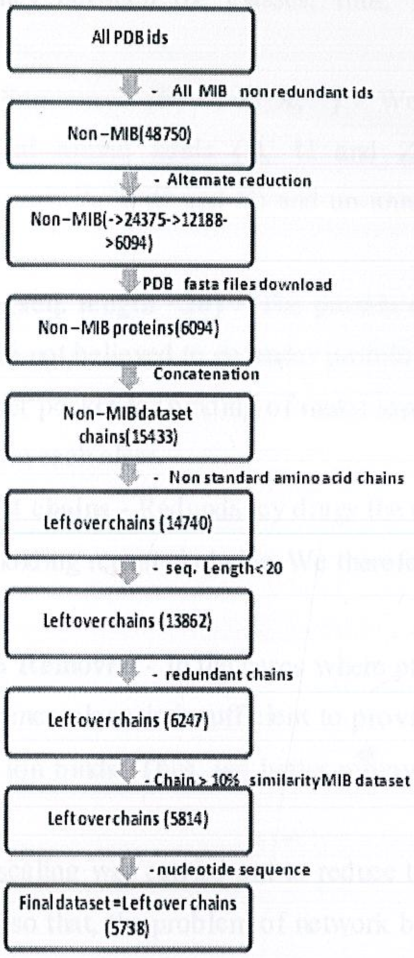


Figure 3.2 Flowchart followed for generating negative dataset

3.2 Data cleaning

Data cleaning is a methodology to remove unnecessary noise from the data. Thus, in order to generate a good training data, we performed the following steps for cleaning the data set. Table 3.2 represents the number of protein chains left after removal in each level of Data Cleaning.

I. **Calculated total no. of chains in each class** - We extracted the polypeptide chains of all proteins of each class.

II. **Removed the multi metal binding proteins** - We removed those proteins which were annotated to be binding to more than one metal ion. Such protein chains would pose redundancy between the classes, thus, would hamper the clear division of classes.

III. **Removed Incorrect Sequences (B, U, Z, X, *)** - We removed protein chains containing non-standard amino acids (B, U and Z), nucleotide sequences (sequences containing only A, T, G and C) and un-annotated amino acids (X or *).

IV. **Sequences Removed (seq. length <20)** - The protein chains having length less than 20 amino acids are not believed to go under protein folding process and thus, would not form a proper pocket for binding of metal ions. We therefore, removed such protein chains from each class.

V. **Removal of redundant chains** - Redundancy drags the network towards biasness for particular classes holding repeated chains. We therefore, removed such protein chains from each class.

VI. **Multi Chain Proteins Removed** - In the cases where proteins were having more than 1 chain, the sequence alone is insufficient to provide information about the chain to which metal ion binds. Thus, we better removed such proteins entirely from the dataset.

VII. **Data Scaling** – Data scaling was carried out to reduce the number of data points to a comparable level, so that, the problem of network biasing due to imbalanced number of data points in each class may be resolved. It was done using BLASTClust and random data point reduction (Table 3.3).

BLASTClust is a program within the standalone BLAST package used to cluster either protein or nucleotide sequences. The program begins with pairwise matches and places a sequence in the cluster where the sequence shows similarity more than the specified similarity parameter. In the case of proteins, the blastp algorithm is used to compute the pairwise matches. BLASTClust can take input in the form of concatenated FASTA-formatted sequences, each with a unique identifier in the comment line. BLASTClust formats the input sequence to produce a temporary BLAST database, performs clustering, and removes the database at completion. The output of BLASTClust is formatted as each cluster in a line having identifiers of the proteins in the cluster. Each new cluster is separated by a new line. The clusters are sorted on the basis of decreasing number of data points in the clusters. It accepts a number of parameters that can be used to control the stringency of clustering including thresholds for score density, percent identity, and alignment length. The BLASTClust program has a number of applications, the simplest of which is to create a non-redundant set of sequences from a source database.

Table 3.2 Number of protein chains left after removal in each level of Data Cleaning.

Metal Ion	Initial Proteins	IC	MMBR	RIS	LR	RCR	MCPR
Ca	3466	7332	5637	5568	5168	1662	1282
Mg	2886	7511	4659	4548	4198	1118	911
K	173	468	245	243	240	86	70
Na	157	441	37	37	25	13	13
Co	200	410	287	286	286	67	55
Cu	887	2270	1336	1334	1319	378	306
Mn	968	2426	1265	1253	1144	313	278
Mo	134	561	509	509	509	103	49
Ni	147	474	419	418	418	94	54
V	11	12	12	12	12	8	8
Zn	4861	13507	10674	10354	9542	3045	2187
Fe	328	1295	1290	1290	1286	174	158
TOTAL	**14218**	**36707**	**26370**	**25852**	**24147**	**7061**	**5371**

Calculated total number of chains in each class (**IC**); Removed the multi metal binding proteins (**MMBR**); Removed Incorrect Sequences (B, U, Z, X,*) (**RIS**); Sequences Removed (seq. length <20) (**LR**); Removal of redundant chains (**RCR**); Multi Chain Proteins Removed (**MCPR**).

The redundant proteins within each dataset were removed by using BLASTClust with a cutoff similarity of 30%. Therefore, each data set consists of proteins which are very diverse with similarity of < 30% hence covers all the features space corresponding to metal binding or not metal binding.

Table 3.3 Dataset Scaling: BLASTClust and Random data point reduction results.

Non metal binding (5738)		30% BLASTClust reduction		Random data point reduction			1035 (Layer 1)
Metal binding dataset							
Alkali earth	Ca(715)	1238 (Layer 3 Alkali Earth)	30% BLASTClust reduction	Random data point reduction			399 (Layer 2)
	Mg (523)						
Alkali	K (70)	83 (Layer 3 Alkali)	83 (Layer 2)				
	Na(13)						
Transition	Cu(306)	30% BLASTClust reduction	Random data point reduction	659 (Layer3 Transition)	30% BLASTClust reduction	Random data point reduction	381 (Layer 2)
	Zn(408)	30% BLASTClust reduction	Random data point reduction				
	Fe(82)						
	Mn (129)						
	Co(55)						
	Mo (49)						
	Ni(54)						
	V(8)						

3.3 Feature Extraction:

The following types of sequence derived features were used for the training and testing of the ANN models.

1. **Amino Acid composition:**

This feature consists 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:

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

2. **Physicochemical Properties:**

This feature consists of 12 properties calculated using EMBOSS (EBI) package. The parameters include Molecular weight, Charge, pI, Mole percentages of Tiny, Small, Aliphatic, Aromatic, Non-polar, Polar, Charged, Acidic and 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), Basic (H+K+R), 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) and Acidic (B+D+E+Z).

3. **Pseudo AA composition:**

It was introduced by Kuo-Chen Chou in 2001 to represent protein sequences for statistical prediction. This descriptor is a collection of 37 factors, 20 of which are weighted amino acid compositions and rest 17 are correlation factors calculated using sequence order among amino acids of the given sequences. The algorithm is explained as follows.

Given a protein sequence **P** with L amino acid resides, the sequence of the protein can be represented as

$$\mathrm{P} = \mathrm{R}_1\mathrm{R}_2\mathrm{R}_3\mathrm{R}_4\mathrm{R}_5\mathrm{R}_6\mathrm{R}_7\cdots\mathrm{R}_L \qquad (1)$$

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

$$\mathbf{P} = \begin{bmatrix} f_1 & f_2 & \cdots & f_{20} \end{bmatrix}^{\mathbf{T}} \qquad (2)$$

where f_u (u = 1, 2, ..., 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 PseAA composition model, the protein **P** of Eq.1 can be formulated as

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

where 20 + λ the components are given by

$$p_u = \begin{cases} \dfrac{f_u}{\sum_{i=1}^{20} f_i + w \sum_{k=1}^{\lambda} \tau_k}, & (1 \le u \le 20) \\[2ex] \dfrac{w \tau_{u-20}}{\sum_{i=1}^{20} f_i + w \sum_{k=1}^{\lambda} \tau_k}, & (20+1 \le u \le 20+\lambda) \end{cases} \qquad (4)$$

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

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

with

$$\mathrm{J}_{i,i+k} = \frac{1}{\Gamma} \sum_{g=1}^{\Gamma} \left[\Phi_\xi\left(\mathrm{R}_{i+k}\right) - \Phi_\xi\left(\mathrm{R}_i\right)\right]^2 \qquad (6)$$

where $\Phi_\xi(R_i)$ is the ξ-th function of the amino acid R_i, and Γ the total number of the functions considered. $\Phi_1(R_i)$, $\Phi_2(R_i)$ and $\Phi_3(R_i)$ are respectively the hydrophobicity value, hydrophilicity value, and side chain mass of amino acid R_i

(Table 3.4); while $\Phi_1(R_{i+k})$, $\Phi_2(R_{i+k})$ and $\Phi_3(R_{i+k})$ the corresponding values for the amino acid R_{i+k}. Therefore, the total number of functions considered is $\Gamma=3$. It can be seen from Eq.3 that the first 20 components, i.e. p_1, p_2, ..., p_{20} are associated with the conventional weight AA composition of protein, while the remaining components p_{20+1}, p_{20+2}, ..., $p_{20+\lambda}$ are the correlation factors that reflect the 1st tier, 2nd tier, …, and the λ-th tier sequence order correlation patterns. It is through these additional λ factors that some important sequence-order effects are incorporated.

Table 3.4 Scales used in PseAA: (a) hydrophobicity values from JACS, 1962, 84: 4240-4246. (C. Tanford), (b) hydrophilicity values from PNAS, 1981, 78:3824-3828 (T.P.Hopp & K.R.Woods) and (c) side-chain mass for each of the 20 amino acids.

Amino acid	Hydrophobicity	Hydrophilicity	Side chain mass
A	0.62	-0.5	15
C	0.29	-1	47
D	-0.9	3	59
E	-0.74	3	73
F	1.19	-2.5	91
G	0.48	0	1
H	-0.4	-0.5	82
I	1.38	-1.8	57
K	-1.5	3	73
L	1.06	-1.8	57
M	0.64	-1.3	75
N	-0.78	0.2	58
P	0.12	0	42
Q	-0.85	0.2	72
R	-2.53	3	101
S	-0.18	0.3	31
T	-0.05	-0.4	45
V	1.08	-1.5	43
W	0.81	-3.4	130
Y	0.26	-2.3	107

3.4 Binary classification of protein sequences

In this classification level, a protein sequence is classified as metal ion binding or non-metal ion binding. If it is metal ion binding, then it is further send to second layer of classification system. The parameters used for training and testing the model are amino acid composition, physicochemical properties, pseudo amino acid composition as well as fusion of these parameters. Figure 3.3 illustrates the Neural Network architecture of Layer 1.

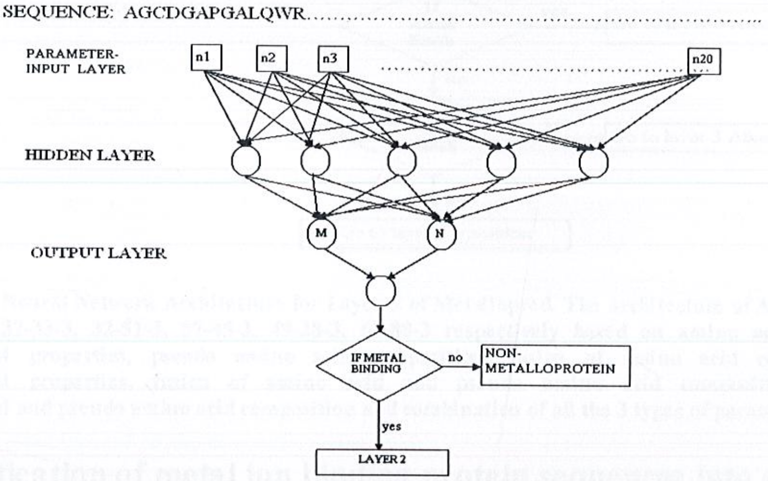


Figure 3.3 The Neural Network Architecture for Layer 1 of Metallopred. The architecture of ANN model is 20-42-2, 12-58-2, 37-41-2, 32-63-2, 57-65-2, 49-80-2, 69-78-2 respectively based on amino acid composition, physicochemical properties, pseudo amino acid composition, fusion of amino acid composition and physicochemical properties, fusion of amino acid and pseudo amino acid compositions, fusion of physicochemical and pseudo amino acid composition and combination of all the 3 types of parameters.

3.5 Classification of metal ion binding protein sequences into major classes

In this classification level the predicted metal ion binding proteins are classified into three major classes: Alkali Metal ion binding, Alkali Earth Metal ion binding or Transition Metal ion binding. After that each predicted metal ion binding class will further be sending to the third layer of classification system. Figure 3.4 illustrates the Neural Network architecture of Layer 2.

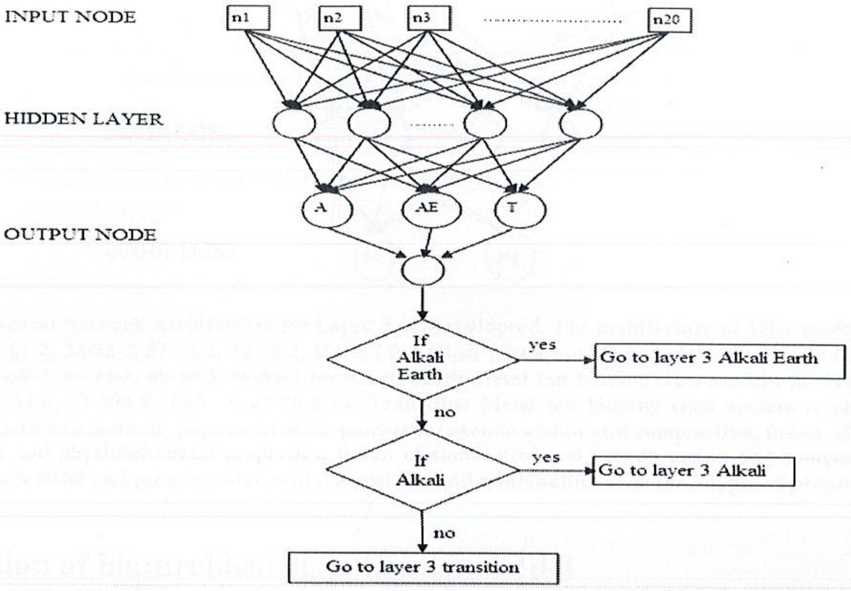


Figure 3.4 The Neural Network Architecture for Layer 2 of Metallopred. The architecture of ANN model is 20-55-3, 12-26-3, 37-33-3, 32-51-3, 57-45-3, 49-38-3, 69-88-3 respectively based on amino acid composition, physicochemical properties, pseudo amino acid composition, fusion of amino acid composition and physicochemical properties, fusion of amino acid and pseudo amino acid compositions, fusion of physicochemical and pseudo amino acid composition and combination of all the 3 types of parameters.

3.6 Classification of metal ion binding protein sequences into subclasses

In this classification level the predicted classes of metal ion binding proteins are further classified into specific metal ion binding proteins. The predicted Alkali Metal ion binding proteins are further classified as Potassium or Sodium Metal ion binding. The Alkali Earth Metal ion binding proteins are classified into Calcium or Magnesium Metal ion binding and the Transition Metal ion binding proteins are classified into 8 sub classes which are Vanadium, Nickel, Molybdenum, Cobalt, Manganese, Iron, Zinc and Copper Metal ion binding. Figure 3.5 illustrates the architecture of the Neural Network of Layer 3.

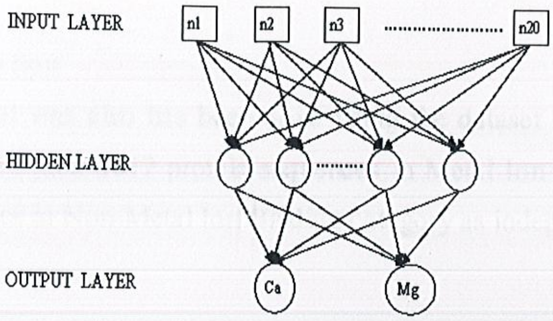


Figure 3.5 The Neural Network Architecture for Layer 3 of Metallopred. The architecture of ANN model is 20-45-2, 12-21-2, 37-61-2, 32-52-2, 57-52-2, 49-30-2, 69-53-2 for Alkali Metal Ion binding class models; 20-80-2, 12-31-2, 37-32-2, 32-68-2, 57-43-2, 49-35-2, 69-85-2 for Alkali Earth Metal Ion binding class models; 20-122-8, 12-37-8, 37-53-8, 32-53-8, 57-100-8, 49-85-8, 69-70-8 for Transition Metal Ion binding class models respectively based on amino acid composition, physicochemical properties, pseudo amino acid composition, fusion of amino acid composition and physicochemical properties, fusion of amino acid and pseudo amino acid compositions, fusion of physicochemical and pseudo amino acid composition and combination of all the 3 types of parameters.

3.7 Validation of hierarchical classification model

The validation is the way to confirm the validity of data, information, processes or a model. We have used three different approaches to validate our tool as follows.

3.7.1 Validation based on self consistency

The performance of our online tool MetalloPred was validated using self consistency method. In this approach the data set of metal binding proteins were given as input to the tool and the predicted result was observed. The predicted accuracy at each level of classification was calculated based on the predicted output. We took 2354 protein sequences in Metal Ion Binding category and 1035 protein sequences in Non-Metal Ion Binding category.

3.7.2. Validation based on Jack-knifing

The dataset was also subject to the jackknife test that is deemed to be one of the most rigorous and objective methods for cross-validation in statistics. We took 2354 protein sequences in Metal Ion Binding category and 1035 protein sequences in Non-Metal Ion Binding category. Both the data set were merged together and classified into two category: metal ion binding (2345 sequences) and non-metal binding (1035) by taking sequences randomly. This randomization process was repeated for 100 times and the average performance accuracy was measured.

3.7.3 External validation

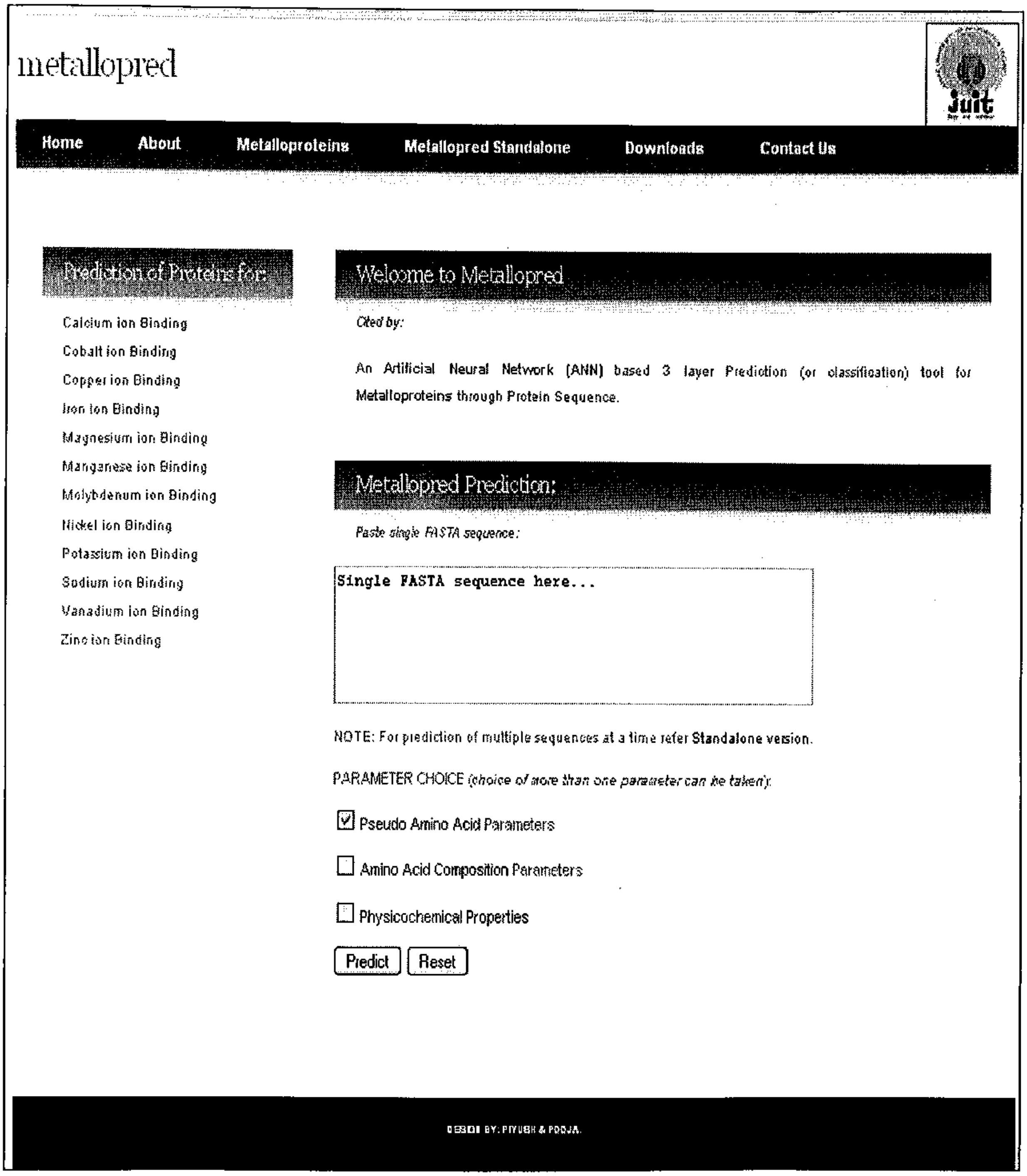
Validation of our tool was also has been done using the dataset that was not used for training or testing. We took 3017 protein sequences in Metal Ion Binding category and 3322 protein sequences in Non-Metal Ion Binding category as independent data set.

3.8 Standalone and Server development of MetalloPred

A standalone as well as online version of our tool (MetalloPred) has been developed and uploaded into our University web server. The following steps have been used for the development of the server:

- Deployed and modified the C codes for each Neural Network model generated.
- Converted the C codes to C library references as Header files.
- Generated a main parser code, which can take in the descriptors from a file and can send them to the particular network models in their corresponding header files and retrieves the output of the model. Based on output, it takes the decision to go which way in the hierarchy or to which particular model to feed the descriptor and retrieve the output. This step is repeated until a case gets predicted to the terminal node in the hierarchy i.e. reaching to a particular ion to which it can potentially bind to if it is predicted as a Metal Ion binding protein.
- Generated Perl parsers to link the webpage with the prediction cluster codes, in case of online tool and to link prediction clusters directly in case of standalone. These perl codes can retrieve the sequence and the choice of sequence based feature from the user and then can convert the protein sequence to features and present them to prediction clusters for prediction.

Figure 3.6 and figure 3.7 illustrates the front end of the online version with sequence to be submitted as query and options for the sequence derived features to be used and its output for an example sequence.



metallopred

Prediction of Proteins for:

Calcium ion Binding
Cobalt ion Binding
Copper ion Binding
Iron ion Binding
Magnesium ion Binding
Manganese ion Binding
Molybdenum ion Binding
Nickel ion Binding
Potassium ion Binding
Sodium ion Binding
Vanadium ion Binding
Zinc ion Binding

Welcome to Metallopred

Cited by:

An Artificial Neural Network (ANN) based 3 layer Prediction (or classification) tool for Metalloproteins through Protein Sequence.

Metallopred Prediction:

Paste single FASTA sequence:

```
Single FASTA sequence here...
```

NOTE: For prediction of multiple sequences at a time refer Standalone version.

PARAMETER CHOICE *(choice of more than one parameter can be taken)*:

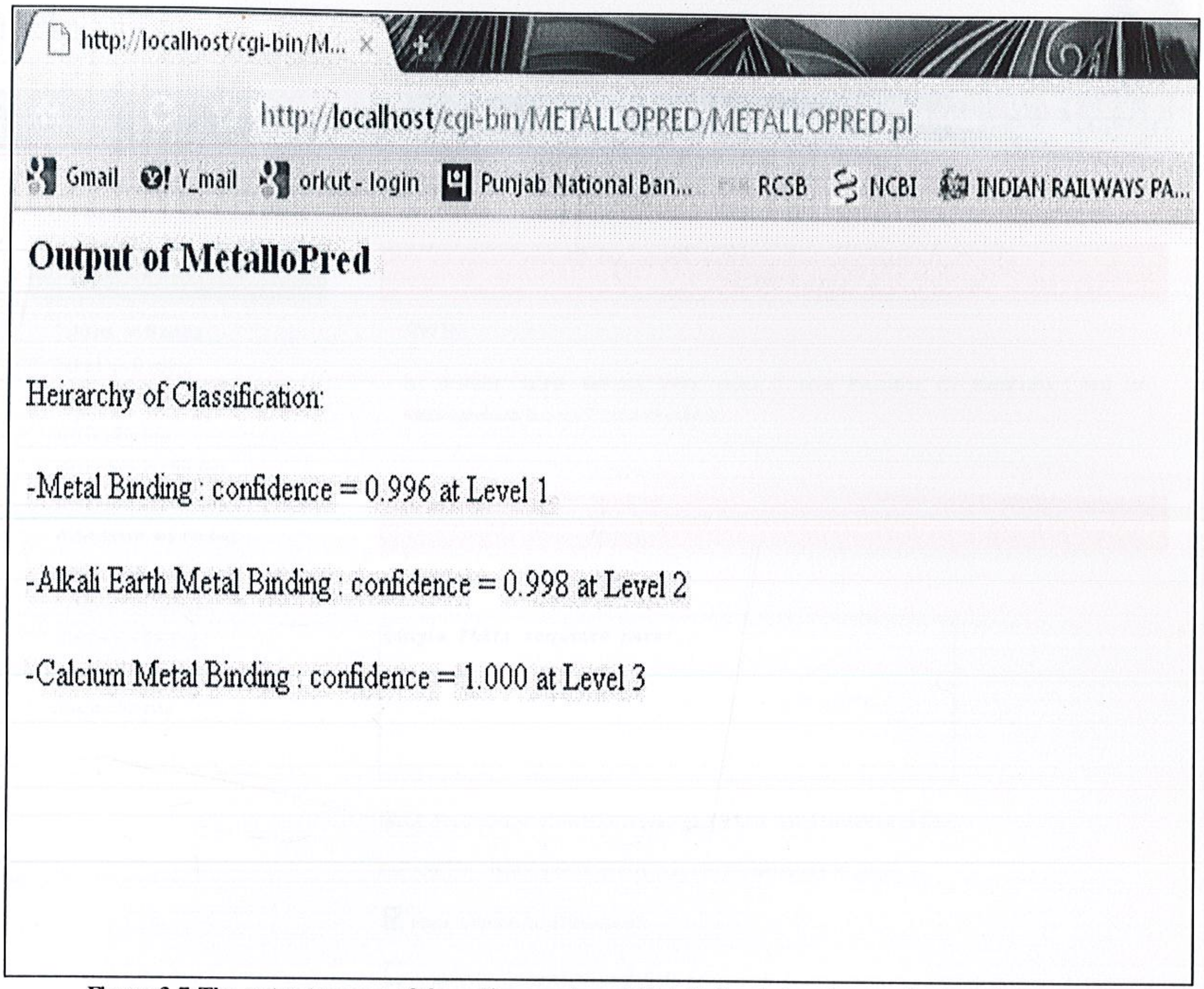
☑ Pseudo Amino Acid Parameters

□ Amino Acid Composition Parameters

□ Physicochemical Properties

DESIGN BY: PIYUSH & POOJA.

Figure 3.6 The frontend of the online version of MetalloPred.


http://localhost/cgi-bin/M...
http://localhost/cgi-bin/METALLOPRED/METALLOPRED.pl
Gmail
Y_mail
orkut - login
Punjab National Ban...
RCSB
NCBI
INDIAN RAILWAYS PA...

Output of MetalloPred

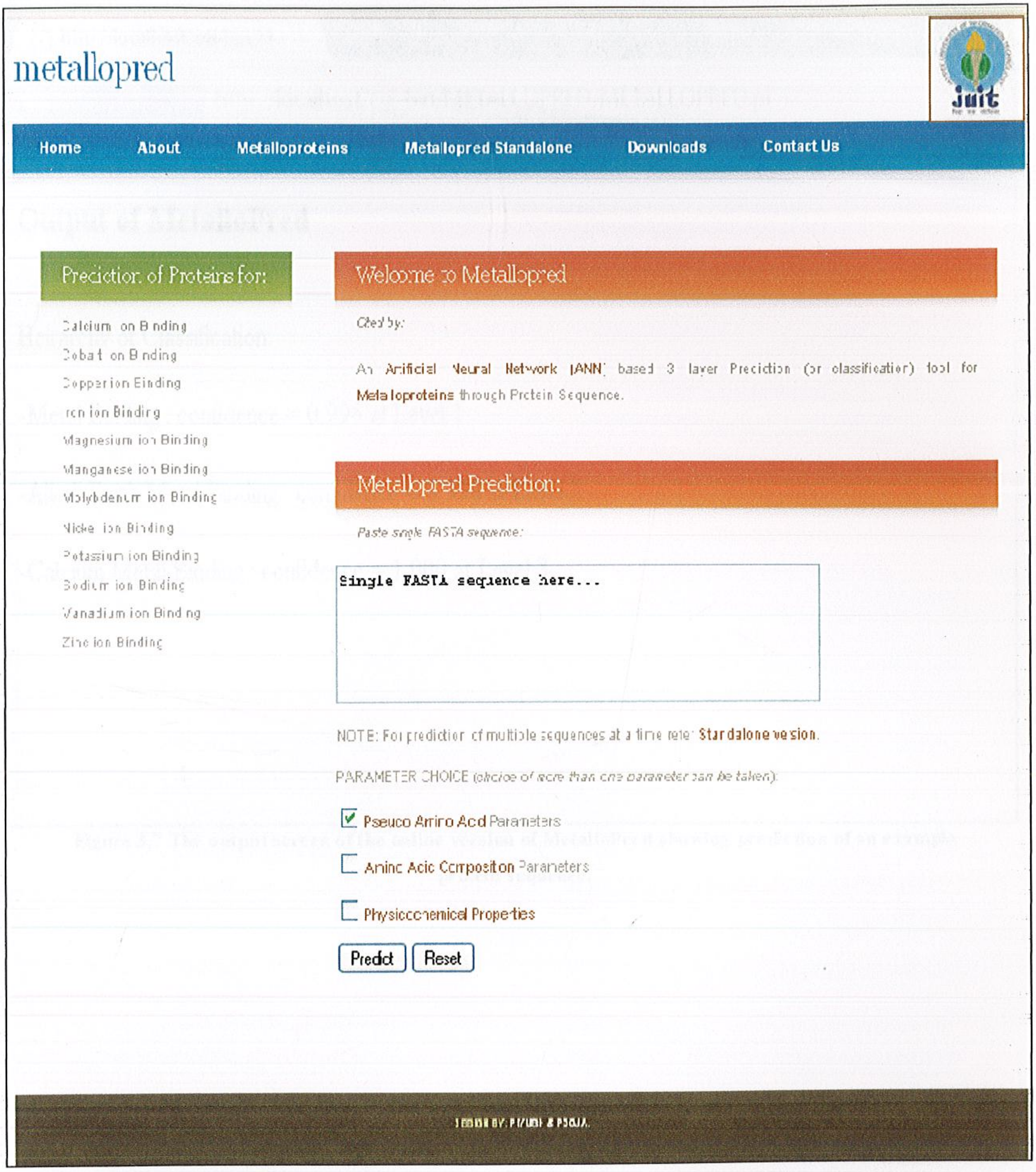
Heirarchy of Classification:

-Metal Binding : confidence = 0.996 at Level 1

-Alkali Earth Metal Binding : confidence = 0.998 at Level 2

-Calcium Metal Binding : confidence = 1.000 at Level 3

Figure 3.7 The output screen of the online version of MetalloPred showing prediction of an example protein sequence.



Prediction of Proteins for:

Calcium ion Binding
Cobalt ion Binding
Copper ion Binding
Iron ion Binding
Magnesium ion Binding
Manganese ion Binding
Molybdenum ion Binding
Nickel ion Binding
Potassium ion Binding
Sodium ion Binding
Vanadium ion Binding
Zinc ion Binding

Welcome to Metallopred

Cited by:

An Artificial Neural Network (ANN) based 3 layer Prediction (or classification) tool for Metalloproteins through Protein Sequence.

Metallopred Prediction:

Paste single FASTA sequence:

Single FASTA sequence here...

NOTE: For prediction of multiple sequences at a time refer Standalone version.

PARAMETER CHOICE *(choice of more than one parameter can be taken)*:

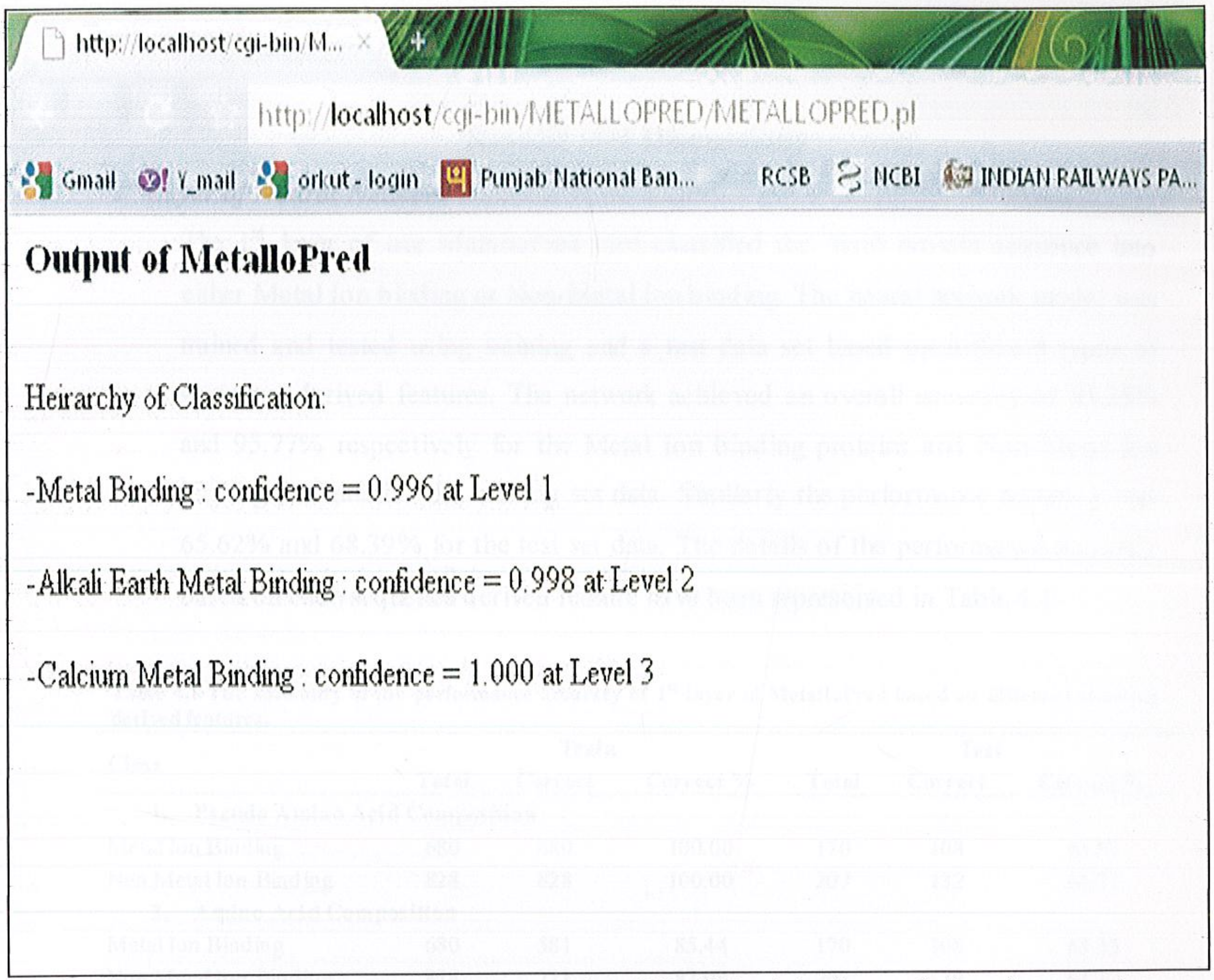
☑ Pseudo Amino Acid Parameters

□ Amino Acid Composition Parameters

□ Physicochemical Properties

DESIGN BY: PIYUSH & POOJA.

Figure 3.6 The frontend of the online version of MetalloPred.


http://localhost/cgi-bin/M...
http://localhost/cgi-bin/METALLOPRED/METALLOPRED.pl
Gmail
Y_mail
orkut - login
Punjab National Ban...
RCSB
NCBI
INDIAN RAILWAYS PA...

Output of MetalloPred

Heirarchy of Classification:

-Metal Binding : confidence = 0.996 at Level 1

-Alkali Earth Metal Binding : confidence = 0.998 at Level 2

-Calcium Metal Binding : confidence = 1.000 at Level 3

Figure 3.7 The output screen of the online version of MetalloPred showing prediction of an example protein sequence.

CHAPTER 4

Results and Discussions

1st Layer of Neural Network

The 1st layer of our MetalloPred tool classified the input protein sequence into either Metal Ion binding or Non-Metal Ion binding. The neural network model was trained and tested using training and a test data set based on different types of sequence derived features. The network achieved an overall accuracy of 95.25% and 95.77% respectively for the Metal Ion binding proteins and Non-Metal Ion binding proteins for the training set data. Similarly the performance accuracy was 65.62% and 68.39% for the test set data. The details of the performance accuracy based on each sequence derived feature have been represented in Table 4.1.

Table 4.1 The summary of the performance accuracy of 1st layer of MetalloPred based on different sequence derived features.

Class	Train			Test		
	Total	Correct	Correct %	Total	Correct	Correct %
1. Pseudo Amino Acid Composition						
Metal Ion Binding	680	680	100.00	170	108	63.53
Non Metal Ion Binding	828	828	100.00	207	132	63.77
2. Amino Acid Composition						
Metal Ion Binding	680	581	85.44	170	108	63.53
Non Metal Ion Binding	828	721	87.08	207	129	62.32
3. Physicochemical Properties						
Metal Ion Binding	680	574	84.41	170	100	58.82
Non Metal Ion Binding	828	732	88.41	207	143	69.08
4. Pseudo Amino Acid Composition + Amino Acid Composition						
Metal Ion Binding	680	671	98.68	170	117	68.82
Non Metal Ion Binding	828	814	98.31	207	155	74.88
5. Pseudo Amino Acid Composition + Physicochemical Properties						
Metal Ion Binding	680	678	99.71	170	100	58.82
Non Metal Ion Binding	828	826	99.76	207	134	64.73
6. Amino Acid Composition + Physicochemical Properties						
Metal Ion Binding	680	671	98.68	170	113	66.47
Non Metal Ion Binding	828	805	97.22	207	125	60.39
7. Pseudo Amino Acid Composition + Amino Acid Composition + Physicochemical Properties						
Metal Ion Binding	680	679	99.85	170	135	79.41
Non Metal Ion Binding	828	825	99.64	207	173	83.57
Average						
Metal Ion Binding			95.25			65.62
Non Metal Ion Binding			95.77			68.39

The performance accuracy was further validated using self consistency test and jackknife test. The overall accuracy of the 1st layer of MetalloPred is 62.41% and 47.05% for the Metal Ion and Non-Metal Ion binding classes based on self consistency test. Similarly, using jackknife test, the accuracy was found to be 50.72% and 37.03% for the Metal Ion and Non-Metal Ion binding classes (Table 4.2). Moreover, the results were robust and hence, the MetalloPred could successfully predict the novel protein sequence into either of Metal Ion binding or Non-Metal Ion binding class as evident form the independent data set used for validation (Table 4.2).

Table 4.2 The performance accuracy of the 1st layer of MetalloPred based on validation techniques (self consistency test, jackknife test and independent set validation).

Class	Total	Self Consistency	Jackknife	Independent Set	
				Total	Correct %
1. Pseudo Amino Acid Composition					
Metal Ion Binding	2354	70.14	57.01	3017	50.98
Non Metal Ion Binding	1035	42.71	33.24	3322	33.93
2. Amino Acid Composition					
Metal Ion Binding	2354	68.86	54.04	3017	54.09
Non Metal Ion Binding	1035	45.51	35.46	3322	35.94
3. Physicochemical Properties					
Metal Ion Binding	2354	57.22	45.16	3017	33.34
Non Metal Ion Binding	1035	37.87	28.70	3322	29.77
4. Pseudo Amino Acid Composition + Amino Acid Composition					
Metal Ion Binding	2354	53.91	41.67	3017	44.75
Non Metal Ion Binding	1035	37.97	25.99	3322	29.95
5. Pseudo Amino Acid Composition + Physicochemical Properties					
Metal Ion Binding	2354	59.35	50.89	3017	39.58
Non Metal Ion Binding	1035	92.66	82.03	3322	61.05
6. Amino Acid Composition + Physicochemical Properties					
Metal Ion Binding	2354	68.73	56.37	3017	51.91
Non Metal Ion Binding	1035	40.39	29.37	3322	32.60
7. Pseudo Amino Acid Composition + Amino Acid Composition + Physicochemical Properties					
Metal Ion Binding	2354	58.71	49.92	3017	51.38
Non Metal Ion Binding	1035	32.27	24.44	3322	25.47
Average					
Metal Ion Binding		62.41	50.72		46.57
Non Metal Ion Binding		47.05	37.03		35.53

By comparing the performance accuracy of the 1st layer of MetalloPred between the individual sequence derived features; it has been observed that the accuracy was better by combining pseudo amino acid and physiochemical properties.

2nd Layer of Neural Network

The 2nd layer of our MetalloPred tool classified the input protein sequence into either Alkali Metal Ion binding, Alkali Earth Metal Ion binding or Transition Metal Ion binding. The neural network model was trained and tested using training and a test data set based on different types of sequence derived features. The network achieved an overall accuracy of 95.40%, 96.55% and 96.02% respectively for the Alkali Metal Ion binding, Alkali Earth Metal Ion binding and Transition Metal Ion binding proteins for the training set data. Similarly the performance accuracy was 70.40%, 63.03% and 68.42% for the test set data. The details of the performance accuracy based on each sequence derived feature have been represented in Table 4.3.

Table 4.3 The summary of the performance accuracy of 2nd layer of MetalloPred based on different sequence derived features.

Class	Train			Test		
	Total	Correct	Correct %	Total	Correct	Correct %
1. Pseudo Amino Acid Composition						
Alkali Earth Metal	319	298	93.42	80	48	60.00
Alkali Metal	56	55	98.21	14	12	85.71
Transition Metal	305	292	95.74	76	54	71.05
2. Amino Acid Composition						
Alkali Earth Metal	319	316	99.06	80	52	65.00
Alkali Metal	56	56	100.00	14	8	57.14
Transition Metal	305	301	98.69	76	52	68.42
3. Physicochemical Properties						
Alkali Earth Metal	319	297	93.10	80	57	71.25
Alkali Metal	56	47	83.93	14	9	64.29
Transition Metal	305	272	89.18	76	45	59.21
4. Pseudo Amino Acid Composition + Amino Acid Composition						
Alkali Earth Metal	319	313	98.12	80	42	52.50
Alkali Metal	56	54	96.43	14	9	64.29
Transition Metal	305	297	97.38	76	48	63.16
5. Pseudo Amino Acid Composition + Physicochemical Properties						
Alkali Earth Metal	319	316	99.06	80	57	71.25
Alkali Metal	56	56	100.00	14	12	85.71
Transition Metal	305	301	98.69	76	51	67.11
6. Amino Acid Composition + Physicochemical Properties						

Alkali Earth Metal	319	300	94.04	80	55	68.75
Alkali Metal	56	50	89.29	14	10	71.43
Transition Metal	305	286	93.77	76	54	71.05
7. Pseudo Amino Acid Composition + Amino Acid Composition + Physicochemical Properties						
Alkali Earth Metal	319	316	99.06	80	42	52.50
Alkali Metal	56	56	100.00	14	9	64.29
Transition Metal	305	301	98.69	76	60	78.95
Average						
Alkali Earth Metal			96.55			63.03
Alkali Metal			95.40			70.40
Transition Metal			96.02			68.42

The performance accuracy was further validated using self consistency test and jackknife test. The overall accuracy of the 2nd layer of MetalloPred is 76.93%, 67.20% and 56.20% respectively for the Alkali Metal Ion binding, Alkali Earth Metal Ion binding and Transition Metal Ion binding classes based on self consistency test. Similarly, using jackknife test, the accuracy was found to be 70.70%, 57.77% and 47.49% respectively for the Alkali Metal Ion binding, Alkali Earth Metal Ion binding and Transition Metal Ion binding classes (Table 4.4). Moreover, the results were robust and hence, the MetalloPred could successfully predict the novel protein sequence into either of Alkali Metal Ion binding, Alkali Earth Metal Ion binding or Transition Metal Ion binding class as evident form the independent data set used for validation (Table 4.4).

Table 4.4 The performance accuracy of the 2nd layer of MetalloPred based on validation techniques (self consistency test, jackknife test and independent set validation).

Class	Total	Self Consistency	Jackknife	Independent Set	
				Total	Correct %
1. Pseudo Amino Acid Composition					
Alkali Earth Metal	1238	68.26	59.13	955	64.19
Alkali Metal	70	92.86	84.29	13	76.92
Transition Metal	1046	71.32	62.24	2049	48.56
2. Amino Acid Composition					
Alkali Earth Metal	1238	67.77	59.21	955	61.15
Alkali Metal	70	90.00	78.57	13	76.92
Transition Metal	1046	68.36	60.33	2049	45.00
3. Physicochemical Properties					
Alkali Earth Metal	1238	66.40	49.11	955	63.04
Alkali Metal	70	72.86	61.43	13	69.23
Transition Metal	1046	43.40	31.74	2049	28.79
4. Pseudo Amino Acid Composition + Amino Acid Composition					
Alkali Earth Metal	1238	75.69	59.94	955	72.67

Alkali Metal	70	45.71	32.86	13	38.46
Transition Metal	1046	34.23	24.19	2049	30.26
5. Pseudo Amino Acid Composition + Physicochemical Properties					
Alkali Earth Metal	1238	59.05	51.78	955	56.34
Alkali Metal	70	97.14	90.00	13	84.62
Transition Metal	1046	56.41	49.62	2049	41.24
6. Amino Acid Composition + Physicochemical Properties					
Alkali Earth Metal	1238	68.42	57.92	955	67.33
Alkali Metal	70	87.14	78.57	13	69.23
Transition Metal	1046	67.69	59.75	2049	44.61
7. Pseudo Amino Acid Composition + Amino Acid Composition + Physicochemical Properties					
Alkali Earth Metal	1238	64.86	49.76	955	63.66
Alkali Metal	70	52.86	42.86	13	46.15
Transition Metal	1046	52.01	40.06	2049	45.68
Average					
Alkali Earth Metal		67.20	57.77		61.54
Alkali Metal		76.93	70.70		62.16
Transition Metal		56.20	47.49		39.94

By comparing the performance accuracy of the 2nd layer of MetalloPred between the individual sequence derived features; it has been observed that the accuracy was better by pseudo amino acid as well as by amino acid composition and by combining pseudo amino acid and physiochemical properties.

3rd Layer of Neural Network for Alkali Earth Metal Ion binding class

The 3rd layer of our MetalloPred tool developed for Alkali Earth Metal Ion binding class classified the input protein sequence to be either Calcium Metal Ion binding or Magnesium Metal Ion binding. The neural network model was trained and tested using training and a test data set based on different types of sequence derived features. The network achieved an overall accuracy of 93.88% and 92.10% respectively for the Calcium Metal Ion binding and Magnesium Metal Ion binding proteins for the training set data. Similarly the performance accuracy was 88.51% and 87.75% for the test set data. The details of the performance accuracy based on each sequence derived feature have been represented in Table 4.5.

Table 4.5 The summary of the performance accuracy of 3rd layer of MetalloPred developed for Alkali Earth Metal Ion binding class based on different sequence derived features.

Class	Train			Test		
	Total	Correct	Correct %	Total	Correct	Correct %
1. Pseudo Amino Acid Composition						
Calcium	572	549	95.98	143	123	86.01
Magnesium	418	396	94.74	105	82	78.10
2. Amino Acid Composition						
Calcium	572	529	92.48	143	130	90.91
Magnesium	418	378	90.43	105	97	92.38
3. Physicochemical Properties						
Calcium	572	512	89.51	143	124	86.71
Magnesium	418	366	87.56	105	91	86.67
4. Pseudo Amino Acid Composition + Amino Acid Composition						
Calcium	572	542	94.76	143	129	90.21
Magnesium	418	391	93.54	105	98	93.33
5. Pseudo Amino Acid Composition + Physicochemical Properties						
Calcium	572	556	97.20	143	126	88.11
Magnesium	418	398	95.22	105	83	79.05
6. Amino Acid Composition + Physicochemical Properties						
Calcium	572	539	94.23	143	127	88.81
Magnesium	418	388	92.82	105	96	91.43
7. Pseudo Amino Acid Composition + Amino Acid Composition + Physicochemical Properties						
Calcium	572	532	93.01	143	127	88.81
Magnesium	418	378	90.43	105	98	93.33
Average						
Calcium			93.88			88.51
Magnesium			92.10			87.75

The performance accuracy was further validated using self consistency test and jackknife test. The overall accuracy of the 3rd layer of MetalloPred for Alkali Earth Metal Ion binding class is 71.55% and 67.93% respectively for the Calcium Metal Ion binding and Magnesium Metal Ion binding classes based on self consistency test. Similarly, using jackknife test, the accuracy was found to be 60.20% and 57.00% respectively for the Calcium Metal Ion binding and Magnesium Metal Ion binding classes (Table 4.6). Moreover, the results were robust and hence, the MetalloPred could successfully predict the novel protein sequence into either of Calcium Metal Ion binding or Magnesium Metal Ion binding class as evident form the independent data set used for validation (Table 4.6).

Table 4.6 The performance accuracy of the 3rd layer of MetalloPred developed for Alkali Earth Metal Ion binding class based on validation techniques (self consistency test, jackknife test and independent set validation).

Class	Total	Self Consistency	Jackknife	Independent Set	
				Total	Correct %
1. Pseudo Amino Acid Composition					
Calcium	715	72.59	63.92	567	63.67
Magnesium	523	66.16	57.55	388	64.95
2. Amino Acid Composition					
Calcium	715	73.29	63.36	567	72.84
Magnesium	523	64.05	54.49	388	63.66
3. Physicochemical Properties					
Calcium	715	69.09	52.59	567	66.84
Magnesium	523	66.73	49.33	388	62.11
4. Pseudo Amino Acid Composition + Amino Acid Composition					
Calcium	715	81.12	64.76	567	74.25
Magnesium	523	77.82	62.91	388	75.52
5. Pseudo Amino Acid Composition + Physicochemical Properties					
Calcium	715	62.52	56.22	567	55.56
Magnesium	523	72.08	64.63	388	57.47
6. Amino Acid Composition + Physicochemical Properties					
Calcium	715	73.71	63.08	567	72.49
Magnesium	523	65.01	56.60	388	57.22
7. Pseudo Amino Acid Composition + Amino Acid Composition + Physicochemical Properties					
Calcium	715	68.53	57.48	567	63.32
Magnesium	523	63.67	53.54	388	61.60
Average					
Calcium		71.55	60.20		66.99
Magnesium		67.93	57.00		63.21

By comparing the performance accuracy of the 3rd layer of MetalloPred for Alkali Earth Metal Ion binding class between the individual sequence derived features; it has been observed that the accuracy was better by combining pseudo amino acid and amino acid composition.

3rd Layer of Neural Network for Alkali Metal Ion binding class

The 3rd layer of our MetalloPred tool developed for Alkali Metal Ion binding class classified the input protein sequence to be either Potassium Metal Ion binding or Sodium Metal Ion binding. The neural network model was trained and tested using training and a test data set based on different types of sequence derived features.

The network achieved an overall accuracy of 99.40% and 91.07% respectively for the Potassium Metal Ion binding and Sodium Metal Ion binding proteins for the training set data. Similarly the performance accuracy was 97.62% and 57.14% for the test set data. The details of the performance accuracy based on each sequence derived feature have been represented in Table 4.7.

Table 4.7 The summary of the performance accuracy of 3rd layer of MetalloPred developed for Alkali Metal Ion binding class based on different sequence derived features.

Class	Train			Test		
	Total	Correct	Correct %	Total	Correct	Correct %
1. Pseudo Amino Acid Composition						
Potassium	48	48	100.00	12	12	100.00
Sodium	8	7	87.50	2	1	50.00
2. Amino Acid Composition						
Potassium	48	48	100.00	12	12	100.00
Sodium	8	7	87.50	2	1	50.00
3. Physicochemical Properties						
Potassium	48	47	97.92	12	11	91.67
Sodium	8	8	100.00	2	1	50.00
4. Pseudo Amino Acid Composition + Amino Acid Composition						
Potassium	48	48	100.00	12	11	91.67
Sodium	8	7	87.50	2	2	100.00
5. Pseudo Amino Acid Composition + Physicochemical Properties						
Potassium	48	48	100.00	12	12	100.00
Sodium	8	7	87.50	2	1	50.00
6. Amino Acid Composition + Physicochemical Properties						
Potassium	48	47	97.92	12	12	100.00
Sodium	8	8	100.00	2	1	50.00
7. Pseudo Amino Acid Composition + Amino Acid Composition + Physicochemical Properties						
Potassium	48	48	100.00	12	12	100.00
Sodium	8	7	87.50	2	1	50.00
Average						
Potassium			99.40			97.62
Sodium			91.07			57.14

The performance accuracy was further validated using self consistency test and jackknife test. The overall accuracy of the 3rd layer of MetalloPred for Alkali Metal Ion binding class is 74.04% and 45.71% respectively for the Potassium Metal Ion binding and Sodium Metal Ion binding classes based on self consistency test. Similarly, using jackknife test, the accuracy was found to be 64.52% and 32.85% respectively for the Potassium Metal Ion binding and Sodium Metal Ion binding classes (Table 4.8).

Moreover, the results were robust and hence, the MetalloPred could successfully predict the novel protein sequence into either of Potassium Metal Ion binding or Sodium Metal Ion binding class as evident form the independent data set used for validation (Table 4.8).

Table 4.8 The performance accuracy of the 3rd layer of MetalloPred developed for Alkali Metal Ion binding class based on validation techniques (self consistency test, jackknife test and independent set validation).

Class	Total	Self Consistency	Jackknife	Independent Set	
				Total	Correct %
1. Pseudo Amino Acid Composition					
Potassium	60	91.67	83.33	10	70.00
Sodium	10	60.00	50.00	3	66.67
2. Amino Acid Composition					
Potassium	60	91.67	81.67	10	80.00
Sodium	10	40.00	40.00	3	33.33
3. Physicochemical Properties					
Potassium	60	70.00	55.00	10	70.00
Sodium	10	20.00	0.00	3	33.33
4. Pseudo Amino Acid Composition + Amino Acid Composition					
Potassium	60	31.67	20.00	10	30.00
Sodium	10	30.00	10.00	3	33.33
5. Pseudo Amino Acid Composition + Physicochemical Properties					
Potassium	60	100.00	96.67	10	80.00
Sodium	10	90.00	80.00	3	100.00
6. Amino Acid Composition + Physicochemical Properties					
Potassium	60	85.00	76.67	10	60.00
Sodium	10	50.00	40.00	3	66.67
7. Pseudo Amino Acid Composition + Amino Acid Composition + Physicochemical Properties					
Potassium	60	48.33	38.33	10	40.00
Sodium	10	30.00	10.00	3	33.33
Average					
Potassium		74.04	64.52		61.42
Sodium		45.71	32.85		52.38

By comparing the performance accuracy of the 3rd layer of MetalloPred for Alkali Metal Ion binding class between the individual sequence derived features; it has been observed that the accuracy was better by combining pseudo amino acid and physiochemical properties.

3rd Layer of Neural Network for Transition Metal Ion binding class

The 3rd layer of our MetalloPred tool developed for Transition Metal Ion binding class classified the input protein sequence to be either of Cobalt, Copper, Iron, Manganese, Molybdenum, Nickel or Vanadium Metal Ion binding. The neural network model was trained and tested using training and a test data set based on different types of sequence derived features. The network achieved an overall accuracy of 98.92%, 95.86%, 98.26%, 99.16%, 100%, 99.63%, 97.61% and 98.11% respectively for the Cobalt, Copper, Iron, Manganese, Molybdenum, Nickel and Vanadium Metal Ion binding proteins for the training set data. Similarly the performance accuracy was 100%, 87.65%, 96.42%, 97.80%, 100%, 91.42%, 100% and 93.02% for the test set data. The details of the performance accuracy based on each sequence derived feature have been represented in Table 4.9.

Table 4.9 The summary of the performance accuracy of 3rd layer of MetalloPred developed for Transition Metal Ion binding class based on different sequence derived features.

Class	Train			Test		
	Total	Correct	Correct %	Total	Correct	Correct %
1. Pseudo Amino Acid Composition						
Cobalt	40	37	92.50	10	10	100.00
Copper	221	217	98.19	55	44	80.00
Iron	66	66	100.00	16	14	87.50
Manganese	103	103	100.00	26	26	100.00
Molybdenum	35	35	100.00	9	9	100.00
Nickel	39	38	97.44	10	9	90.00
Vanadium	6	5	83.33	2	2	100.00
Zinc	326	322	98.77	82	77	93.90
2. Amino Acid Composition						
Cobalt	40	40	100.00	10	10	100.00
Copper	221	218	98.64	55	48	87.27
Iron	66	65	98.48	16	16	100.00
Manganese	103	103	100.00	26	26	100.00
Molybdenum	35	35	100.00	9	9	100.00
Nickel	39	39	100.00	10	9	90.00
Vanadium	6	6	100.00	2	2	100.00
Zinc	326	324	99.39	82	74	90.24
3. Physicochemical Properties						
Cobalt	40	40	100.00	10	10	100.00
Copper	221	190	85.97	55	47	85.45
Iron	66	59	89.39	16	15	93.75
Manganese	103	97	94.17	26	24	92.31
Molybdenum	35	35	100.00	9	9	100.00
Nickel	39	39	100.00	10	9	90.00
Vanadium	6	6	100.00	2	2	100.00
Zinc	326	306	93.87	82	74	90.24
4. Pseudo Amino Acid Composition + Amino Acid Composition						
Cobalt	40	40	100.00	10	10	100.00

Copper	221	218	98.64	55	51	92.73
Iron	66	66	100.00	16	15	93.75
Manganese	103	103	100.00	26	26	100.00
Molybdenum	35	35	100.00	9	9	100.00
Nickel	39	39	100.00	10	9	90.00
Vanadium	6	6	100.00	2	2	100.00
Zinc	326	324	99.39	82	76	92.68
5. Pseudo Amino Acid Composition + Physicochemical Properties						
Cobalt	40	40	100.00	10	10	100.00
Copper	221	217	98.19	55	49	88.09
Iron	66	66	100.00	16	16	100.00
Manganese	103	103	100.00	26	25	96.15
Molybdenum	35	35	100.00	9	9	100.00
Nickel	39	39	100.00	10	9	90.00
Vanadium	6	6	100.00	2	2	100.00
Zinc	326	322	98.77	82	74	90.24
6. Amino Acid Composition + Physicochemical Properties						
Cobalt	40	40	100.00	10	10	100.00
Copper	221	217	98.19	55	49	89.09
Iron	66	66	100.00	16	16	100.00
Manganese	103	103	100.00	26	26	100.00
Molybdenum	35	35	100.00	9	9	100.00
Nickel	39	39	100.00	10	9	90.00
Vanadium	6	6	100.00	2	2	100.00
Zinc	326	322	98.77	82	75	91.46
7. Pseudo Amino Acid Composition + Amino Acid Composition + Physicochemical Properties						
Cobalt	40	40	100.00	10	10	100.00
Copper	221	206	93.21	55	45	81.82
Iron	66	66	100.00	16	16	100.00
Manganese	103	103	100.00	26	25	96.15
Molybdenum	35	35	100.00	9	9	100.00
Nickel	39	39	100.00	10	9	90.00
Vanadium	6	6	100.00	2	2	100.00
Zinc	326	319	97.85	82	78	95.12
Average						
Cobalt			98.92			100
Copper			95.86			87.65
Iron			98.26			96.42
Manganese			99.16			97.80
Molybdenum			100			100
Nickel			99.63			91.42
Vanadium			97.61			100
Zinc			98.11			93.02

The performance accuracy was further validated using self consistency test and jackknife test. The overall accuracy of the 3rd layer of MetalloPred for Transition Metal Ion binding class is 66.28%, 57.29%, 63.06%, 57.58%, 69.80%, 59.18%, 60.71% and 56.68% respectively for the Cobalt, Copper, Iron, Manganese, Molybdenum, Nickel and Vanadium Metal Ion binding classes based on self consistency test. Similarly, using jackknife test, the accuracy was found to be 56.00%, 46.53%, 52.96%, 47.06%, 60.06%, 47.81%, 50.00% and 46.00% respectively for the Cobalt, Copper, Iron, Manganese, Molybdenum, Nickel and Vanadium Metal Ion binding classes (Table 4.10). Moreover, the results were robust and hence, the MetalloPred could successfully predict the novel protein sequence into either of Cobalt, Copper, Iron, Manganese, Molybdenum, Nickel or Vanadium Metal Ion binding class as evident form the independent data set used for validation (Table 4.10).

Table 4.10 The performance accuracy of the 3rd layer of MetalloPred developed for Transition Metal Ion binding class based on validation techniques (self consistency test, jackknife test and independent set validation).

Class	Total	Self Consistency	Jackknife	Independent Set	
				Total	Correct %
1. Pseudo Amino Acid Composition					
Cobalt	50	88.00	78.00	5	60.00
Copper	276	78.26	69.20	30	53.33
Iron	82	62.20	53.66	76	60.53
Manganese	129	76.74	68.22	149	66.44
Molybdenum	44	84.09	72.73	5	60.00
Nickel	49	93.88	83.67	5	60.00
Vanadium	8.00	87.50	75.00	-	-
Zinc	408	60.29	51.23	1779	49.13
2. Amino Acid Composition					
Cobalt	50	88.00	80.00	5	80.00
Copper	276	71.38	59.06	30	66.67
Iron	82	67.07	58.54	76	48.68
Manganese	129	70.54	61.24	149	65.77
Molybdenum	44	93.18	84.09	5	60.00
Nickel	49	79.59	67.35	5	80.00
Vanadium	8.00	37.50	37.50	-	-
Zinc	408	60.05	52.45	1779	48.62
3. Physicochemical Properties					
Cobalt	50	72.00	58.00	5	40.00
Copper	276	47.10	33.33	30	50.00
Iron	82	65.85	47.56	76	52.63
Manganese	129	56.59	41.09	149	51.68
Molybdenum	44	79.55	63.64	5	60.00
Nickel	49	46.94	36.73	5	40.00
Vanadium	8.00	87.50	50.00	-	-
Zinc	408	46.81	31.13	1779	42.72
4. Pseudo Amino Acid Composition + Amino Acid Composition					
Cobalt	50	34.00	26.00	5	20.00
Copper	276	38.41	26.81	30	26.67

Iron	82	29.27	18.29	76	18.42
Manganese	129	25.58	14.73	149	18.12
Molybdenum	44	27.27	18.18	5	20.00
Nickel	49	28.57	16.33	5	20.00
Vanadium	8.00	12.50	0.00	-	-
Zinc	408	40.69	26.72	1779	33.05
5. Pseudo Amino Acid Composition + Physicochemical Properties					
Cobalt	50	84.00	76.00	5	80.00
Copper	276	45.29	39.49	30	40.00
Iron	82	75.61	70.73	76	73.68
Manganese	129	63.57	56.59	149	51.68
Molybdenum	44	81.82	77.27	5	80.00
Nickel	49	71.43	63.27	5	80.00
Vanadium	8.00	100.00	100.00	-	-
Zinc	408	64.46	56.86	1779	50.48
6. Amino Acid Composition + Physicochemical Properties					
Cobalt	50	62.00	52.00	5	60.00
Copper	276	70.29	59.06	30	46.67
Iron	82	80.49	71.95	76	78.95
Manganese	129	67.44	55.81	149	61.74
Molybdenum	44	84.09	75.00	5	80.00
Nickel	49	55.10	44.90	5	60.00
Vanadium	8.00	87.50	87.50	-	-
Zinc	408	63.24	54.41	1779	41.65
7. Pseudo Amino Acid Composition + Amino Acid Composition + Physicochemical Properties					
Cobalt	50	36.00	22.00	5	20.00
Copper	276	50.36	38.77	30	40.00
Iron	82	60.98	50.00	76	44.74
Manganese	129	42.64	31.78	149	34.90
Molybdenum	44	38.64	29.55	5	40.00
Nickel	49	38.78	22.45	5	20.00
Vanadium	8.00	12.50	0.00	-	-
Zinc	408	61.27	49.26	1779	46.66
Average					
Cobalt		66.28	56.00		46.00
Copper		57.29	46.53		45.06
Iron		63.06	52.96		49.81
Manganese		57.58	47.06		45.76
Molybdenum		69.80	60.06		49.93
Nickel		59.18	47.81		46.06
Vanadium		60.71	50.00		-
Zinc		56.68	46.00		45.70

By comparing the performance accuracy of the 3rd layer of MetalloPred for Transition Metal Ion binding class between the individual sequence derived features; it has been observed that the accuracy was better by pseudo amino acid as well as by combining pseudo amino acid and physiochemical properties.

Chapter 5

CONCLUSION

From a practical point of view, the most important aspect of a prediction model 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 metal ion binding protein. 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 metal ion binding protein and classes. Additionally much encouraging results have been predicted using the sequence derived parameters technique. Therefore, a much accurate and reliable method is that predicts the metal ion binding proteins and metal ion binding protein classes based on both strategies.

This thesis contains detailed work on metal ion binding protein prediction and classification. We achieved an accuracy of ~ 75% for the prediction of the Metal Ion binding proteins and its classification into major class and sub-classes using three layer artificial neural networks. The first level of network imitates the binary model, the second level of network classify the predicted Metal Ion binding protein into 3 major classes and the third level of network uses the predicted results of the former to provide a much detailed and useful classification. 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 the only best prediction tool available till date, but to the contrary, uses a much simpler and efficient prediction method based on sequence features. This application not only gives optimal results with the dataset used but also predicts metal ion 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.

PUBLICATIONS

1. MetalloPred: An online tool for hierarchical prediction of Metal Ion Binding proteins using cluster of neural networks and sequence derived features.

   Piyush Ranjan, Pooja Kesari, Sankalp Jain and Pradeep Kumar Naik
   (Communicated to: Journal of Computational Biology)

2. TpPred: An online tool for hierarchical prediction of Transport proteins using cluster of neural networks and sequence derived features.

   Sankalp Jain, Piyush Ranjan, Pooja Kesari and Pradeep Kumar Naik
   (Communicated to: Journal of Computational Biology)

References

1. Fontecilla-Camps J.C., Amara P., Cavazza C., Nicolet Y. and Volbeda A. (2009) Structure–function relationships of anaerobic gas-processing Metalloenzymes. Nature 460, 814-822.
2. Stearns D.M. (2000) "Is chromium a trace essential metal?" Biofactors 11, 149–162.
3. Cavalieri R.R. (1997) Iodine metabolism and thyroid physiology: current concepts. Thyroid 7, 177–181.
4. Clapham D.E. (2007) Calcium signaling. Cell 131, 1047–1058.
5. Niki I., Yokokura H., Sudo T., Kato M. and Hidaka H. (1996) Ca2+ signaling and intracellular Ca2+ binding proteins. J. Biochem. 120, 685–698.
6. Eady R.R. (1988) The vanadium-containing nitrogenase of Azotobacter. Biofactors 1, 111-116.
7. Chan M.K., Mukund S., Kletzin A., Adams M.W. and Rees D.C. (1995) Structure of a hyperthermophilic tungstopterin enzyme, aldehyde ferredoxin oxidoreductase. Science 267, 1463–1469.
8. Lane T.W. and Morel F.M. (2000) A biological function for cadmium in marine diatoms. Proc. Natl. Acad. Sci. U.S.A. 97, 4627–4631.
9. Lane T.W., Saito M.A., George G.N., Pickering I.J., Prince R.C. and Morel F.M. (2005) Biochemistry: a cadmium enzyme from a marine diatom. Nature **435**, 42.
10. Karlin K.D., Cruse R.W., Gultneh Y., Farooq A., Hayes J.C. and Zubieta J. (1987) Dioxygen-copper reactivity. Reversible binding of O2 and CO to a phenoxo-bridged dicopper (I) complex. J. Am. Chem. Soc. **109**, 2668–2679.
11. Kitajima N., Fujisawa K., Fujimoto C., Morooka Y., Hashimoto S., Kitagawa T., Toriumi K., Tatsumi K. and Nakamura A. (1992). A new model for dioxygen binding in hemocyanin. J. Am. Chem. Soc. 114, 1277–1291.
12. Moore G.R. and Pettigrew G.W. (1990) Cytochrome c : structural and physicochemical aspects. Berlin: Springer.
13. Sigel A., Sigel H. and Sigel R.K.O., ed (2008). Metal-carbon bonds in enzymes and cofactors. Metal Ions in Life Sciences. **6**. Wiley.
14. Berg J.M. (1990) Zinc finger domains: hypotheses and current knowledge. Annu Rev Biophys Biophys Chem **19**: 405–21.

15. Bishop C.M. (1995) Neural Networks for Pattern Recognition, Oxford: Oxford University Press.
16. Gurney K. (1997) Introduction to Neural Networks. London: Routledge.
17. Haykin S. (1999) Neural Networks: A Comprehensive Foundation, Prentice Hall.
18. Jeanette L. (1994) Introduction to Neural Networks. California Scientific Software Press.
19. Altschul S.F., Gish W., Miller W., Myers E.W. and Lipman D.J. (1990) Basic local alignment search tool. *Journal of Molecular Biology*, **215**, 403-410.
20. Chou K.C. (2001) Prediction of protein cellular attributes using pseudo amino acid composition. PROTEINS: Structure, Function, and Genetics, 43, 246-255.
21. Tanford C. (1962) J Amer. Chem. Soc., 84, 4240-4246.
22. Hopp T.P. and Woods K.R. (1981) P.N.A.S., 78, 3824-3828.
23. Mardia K.V., Kent J.T. and Bibby J.M. (1971) Multivariate Analysis: Chapter 11, Discriminant Analysis; Chapter 12, Multivariate Analysis of Variance; Chapter 13, Cluster Analysis, Academic Press, London, pp.322- 381.

Appendix I

Cluster c code:

An example c parser code of $\text{ANN}_{\text{PseAA}}$ cluster.

```
#include<stdio.h>
#include<math.h>
#include<stdlib.h>
#include<conio.h>
#include<string.h>
#include "L1.h"
#include "L2.h"
#include "L3A.h"
#include "L3AE.h"
#include "L3T.h"
#include "L4TG1.h"
#include "L4TG2.h"
#include "L4TG3.h"

int main()
{

//making Outfile
FILE *OUT;
OUT=fopen("metallopred_out.txt","w");
fclose(OUT);

//Inputting Descriptors
FILE *PAR;
PAR=fopen("par.xls","r");
double desc[37];
char r;
int i;
if(PAR == NULL)
        {
        printf("cannot open file");
        }
for(i=0;i<37;i++)
        {
        fscanf(PAR,"%lg",&desc[i]);
        }
fclose(PAR);

//sending to Layer 1
r=L1::pseaaL1(desc);
if(r=='M')
        {
        //sending to Layer 2
        r=L2::pseaaL2(desc);
        if(r=='A')
                {
                //sending to Layer 3 Alkali
                r=L3A::pseaaL3A(desc);
                }
        else if(r=='E')
                {
                //sending to Layer 3 Alkali Earth
                r=L3AE::pseaaL3AE(desc);
                if(r=='C')
                        {

                        }
                else if(r=='M')
                        {

                        }
                }
        else if(r=='T')
                {
                //sending to Layer 3 Transition
                r=L3T::pseaaL3T(desc);
                if(r=='1')
                        {
                        r=L4TG1::pseaaL4TG1(desc);
                        }
                else if(r=='2')
                        {
                        r=L4TG2::pseaaL4TG2(desc);
                        }
                else if(r=='3')
                        {
                        r=L4TG3::pseaaL4TG3(desc);
                        }
                }
        }

return 0;
}
```

Appendix II

Parser perl code:

Perl parser which links frontend with the descriptor calculation codes and prediction clusters

```
#!"C:/xampp/perl/bin/perl.exe"
#!"C:/xampp/perl/lib"

#prediction starter and output web page compiler

print "Content-type: text/html; charset=iso-8859-1\n\n";
print "<html>";
use CGI qw(:standard);
$pred=new CGI;
use FileHandle;
#taking in the sequence from the html page
$sequence=$pred->param("sequence");
#taking choice of parameters form html page
$pseaa=$pred->param("pseaa");
$aa=$pred->param("aa");
$pep=$pred->param("pepstat");
#checking for errors
#error type - no parameter
if(($pseaa ne "y") && ($aa ne "y") && ($pep ne "y"))
        {
        print "Error!!<br>No Parameter type selected... Go Back Again";
        goto end;
        }
#error type - no sequence
if(!$sequence)
        {
        print "Error!!<br>Sequence field empty... Go Back Again";
        goto end;
        }
#preparing input sequence file
print "<br>Input Sequence:<br>";
open(INP,"+>par.txt");
@seq=split(/[\n]/,$sequence);
$sequence="";
#removing Fasta comment line
if($seq[0] =~ /^>/)
        {
        print splice(@seq,0,1);
        }
```

```
#formatting sequence to be in a single line
$sequence=join("",@seq);
$sequence =~ tr/a-z/A-Z/;
@seq=();
@seq=split(//,$sequence);
$sequence="";
#removing any other exception (errors/non-standard aa) in the sequence
foreach $y(@seq)
        {
        if($y =~ /[ACDEFGHIKLMNPQRSTVWY]/)
                {
                $sequence=$sequence.$y;
                }
        }
print INP ">Query|PDBID|CHAIN|SEQUENCE\n$sequence";
print "<br>$sequence<br>";
close(INP);
#STARTING PREDICTION based on the choice of parameters(user given)
#firing predictor executers accordingly
if(($pseaa eq "y") && ($aa ne "y") && ($pep ne "y"))
        {
        system "modified_pseaa_desc_calc.exe";
        system "1_PSEAA_ANN.exe";
        }
elsif(($pseaa ne "y") && ($aa eq "y") && ($pep ne "y"))
        {
        `2_AA_aa_comp_desc_calc.pl`;
        system "2_AA_ANN.exe";
        }
elsif(($pseaa ne "y") && ($aa ne "y") && ($pep eq "y"))
        {
        `pepstats_calc.pl`;
        `3_PEP_pep_parser.pl`;
        system "3_PEP_ANN.exe";
        }
elsif(($pseaa eq "y") && ($aa eq "y") && ($pep ne "y"))
        {
        system "modified_pseaa_desc_calc.exe";
        `4_PSEAA_AA_aa_comp_desc_calc.pl`;
        system "4_PSEAA_AA_ANN.exe";
        }
elsif(($pseaa eq "y") && ($aa ne "y") && ($pep eq "y"))
        {
        system "modified_pseaa_desc_calc.exe";
        `pepstats_calc.pl`;
        `5_PSEAA_PEP_pep_parser.pl`;
        system "5_PSEAA_PEP_ANN.exe";
        }
elsif(($pseaa ne "y") && ($aa eq "y") && ($pep eq "y"))
        {
        `6_AA_PEP_aa_comp_desc_calc.pl`;
        `pepstats_calc.pl`;
        `6_AA_PEP_pep_parser.pl`;
        system "6_AA_PEP_ANN.exe";
        }
elsif(($pseaa eq "y") && ($aa eq "y") && ($pep eq "y"))
        {
        system "modified_pseaa_desc_calc.exe";
        `7_PSEAA_AA_PEP_aa_comp_desc_calc.pl`;
        `pepstats_calc.pl`;
        `7_PSEAA_AA_PEP_pep_parser.pl`;
        system "7_PSEAA_AA_PEP_ANN.exe";
        }
#printing the output
open(OUT,"metallopred_out.txt");
@output=<OUT>;
close(OUT);
`del par.xls`;
`del par.txt`;
if(glob("pepstat.xls")) {`del pepstat.xls`;}
`del metallopred_out.txt`;
print "<p align=\"center\"><h3>Output of MetalloPred</h3></p><br>";
foreach $y(@output)
        {
        print "$y<br>";
        }
end:
print "</html>";
```

Appendix III

Pseudo amino acid c code:

C code for calculation of sequence derived features which preserves sequence order information

```
/* Pseudo Amino Acid Composition */

#include<stdio.h>
#include<string.h>
#include<stdlib.h>
#include<conio.h>
#include<fstream.h>
#include<iostream.h>
#include<math.h>

int pcount=0;
void getseq();
int aacheck(char);
float H1(int);
float H2(int);
float M(int);
float SD(float A[20]);
float avg(float A[20]);
float J(int,int);
void main()
{
        clrscr();
        getseq();
        cout<<"No of proteins in the file :"<<pcount;
        getch();
}

void getseq()
{
        char ch,file[15],file1[15]={0};
        cout<<"Enter the file containing the sequenecs :";
        cin>>file;
        ifstream infile(file);
        int v=0;
        while(file)
        {
                file1[v]=file[v];
                if(file[v]=='.')
                {
```

```
                file1 [v+1]='x';
                file1 [v+2]='l';
                file1 [v+3]='s';
                break;
            }
            v++;
        }
        ofstream outfile(file1);
        while(infile)
        {
            infile.get(ch);
            if(ch=='>')
            {
                char pname[15]={0};
                int plength=0;
                int i=0;
                while(ch)
                {
                    infile.get(ch);
                    if(ch=='\n')
                    {
                        break;
                    }
                    if(ch=='|')
                        i++;
                    int j=0;
                    while(i==0)
                    {
                            infile.get(ch);
                            pname[j]=ch;
                            j++;
                            if(ch=='|')
                                i++;
                    }
                }
                cout<<pname<<"\n";
                char seq[1800];
                int n=0;
                while(infile)
                {
                        infile.get(ch);
                        if(ch=='\n')
                        {
                            infile.get(ch);
                            if(ch=='\n')
                                break;
                }
                seq[n]=ch;
                n++;
}
plength=n;
int count[21]={0},f;
for(i=0;i<plength;i++)
{
        f=aacheck(seq[i]);
        count[f]=count[f]+1;
}
float arr[20],P[37];
for(int j=0;j<20;j++)
{
        arr[j]=(float)count[j]/plength;
}
float T[17];
for(int r=1;r<18;r++)
{
        float k=0.0;
        for(i=0;i<plength-r;i++)
        {
                int e,f;
                e=aacheck(seq[i]);
                f=aacheck(seq[i+r]);
                if(e!=20 && f!=20)
                        k=k+J(e,f);
        }
        T[r-1]=(k/(plength-r));
}
float t=0.0;
for(i=0;i<17;i++)
        t=t+T[i];

float g=0.0;
for(i=0;i<20;i++)
{
        g=g+arr[i];
}
float tmp=0.0;
tmp=g+(0.5*t);
for(i=0;i<20;i++)
{
        P[i]=(arr[i]*100)/tmp;
}
for(i=0;i<17;i++)
```

```
                        {
                                P[20+i]=(0.5*T[i]*100)/tmp;
                        }
                        for(i=0;i<37;i++)
                        {
                                outfile<<P[i];
                                outfile<<'\t';
                        }
                        outfile<<'\n';
                pcount++;
                        }
                }
}
int aacheck(char h)
{
        int a;
        if(h=='A')
                a=0;
        else if(h=='C')
                a=1;
        else if(h=='D')
                a=2;
        else if(h=='E')
                a=3;
        else if(h=='F')
                a=4;
        else if(h=='G')
                a=5;
        else if(h=='H')
                a=6;
        else if(h=='I')
                a=7;
        else if(h=='K')
                a=8;
        else if(h=='L')
                a=9;
        else if(h=='M')
                a=10;
        else if(h=='N')
                a=11;
        else if(h=='P')
                a=12;
        else if(h=='Q')
                a=13;
        else if(h=='R')
                a=14;
```

```
          else if(h=='S')
                    a=15;
          else if(h=='T')
                    a=16;
          else if(h=='V')
                    a=17;
          else if(h=='W')
                    a=18;
          else if(h=='Y')
                    a=19;
          else
                    a=20;
          return a;
}
float J(int x1,int x2)
{
          float j,k;
          k=(pow((H1(x2)-H1(x1)),2)+pow((H2(x2)-H2(x1)),2)+pow((M(x2)-M(x1)),2));
          j=k/3;
          return j;
}
float H1(int s)
{
          float H1[20]={0.62,0.29,-0.90,-0.74,1.19,0.48,-0.40,1.38,-1.50,1.06,0.64,-
0.78,0.12,-0.85,-2.53,-0.18,-0.05,1.08,0.81,0.26};
          float H;
          H=(H1[s]-avg(H1))/SD(H1);
          return H;
}
float H2(int s)
{
          float H2[20]={-0.5,-1.0,3.0,3.0,-2.5,0.0,-0.5,-1.8,3.0,-1.8,-1.3,0.2,0.0,0.2,3.0,0.3,-
0.4,-1.5,-3.4,-2.3};
          float H;
          H=(H2[s]-avg(H2))/SD(H2);
          return H;
}
float M(int s)
{
          float
M[20]={15.0,47.0,59.0,73.0,91.0,1.0,82.0,57.0,73.0,57.0,75.0,58.0,42.0,72.0,101.0,31.0,
45.0,43.0,130.0,107.0};
          float m;
          m=(M[s]-avg(M))/SD(M);
          return m;
}
```

```
float SD(float A[20])
{
        float sd,a,s=0.0;
        a=avg(A);
        for(int i=0;i<20;i++)
                s=s+pow((A[i]-a),2);
        sd=sqrt(s/20);
        return sd;
}
float avg(float A[20])
{
        float avg,a=0.0;
        for(int i=0;i<20;i++)
                a=a+A[i];
        avg=a/20;
        return avg;
}
```

Appendix IV

Amino acid perl code:

Perl code for calculation of sequence derived features based on amino acid composition

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

#reading file into array
$i=0;
while(<file>)
        {
        if(/^>/)
                {
                $i++;
                $name[$i]=$_;
                }
        else
                {
                chomp($_);
                $seq[$i]=$seq[$i].$_;
                }
        }


#Reference array
$ref=(ACDEFGHIKLMNPQRSTVWY);
@ref=split('',$ref);

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

#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;
                              }
                    }
          }
$proname=(split /[|]/,$name[$i])[0];
print "protein: $proname\t@freq\n";
print desc "$proname\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";
}
```